UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Daniel Ricardo dos Santos

UMA ARQUITETURA DE CONTROLE DE ACESSO DINÂMICO BASEADO EM RISCO PARA COMPUTAÇÃO EM NUVEM

Florianópolis(SC)

2013

Daniel Ricardo dos Santos

UMA ARQUITETURA DE CONTROLE DE ACESSO DINÂMICO BASEADO EM RISCO PARA COMPUTAÇÃO EM NUVEM

Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.
Orientadora: Dr[a]. Carla Merkle Westphall,

Florianópolis(SC)

2013

Catalogação na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:
http://www.bu.ufsc.br/design/Catalogacao.html

Daniel Ricardo dos Santos

UMA ARQUITETURA DE CONTROLE DE ACESSO DINÂMICO BASEADO EM RISCO PARA COMPUTAÇÃO EM NUVEM

Esta Dissertação foi julgada aprovada para a obtenção do Título de "Mestre em Ciência da Computação", e aprovada em sua forma final pelo Programa de Pós-graduação em Ciência da Computação.

Florianópolis(SC), 09 de agosto 2013.

Dr. Ronaldo dos Santos Mello
Coordenador

Dr[a]. Carla Merkle Westphall,
Orientadora

Banca Examinadora:

Dr[a]. Carla Merkle Westphall
Presidente

Dr. Mário Antônio Ribeiro Dantas

Dr. Aldri Luiz dos Santos

Dr. Carlos Becker Westphall

“Essentially, all models are wrong, but some are useful.”

George Box

RESUMO

Computação em nuvem é um modelo para computação distribuída que ainda enfrenta problemas. Novas ideias surgem para aproveitar ainda mais suas características e entre os desafios de pesquisa encontrados na computação em nuvem destaca-se a gerência de identidades e controle de acesso. Os principais problemas da aplicação de controle de acesso em computação em nuvem são a necessária flexibilidade e escalabilidade para suportar um grande número de usuários e recursos em um ambiente dinâmico e heterogêneo, com as necessidades de colaboração e compartilhamento de recursos e informações. Esse trabalho de pesquisa propõe o uso de controle de acesso dinâmico baseado em risco para computação em nuvem. A proposta é apresentada na forma de um modelo para controle de acesso, baseado em uma extensão do padrão XACML com três novos componentes principais: o *Risk Engine*, os *Risk Quantification Web Services* e as políticas de risco. As políticas de risco apresentam um método para descrever métricas de risco e sua quantificação, que pode ser através de funções locais ou remotas. O uso de políticas de risco permite que usuários e provedores de serviços de nuvens definam como desejam tratar o controle de acesso baseado em risco para seus recursos, utilizando métodos de quantificação e agregação de risco apresentados em trabalhos relacionados. O modelo atinge a decisão de acesso baseado em uma combinação de decisões XACML e análise de risco. Uma especificação das políticas de risco utilizando XML é apresentada e um estudo de caso utilizando federações de nuvens é descrito. Um protótipo do modelo é implementado, mostrando que tem expressividade suficiente para descrever os modelos de trabalhos relacionados. Nos resultados experimentais o protótipo atinge decisões de acesso com o uso de políticas de trabalhos relacionados com um tempo entre 2 e 6 milissegundos. Uma discussão sobre os aspectos de segurança do modelo também é apresentada.

Palavras-chave:
Computação em Nuvem; Controle de Acesso; Risco; Federação de nuvens;

ABSTRACT

Cloud computing is a distributed computing model that still faces problems. New ideas emerge to take advantage of its features and among the research challenges found in cloud computing, we can highlight Identity and Access Management. The main problems of the application of access control in the cloud are the necessary flexibility and scalability to support a large number of users and resources in a dynamic and heterogeneous environment, with collaboration and information sharing needs. This research work proposes the use of risk-based dynamic access control for cloud computing. The proposal is presented as an access control model based on an extension of the XACML standard with three new main components: the Risk Engine, the Risk Quantification Web Services and the Risk Policies. The risk policies present a method to describe risk metrics and their quantification, using local or remote functions. The use of risk policies allows users and cloud service providers to define how they wish to handle risk-based access control for their resources, using quantification and aggregation methods presented in related works. The model reaches the access decision based on a combination of XACML decisions and risk analysis. A specification of the risk policies using XML is presented and a case study using cloud federations is described. A prototype of the model is implemented, showing it has enough expressivity to describe the models of related works. In the experimental results, the prototype reaches access decisions using policies based on related works with a time between 2 and 6 milliseconds. A discussion on the security aspects of the model is also presented.

Keywords:
Cloud Computing; Access Control; Risk; Cloud Federation;

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1 INTRODUÇÃO

Nesse capítulo apresenta-se uma introdução ao trabalho desenvolvido, na forma de uma motivação, descrição do problema a ser tratado e hipótese a ser testada, objetivos gerais e específicos, limitações e método de pesquisa. A organização do restante do texto é apresentada ao final do capítulo.

1.1 MOTIVAÇÃO

Computação em nuvem é um modelo de entrega de recursos e serviços computacionais para usuários através da Internet, fornecendo características como facilidade de acesso, rápida elasticidade e compartilhamento de recursos entre usuários. Essa abordagem apresenta vantagens para clientes e provedores de serviços, sendo as principais a redução de custos de Tecnologia da Informação (TI), o aumento na escalabilidade, a utilização mais eficiente de recursos e o modelo de cobrança por uso (BOSS et al., 2007).

Com o desenvolvimento da computação em nuvem e sua adoção mais generalizada, surgem novas ideias para explorar ainda mais esse modelo e torná-lo mais eficiente e escalável. Entre essas ideias encontra-se o conceito de federação de nuvens, que visa tornar possível o compartilhamento de dados e recursos entre diversas nuvens que colaboram na mesma federação. Esse conceito foi proposto recentemente e já existem projetos de implementações concretas (CARLINI et al., 2012; ROCHWERGER et al., 2009; EGI, 2012).

Alguns problemas, no entanto, ainda perduram nos modelos de computação em nuvem e federações de nuvens. Entre esses problemas, os que envolvem as áreas de segurança e privacidade merecem destaque especial (REN; WANG; WANG, 2012), sendo a segurança fundamental para garantir o sucesso da computação em nuvem (LEE; JEUN; JUNG, 2009; TAKABI; JOSHI; AHN, 2010; GROBAUER; WALLOSCHEK; STOCKER, 2011). O aumento no número de usuários e serviços disponíveis na nuvem, além da grande dinamicidade e heterogeneidade desse ambiente, torna necessário gerenciar de maneira segura e eficiente quem são esses usuários e quais recursos podem acessar.

Os processos que tratam de identidades de usuários e acesso a recursos são conhecidos de maneira geral como “gerência de identidades e controle de acesso”, ou *Identity and Access Management* (IAM) e são fundamentais em ambientes de computação em nuvem para garantir requisitos como privacidade, confidencialidade e integridade dos dados (GROBAUER; WALLOSCHEK; STOCKER, 2011; BERTINO; TAKAHASHI, 2011).

O controle de acesso é um mecanismo essencial para garantir a segurança das informações armazenadas na nuvem (ZISSIS; LEKKAS, 2012). No entanto, os modelos de controle de acesso tradicionais, que ainda são os mais comumente implementados em sistemas de computação em nuvem, apresentam alguns problemas nesse tipo de ambiente altamente complexo, dinâmico e distribuído, entre eles: a falta de escalabilidade e flexibilidade e o uso de políticas estáticas (FALL et al., 2011). O reconhecimento desses problemas leva ao estudo e a criação de novos modelos de controle de acesso para o uso em ambientes de nuvens.

Por outro lado, modelos de controle de acesso dinâmico, como os baseados em risco e contexto, surgiram para lidar com o problema de ambientes muito dinâmicos em que a aplicação dos modelos tradicionais apresenta problemas (JASON Program Office, 2004). Além disso, esses modelos conseguem tratar situações excepcionais, em que uma requisição de acesso deve ser liberada para um usuário não previamente autorizado, uma operação conhecida como *"break the glass"*.

O principal problema resolvido por esse tipo de modelo de controle de acesso é a flexibilidade no acesso aos recursos. Os modelos tradicionais utilizam políticas de controle de acesso rígidas e estáticas e essas políticas atingem os objetivos propostos, mas não se adequam bem à ambientes dinâmicos e heterogêneos como a nuvem, que apresentam alteração constante nos usuários e recursos disponíveis.

Os modelos dinâmicos tendem a realizar uma autorização em "tempo real", utilizando alguma estratégia calculada para cada tentativa de acesso, seja ela baseada em risco, confiança ou contexto (SUHENDRA, 2011).

1.2 PROBLEMA E HIPÓTESE

O problema a ser tratado nesse trabalho é a falta de um modelo para controle de acesso flexível o suficiente e adaptado para utilização em um cenário de computação em nuvem, que suporte o tratamento de requisições excepcionais de acesso.

Uma aplicação importante desse conceito se dá em ambientes de federação de nuvens, que normalmente utilizam a abordagem de identidades federadas para a gerência de identidades e um modelo de controle de acesso baseado em papéis ou atributos para controle de acesso. O uso dessas abordagens, na prática, limita a escalabilidade dessas arquiteturas, por causa da necessidade de acordos prévios de confiança entre os membros da federação e da falta de interoperabilidade entre padrões de gerência de identidades (LAMPROPOULOS; DENAZIS, 2012).

A hipótese a ser testada é que um modelo de controle de acesso dinâmico baseado em risco consegue gerenciar o controle de acesso em uma nuvem ou federação de nuvens, mantendo as características de segurança desejadas por provedores de serviços de nuvem e clientes e, ao mesmo tempo, flexibilizando o acesso aos recursos compartilhados.

1.3 OBJETIVO

1.3.1 Objetivo geral

O objetivo deste trabalho é propor e implementar um modelo para controle de acesso dinâmico baseado em risco para computação em nuvem. O sistema deve gerenciar o controle de acesso aos recursos compartilhados na nuvem e atender aos requisitos mínimos de segurança de provedores de serviço de nuvem e clientes. Deve ainda permitir o acesso a recursos em situações excepcionais. O modelo deve apresentar um desempenho satisfatório e permitir grande escalabilidade.

Até onde a literatura foi consultada, não foi possível encontrar o desenvolvimento dessa hipótese em nenhum trabalho. Os trabalhos mais importantes são descritos e comentados no capítulo 3.

1.3.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Identificar problemas no controle de acesso tradicional em computação em nuvem;
- Propor um modelo para controle de acesso em computação em nuvem que combine abordagens existentes e uma abordagem baseada em risco;
- Definir um cenário de utilização do modelo; e
- Implementar, testar e validar a modelo e discutir seus resultados.

1.4 LIMITAÇÕES

Como o foco do trabalho é o controle de acesso em computação em nuvem, algumas questões são desconsideradas para limitar a extensão da pes-

quisa. Em especial, o escopo deste trabalho é limitado pelos seguintes fatores:

- O modelo se concentra na autorização dos usuários, por isso assume que a autenticação funciona corretamente;
- O modelo considera que os provedores de serviços de nuvem são confiáveis, para que não seja necessário cifrar e decifrar todos os recursos disponíveis; e
- O modelo considera que todas as funções necessárias ao suporte da função de autorização não apresentam problemas, para que não seja necessário implementar tolerância a falhas.

1.5 MÉTODO DE PESQUISA

O método de pesquisa utilizado é a pesquisa experimental quantitativa, obtendo medições de desempenho e estatísticas de uso da implementação.

As etapas da pesquisa são as seguintes:

- Identificação dos problemas de controle de acesso através da leitura de trabalhos relacionados e experimentos em ambientes de computação em nuvem;
- Definição de um modelo para controle de acesso dinâmico baseado em risco para computação em nuvem;
- Definição de um cenário de utilização do modelo para controle de acesso;
- Implementação do modelo baseado no cenário definido;
- Definição dos experimentos a serem realizados;
- Realização dos experimentos e obtenção dos resultados; e
- Discussão a respeito dos resultados obtidos.

1.6 ORGANIZAÇÃO DO TEXTO

O restante desse texto está organizado da seguinte forma: o capítulo 2 descreve os conceitos básicos de computação em nuvem, federações de nuvens, gerência de identidades e controle de acesso; o capítulo 3 apresenta e discute os principais trabalhos relacionados; o capítulo 4 apresenta o modelo

proposto no trabalho; o capítulo 5 descreve a implementação dessa proposta, os experimentos realizados e resultados obtidos e, finalmente, o capítulo 6 traz as conclusões e possibilidades de trabalhos futuros.

2 COMPUTAÇÃO EM NUVEM, GERÊNCIA DE IDENTIDADES E CONTROLE DE ACESSO

Nesse capítulo apresentam-se os principais conceitos relacionados a computação em nuvem: definição, modelos de implantação e modelos de serviço. Além disso, discute-se sobre o conceito de federações de nuvens e são feitas algumas considerações sobre segurança em ambientes de computação em nuvem.

Apresentam-se também os conceitos básicos sobre gerência de identidades e controle de acesso, discutindo-se definições, modelos, arquiteturas e implementações.

2.1 COMPUTAÇÃO EM NUVEM

De acordo com Mell e Grance (2011), computação em nuvem é um modelo de computação que permite acesso ubíquo, conveniente e sob demanda a um conjunto compartilhado de recursos computacionais configuráveis. Esses recursos podem ser servidores, armazenamento ou mesmo aplicações e serviços e podem ser rapidamente e facilmente fornecidos e liberados aos usuários.

A computação em nuvem apresenta cinco características essenciais (MELL; GRANCE, 2011):

Auto serviço sob demanda Os recursos devem ser providos aos usuários de forma automática, sem a necessidade de interação humana com cada provedor;

Amplo acesso de rede Os recursos devem estar disponíveis pela rede e deve ser possível acessá-los através de diferentes dispositivos, como computadores, celulares e *tablets*;

Compartilhamento de recursos Os recursos de um provedor devem ser compartilhados por diferentes clientes, num modelo de multi-inquilinos. Os clientes que utilizam esses recursos não precisam conhecer a localização e características exatas desses recursos, apenas sua utilização;

Elasticidade rápida Os recursos podem ser providos elasticamente, aumentando ou diminuindo conforme a necessidade do cliente e, muitas vezes, automaticamente. Sob o ponto de vista do usuário os recursos disponíveis parecem ilimitados; e

Serviço mensurado Os recursos são medidos e otimizados automaticamente e seu uso é monitorado e relatado, de forma que provedores e clientes tenham transparência nos serviços utilizados.

Existem diversas definições e listas de características essenciais da computação em nuvem disponíveis na literatura (MATHER; KUMARASWAMY; LATIF, 2009; ARMBRUST et al., 2010; FOSTER et al., 2008; BUYYA; BROBERG; GOSCINSKI, 2011). No entanto, a definição do *National Institute of Standards and Technology* (NIST), apresentada anteriormente, é a que vem sendo mais adotada atualmente.

Computação em nuvem é uma aplicação da ideia de computação utilitária, baseada no fornecimento de recursos computacionais como um serviço sob demanda, e uma evolução da aplicação de vários conceitos provenientes de diversas áreas da computação como *hardware*, computação distribuída, gerência de redes e tecnologias de Internet (BUYYA; BROBERG; GOSCINSKI, 2011). Para usuários e provedores de serviços, a computação em nuvem representa um novo paradigma na forma de acesso e utilização de recursos computacionais.

As nuvens computacionais são apoiadas sobre tecnologias como virtualização e *Service-oriented Architecture* (SOA)e, assim, conseguem trazer vantagens como a redução de custos de manutenção e uma rápida e fácil escalabilidade, além de possibilitarem a migração eficiente de serviços.

Apesar de todas as vantagens e de uma forte adoção por parte de provedores de serviços, o modelo de computação em nuvem ainda apresenta alguns problemas não resolvidos, especialmente do ponto de vista do usuário. A segurança é, geralmente, apontada como o principal receio de potenciais usuários da nuvem, especialmente no caso de nuvens públicas, onde os dados ficam sob a custódia de terceiros (REN; WANG; WANG, 2012). Em uma pesquisa realizada em 2009, 90% dos respondentes afirmaram que o controle de acesso é uma tecnologia essencial para a construção da computação em nuvem (F5 Networks, 2009).

2.1.1 Modelos de implantação

Existem quatro modelos básicos de implantação de uma nuvem computacional. Esses modelos se referem principalmente à localização da infraestrutura da nuvem e ao público a que se destina essa infraestrutura. Os quatro modelos básicos são (MELL; GRANCE, 2011):

Privada Nesse modelo a nuvem é projetada e implantada para ser utilizada apenas pelos membros de uma única organização. A infraestrutura

pode estar localizada interna ou externamente à organização que a utiliza e pode ser gerenciada por membros da organização ou por terceiros.

Comunitária A nuvem é implantada para ser utilizada por membros de uma comunidade, geralmente de organizações diferentes, que compartilham interesses e necessidades em comum. Novamente, a localização física e a gerência da nuvem podem se dar em uma das organizações participantes da comunidade ou através de terceiros.

Pública A nuvem é implantada para utilização pelo público em geral. Nesse modelo um provedor de serviços de nuvem é responsável pela infraestrutura e gerência da nuvem. Esse provedor pode ser uma organização comercial, acadêmica, governamental ou qualquer combinação destas.

Híbrida A infraestrutura da nuvem híbrida é composta por nuvens de dois ou mais tipos das anteriores, que são entidades separadas, mas se comunicam através de protocolos estabelecidos e permitem a portabilidade de informações entre si.

2.1.2 Modelos de serviço

O objetivo final de uma nuvem computacional é prover serviços para os usuários. As nuvens podem, então, ser classificadas de acordo com o serviço que proveem. Normalmente, três níveis de serviço são considerados nessa classificação, são eles (MELL; GRANCE, 2011):

***Software as a Service* (SaaS)** Nesse modelo os usuários finais acessam aplicações que executam na nuvem. Essas aplicações executam na infraestrutura do provedor de serviços de nuvem e são gerenciadas por um usuário desse provedor. A gerência de toda a infraestrutura e das plataformas sobre as quais as aplicações executam recaem sobre o provedor de serviços de nuvem (*Cloud Service Provider* - CSP).

***Platform as a Service* (PaaS)** Nesse modelo o CSP oferece como serviço o acesso a plataformas de desenvolvimento na nuvem que podem ser utilizadas por usuários para desenvolverem suas aplicações e posteriormente executá-las na infraestrutura de nuvem fornecida. Geralmente os usuários têm acesso a um conjunto de ferramentas de desenvolvimento, além de APIs e bibliotecas específicas que permitem a construção de serviços de nuvem.

***Infrastructure as a Service* (IaaS)** Esse é o modelo de mais baixo nível, em que os usuários têm acesso direto às máquinas virtuais que executam os serviços desejados. É comum que os usuários tenham acesso a configurações como processamento e memória das máquinas, além de definirem o número de máquinas que serão instanciadas.

Além desses três modelos é comum alguns autores considerarem outras classificações, que dizem respeito a serviços específicos providos por uma nuvem. Essas classificações são conhecidas como XaaS, onde X representa a sigla do serviço que se deseja prover (SCHAFFER, 2009). Como exemplos mais relevantes a esse trabalho podemos citar: *Identity as a Service* (IDaaS) e *Authorization as a Service* (AaaS) (OLDEN, 2011), outros exemplos seriam *Storage as a Service* (STaaS) e *Security as a Service* (SECaaS) (CARROLL; MERWE; KOTZE, 2011). No entanto, essas classificações são apenas especializações das três categorias anteriores.

2.1.3 Exemplos de serviços

Atualmente, diversas grandes empresas de *software* oferecem alternativas para seus produtos na nuvem ou migraram totalmente suas operações para esse novo modelo.

Alguns dos exemplos mais utilizados de SaaS são *software* de *Customer Relationship Management* (CRM), como o da Salesforce.com e aplicações de *Enterprise Resource Planning* (ERP), como a da SAP (CLOUDS360, 2013c).

No nível de PaaS, alguns dos serviços mais conhecidos são o Google AppEngine e Microsoft Azure, que oferecem plataformas de desenvolvimento e execução de aplicações na nuvem, com opções de linguagens de programação, APIs próprias, armazenamento integrado e possibilidade de escalabilidade automática das aplicações (CLOUDS360, 2013b).

Entre os maiores provedores comerciais de IaaS, é possível citar Amazon Web Services, Rackspace e GoGrid (CLOUDS360, 2013a). Além dessas ofertas comerciais, existem diversas alternativas de *software* para criação e gerenciamento de ambientes de nuvens, como Eucalyptus, OpenNebula e OpenStack (LASZEWSKI et al., 2012).

2.1.4 Federações de nuvens

O paradigma de computação em nuvem vem alcançando um relativo sucesso devido às suas já mencionadas vantagens. Entretanto, para utilizar

todo o potencial desse modelo é necessário dar um passo adiante na direção das federações de nuvens (HARSH et al., 2011).

De acordo com Kurze et al. (2011), uma federação de nuvens compreende serviços de diferentes provedores agregados em um conjunto único apoiando três características básicas de interoperabilidade: migração de recursos, redundância de recursos e combinação de recursos ou serviços complementares.

A migração de recursos permite que um usuário instancie seus recursos em diferentes domínios e possa depois relocá-los. A principal vantagem da migração para o usuário é evitar que um recurso, após instanciado em um provedor, fique sempre "preso" a esse mesmo provedor. Dessa forma o usuário pode mover seu recurso entre os provedores da federação para aproveitar benefícios oferecidos por cada provedor individual.

A redundância de recursos permite que os usuários utilizem serviços semelhantes em diferentes domínios, evitando pontos únicos de falhas e, por fim, a combinação de recursos permite uma maior modularização de serviços, resultando em um ambiente mais eficiente e flexível.

Várias propostas e arquiteturas de federação de nuvens vêm sendo discutidas na literatura, tendo como ideia principal agregar conjuntos de nuvens através de protocolos padrão, de forma que as nuvens possam interagir entre si e aproveitar os recursos disponíveis umas das outras. Alguns autores também chamam essa prática de multi-nuvens ou ainda nuvens de nuvens (ALZAIN et al., 2012; VUKOLIć, 2010).

Segundo Harsh et al. (2011), uma verdadeira federação de nuvens coloca nuvens públicas e privadas sob uma mesma federação onde usuários tem a opção de instanciar serviços usando os recursos de múltiplos provedores. Os usuários devem ser apresentados à características únicas de API, cobrança e monitoramento, e qualquer organização pode se tornar parte da federação, tanto provendo quanto utilizando os serviços de nuvem.

Kurze et al. (2011) considera os modelos de serviço da computação em nuvem (IaaS, PaaS e SaaS) como uma pilha e classifica duas dimensões de federações em nuvem. A federação horizontal ocorre em apenas um nível da pilha de nuvem, por exemplo, apenas no nível de aplicações, enquanto a federação vertical acontece entre diferentes níveis.

As principais vantagens desse modelo são um aumento ainda maior na escalabilidade e disponibilidade. A federação também permite a possibilidade de que provedores de serviços de nuvens menores ou nuvens privadas possam se unir para compartilhar recursos, aumentando a percepção que os clientes têm de recursos ilimitados. Além disso, espera-se que, com a migração de máquinas virtuais e recursos entre nuvens federadas, seja possível melhorar a migração e interoperabilidade de dados e evitar proble-

mas como o aprisionamento tecnológico de recursos em um único provedor. A federação também traz vantagens econômicas, já que o compartilhamento de recursos entre provedores diminui a necessidade de recursos de cada provedor individual (GOIRI; GUITART; TORRES, 2010).

Pensando ainda mais adiante, já existe a proposta da criação de uma *Intercloud*, nos moldes da Internet, mas que em vez de redes possa agregar recursos computacionais de diferentes nuvens. Existem algumas propostas de implementação dessa ideia (BERNSTEIN et al., 2009; BUYYA; RANJAN; CALHEIROS, 2010) e um comitê do *Institute of Electrical and Electronics Engineers* (IEEE), o P2302 - *Standard for Intercloud Interoperability and Federation* (SIIF), está trabalhando na padronização da arquitetura e dos protocolos necessários (DIAMOND, 2012).

O trabalho de Celesti et al. (2010b) propõe a ideia de que a evolução da computação em nuvem se dá em três estágios: Monolítico, onde atualmente nos encontramos, com grandes CSPs separados, baseados em arquiteturas proprietárias e provendo serviços apenas a seus usuários; Cadeia de Suprimentos Vertical, onde alguns provedores começam a utilizar serviços de outros provedores, mas ainda com a existência de "ilhas" de arquiteturas proprietárias; e, finalmente, Federação Horizontal, onde a federação se dá entre provedores de vários tamanhos para que atinjam uma economia de escala, um uso eficientes de seus recursos físicos e um aumento das suas capacidades. Os autores indicam as limitações das arquiteturas usuais para atingir o nível de Federação Horizontal e ainda propõem um mecanismo de gerência de federação para permitir que se atinja esse objetivo.

Os principais projetos de federação de nuvens em atividade atualmente são o Contrail, o Reservoir fp7 e o mOSAIC, todos da União Europeia (GROZEV; BUYYA, 2012).

Além da federação de recursos, existem propostas focadas em serviços de agenciamento (*brokering*) entre provedores. O objetivo do agente (*broker*) é possibilitar que um usuário conecte as diversas nuvens de que faz parte e possa acessá-las através de uma mesma interface e, para que isso aconteça, é necessário que o agente entenda as APIs dos diferentes provedores. Os agentes atualmente mais importantes que podem ser citados são: RightScale e SpotCloud (ORBEGOZO et al., 2011), que oferece um mercado onde os usuários podem escolher os provedores que oferecem os menores preços e ao mesmo tempo disponibilizar recursos não alocados dos seus *data centers*.

2.1.5 Considerações sobre segurança

A segurança é uma das características mais importantes quando se trata da migração e da utilização de serviços nas nuvens.

A questão da segurança na computação em nuvem pode ser analisada de diversos ângulos. Primeiramente, podemos pensar na segurança da informação em nuvem. Questões como governança, auditoria, gerência de risco e continuidade de negócios apresentam novas nuances nesse cenário, já que muitos dos ativos de uma organização podem passar a estar sob controle terceirizado (CATTEDDU; HOGBEN, 2009).

A conformidade com a legislação e regulamentações específicas também é de suma importância, já que é comum que as nuvens estejam envolvidas em operações de negócios internacionais, atravessando diversas jurisdições e manipulando dados com muitas origens e destinos diferentes. Quando uma organização que precisa seguir regulamentações específicas, como as que lidam com dados médicos e financeiros, migra suas operações para a nuvem, ela precisa de garantias do provedor de serviços de nuvem de que este também segue as mesmas regulamentações. Aplicações médicas podem tirar muitas vantagens da nuvem, mas apresentam necessidades de segurança específicas (SOUZA et al., 2013a, 2013b).

Além disso existem as questões técnicas de segurança como gerência de identidade, controle de acesso, privacidade, confidencialidade, disponibilidade e integridade.

Quando se analisa a segurança de ambientes de nuvem é importante definir exatamente o cenário em que se está trabalhando. As necessidades de segurança em uma nuvem privada são diferentes das de uma nuvem pública. Da mesma maneira as ameaças no nível de SaaS são diferentes daquelas encontradas no nível de IaaS (Cloud Security Alliance, 2011; GROBAUER; WALLOSCHEK; STOCKER, 2011).

A responsabilidade de usuários e provedores também muda conforme altera-se o nível em que se trabalha. Os contratos de serviço e acordos de nível de serviço (*Service Level Agreement* - SLA) devem ser claros a respeito das responsabilidades de segurança quando da contratação de um serviço de nuvem. Deve-se, inclusive, considerar a existência de SLAs específicos para segurança, onde sejam descritas métricas de segurança e níveis aceitáveis para essas métricas (CHAVES; WESTPHALL; LAMIN, 2010).

Segundo Grobauer, Walloschek e Stocker (2011) as tecnologias utilizadas para prover os serviços de nuvem trazem consigo possíveis problemas de segurança. Por exemplo, os meios de acesso aos recursos armazenados na nuvem tendem a ser inseguros, assim como as aplicações *web* e *web services* muitas vezes utilizados para a gerência das nuvens apresentam um longo

histórico de problemas de segurança (BRINHOSA et al., 2013).

A *Cloud Security Alliance* (CSA) é uma organização sem fins lucrativos que define e promove o uso de boas práticas de segurança em computação em nuvem. A CSA publica guias abrangentes de segurança em ambientes de nuvem e propôs um conjunto de controles de segurança a ser utilizado nesses ambientes, conhecido como *Cloud Security Matrix* (CSM).

A abordagem da CSA é dividir a segurança na computação em nuvem em diferentes domínios, cada um representando uma área de interesse da segurança. Para cada domínio são apresentados os principais desafios e sugestões de soluções. Atualmente são considerados 14 domínios, sendo eles (Cloud Security Alliance, 2011): (1) Arcabouço arquitetural para computação em nuvem; (2) Governança e Gerência de Risco Empresarial; (3) Questões legais: contratos e *e-discovery*; (4) Gerência de conformidade e auditoria; (5) Gerência de informações e segurança de dados; (6) Interoperabilidade e portabilidade; (7) Segurança tradicional, continuidade dos negócios e recuperação de desastres; (8) Operações de *data center*; (9) Resposta à incidentes; (10) Segurança de aplicações; (11) Criptografia e gerência de chaves; (12) Gerência de identidades, delegação e controle de acesso; (13) Virtualização; e (14) Segurança como um serviço.

Esse trabalho é focado nas questões técnicas do controle de acesso em computação em nuvem e, por isso, se encaixa no domínio 12.

2.2 GERÊNCIA DE IDENTIDADES E CONTROLE DE ACESSO

Gerência de identidades e controle de acesso é um conceito que engloba vários processos relacionados à identificação, autenticação, autorização e responsabilidade de usuários em sistemas computacionais. Nesse contexto é importante diferenciar conceitos relacionados como: identidade, autenticação e autorização.

Identidade é um conjunto de informações que representa uma entidade em um sistema (ITU-T, 2009b). Essa identidade geralmente não representa tudo que é possível sobre aquela entidade e, por isso, trata-se de uma identidade parcial, contendo dados sobre a entidade que são relevantes no contexto do sistema. É possível, e bastante comum, que uma mesma entidade tenha diversas identidades diferentes, dependendo do serviço que está utilizando (PFITZMANN; HANSEN, 2010).

Autenticação é o processo pelo qual uma entidade comprova sua identidade perante um sistema (BENANTAR, 2006). A autenticação pode ocorrer de diversas formas: através do uso de senhas, certificados digitais, biometria, padrões de comportamento ou uma combinação entre essas e outras

formas. A forma de autenticação utilizada em um sistema é importante, pois carrega consigo a noção de um nível de confiança (*Level of Assurance* - LoA) (CHADWICK, 2009). Sistemas que utilizam biometria, por exemplo, costumam possuir um nível de confiança maior do que aqueles que utilizam senhas, já que geralmente é mais fácil para um atacante ter acesso a uma senha do que forjar uma autenticação biométrica.

Autorização, também chamado de controle de acesso é o processo pelo qual o sistema garante que as requisições de acesso a recursos feitas por usuários são validadas perante regras pré-definidas (BENANTAR, 2006). Essas regras são conhecidas como políticas e a maneira pela qual essas políticas são definidas e administradas constituem um modelo de controle de acesso.

O nível de detalhes, quantidade de informações e certeza acerca dessas informações em uma identidade, além do nível de confiança requerido na autenticação e do modelo de controle de acesso utilizado dependem da criticidade do sistema desenvolvido, da manipulação de informações sensíveis ou confidenciais e dos custos associados.

Como em todos os controles de segurança, os princípios de disponibilidade, integridade e confidencialidade são fundamentais nos processos de gerência de identidades e controle de acesso (HARRIS, 2013). Além desses princípios, um elemento fundamental relacionado à gerência de identidades é a privacidade (SANTOS; WESTPHALL, 2011).

2.2.1 Gerência de Identidades

Segundo Lee, Jeun e Jung (2009), um serviço de gerência de identidades pode ser definido como "o processo de criação, gerência e utilização de identidades de usuários e a infraestrutura que suporta esse processo". Três entidades principais compõem esse tipo de sistema, são elas (BERTINO; TAKAHASHI, 2011):

Usuário É a entidade que possui uma identidade e utiliza os serviços tanto do provedor de identidades quanto do provedor de serviços.

Provedor de Identidades (*Identity Provider* - IdP) É aquele que fornece os serviços de gerenciamento de identidades, necessários para que o usuário utilize o provedor de serviços.

Provedor de serviços (*Service Provider*- SP) É aquele que fornece os serviços que o usuário efetivamente deseja utilizar, por exemplo, *e-mail*, *e-commerce* e outros. O provedor de serviços pode delegar a autenticação dos usuários que acessam seus serviços a um IdP e, normalmente, fica encarregado da autorização.

Para tornar os processos mais eficientes e seguros e permitir características interessantes como *Single Sign-On* (SSO), diferentes modelos de gerência de identidades são definidos na literatura. Cada modelo apresenta suas vantagens e desvantagens e tem sua área de aplicação específica.

Lee, Jeun e Jung (2009) consideram os seguintes modelos de sistemas de gerência de identidades:

Silo O modelo mais comum e mais fácil de ser implementado. O próprio provedor de serviços gerencia as identidades utilizadas e que são válidas apenas no seu domínio de serviço. Nesse caso o usuário deve ter uma identidade diferente para cada provedor de serviços;

Centralizado Nesse modelo um único provedor de identidades gerencia identidades que podem ser usadas em diferentes serviços providos num mesmo domínio. É o modo mais simples de se implementar o SSO;

Federado No modelo federado o IdP compartilha as identidades entre provedores de serviço incluídos num círculo de confiança. Esse círculo é montado através de acordos prévios entre os provedores de serviços. Esse modelo também provê SSO, com a vantagem de estender esse serviço para organizações em diferentes domínios;

Centrado no usuário Nesse último modelo o próprio usuário gerencia as políticas de uso de suas identidades e informações, bem como controla a criação, uso e remoção dessas mesmas informações.

Segundo o padrão ITU-T X.1250, uma federação de identidades é "uma associação composta por qualquer número de provedores de serviços e provedores de identidades" (ITU-T, 2009a). A gerência de identidades federadas (*Federated Identity Management* - FIM) tem as vantagens de permitir o SSO entre diversos domínios, mover a carga da gerência de atributos e credenciais de usuários para os IdPs e prover escalabilidade.

De acordo com Chadwick (2009), a confiança está implícita na definição de federação. Com a gerência de identidades federadas, espera-se que todos os participantes da federação concordem que as informações recebidas de outros participantes são corretas e confiáveis, no que é conhecido como círculo de confiança (*Circle of Trust* - CoT).

A gerência de identidades federadas, no entanto, ainda enfrenta alguns problemas. Entre esses problemas, destaca-se o fato de que a formação do círculo de confiança requer uma negociação prévia, o que pode se tornar um processo extensivo e dificultar a colaboração dinâmica. Existem propostas para estender os padrões atuais para que possam suportar esse tipo de colaboração, como em Cabarcos et al. (2009) e em Olshansky (2008).

Outro problema enfrentado por federações de identidades é o extenso número de protocolos e padrões, o que acaba por reduzir a interoperabilidade. Federações tendem a se tornar cada vez maiores e usuários podem participar de diferentes federações. Todos esses fatores combinados levam à, na prática, uma escalabilidade reduzida da federações de identidade, reduzindo sua eficácia em aplicações reais (LAMPROPOULOS; DENAZIS, 2012).

Nesse momento, é importante diferenciar os conceitos de federação de identidades e federação de nuvens. Uma federação de nuvens tem como objetivo compartilhar os recursos virtuais de diversos CSPs, enquanto uma federação de identidades tem como objetivo tornar possível o uso de uma mesma identidade em diferentes domínios. Os requisitos de confiança necessários ao estabelecimento de uma federação de nuvens e de uma federação de identidades também são diferentes.

2.2.2 Controle de Acesso

De acordo com Samarati e Vimercati (2001), controle de acesso é o processo pelo qual se garante que todo acesso à informações ou recursos em um sistema computacional é controlado e somente acessos devidamente autorizados podem ocorrer. Esse processo é determinado por um modelo que especifica as regras que serão utilizadas na avaliação das requisições que chegam ao sistema.

Num sistema de controle de acesso existem três conjuntos de entidades principais:

Sujeitos São as entidades que pretendem utilizar os recursos e, para isso, realizam as requisições de acesso. Também são conhecidos como *principals*;

Recursos São os objetos acessados pelos sujeitos. Representam informações, dados e recursos necessários aos sujeitos; e

Ações São as operações realizadas pelos sujeitos sobre os recursos e representam a forma de acesso desejada. Essas ações podem ou não alterar o estado de um ou mais recursos ou do sistema, bem como afetar características como disponibilidade, confidencialidade e integridade dos recursos.

Um sistema de controle de acesso é composto por políticas e mecanismos. As políticas são descrições do que é permitido ou não no sistema, ou seja, representam um conjunto de comportamentos aceitáveis. Políticas podem ser descritas de formas legíveis por humanos ou por máquinas, sendo

que políticas descritas em linguagens legíveis por máquinas, como *eXtensible Markup Language* (XML) têm a vantagem de facilitar a automação dos processos de autorização. Os mecanismos são os procedimentos implementados pelo sistema que forçam o cumprimento das políticas, ou seja, as funções de *hardware* e *software* de baixo nível que implementam os controles exigidos pelas políticas.

Os modelos de controle de acesso mais tradicionais e historicamente mais relevantes são o *Discretionary Access Control* (DAC), o *Mandatory Access Control* (MAC) e o *Role-based Access Control* (RBAC) (SAMARATI; VIMERCATI, 2001).

O DAC, ou controle de acesso discricionário, é uma política de acesso determinada pelo dono do recurso. O próprio dono decide quem tem acesso aos recursos e que tipo de acesso pode ter. Segundo o *Trusted Computer System Evaluation Criteria* (TCSEC) (Department of Defense, 1985), também conhecido como *The Orange Book*, o controle de acesso discricionário é, em tradução livre:

> "um modo de restringir o acesso a objetos baseado na identidade dos sujeitos e/ou grupos aos quais pertencem. Os controles são discricionários no sentido de que um sujeito com uma certa permissão de acesso é capaz de delegar essa permissão (talvez indiretamente) para qualquer outro sujeito (a menos que seja restringido pelo controle de acesso obrigatório)."

A principal proposta de controle de acesso discricionário é o modelo de matriz de acesso, proposto por (LAMPSON, 1974). A definição formal do modelo é a de uma tupla (S, O, A). Nessa tupla, S é um conjunto de sujeitos; O é um conjunto de objetos; e A é a matriz de acesso, onde as colunas representam os objetos, as linhas representam os sujeitos e uma entrada $A[s, o]$ representa os privilégios do sujeito s sobre o objeto o.

Apesar de conceitualmente importante, a matriz de acesso não é eficientemente implementada na prática, porque tende a ser uma matriz enorme e bastante esparsa. Por isso, existem três abordagens de implementação da matriz de acesso:

Tabela de autorização As entradas preenchidas da matriz de acesso são armazenadas em uma tabela com três colunas: sujeitos, ações e objetos. Cada tupla na tabela corresponde a uma autorização;

***Access Control List* (ACL)** Nas listas de controle de acesso, a matriz de acesso é armazenada por coluna. Cada objeto é associado a uma lista indicando, para cada sujeito, as ações que ele pode realizar sobre o objeto; e

Capabilities A matriz é armazenada por linhas. Cada usuário tem associada a si uma lista que indica, para cada objeto, os acessos que ele pode realizar.

No MAC, ou controle de acesso obrigatório, a política é determinada pelo sistema, através da valoração das informações e da sua classificação em níveis de sensibilidade. Ainda segundo (Department of Defense, 1985) o MAC é:

> "uma forma de restringir o acesso a objetos baseado na sensibilidade (representada por um rótulo) da informação contida nos objetos e da autorização formal (*clearance* ou nível de habilitação) dos sujeitos para acesso a informações de tal sensibilidade."

Os principais modelos que implementam o MAC são: Bell-LaPadula (BELL; LAPADULA, 1973), que enfatiza a confidencialidade; e Biba (BIBA, 1977), que enfatiza a integridade, além do modelo Clark-Wilson (CLARK; WILSON, 1987).

O RBAC, ou controle de acesso baseado em papéis, apareceu pela primeira vez em Ferraiolo e Kuhn (1992) e posteriormente foi padronizado por Sandhu, Ferraiolo e Kuhn (2000). Nesse modelo os sujeitos são representados por papéis - como convidado, usuário e administrador - e as permissões de acesso a recursos são definidas com base nesses papéis.

Esse modelo é atualmente amplamente utilizado em produtos como sistemas operacionais e sistemas de gerenciamento de bancos de dados. O modelo RBAC na verdade engloba diversos modelos, sendo os principais o RBAC Básico, o RBAC Hierárquico, a Separação Estática de Deveres e a Separação Dinâmica de Deveres.

Recentemente as necessidades do controle de acesso vêm sendo alteradas com o surgimento de novas arquiteturas de sistemas, como sistemas distribuídos e sistemas *Web*. Entre esse novos modelos podemos destacar o *Attribute Based Access Control* (ABAC), o *Usage Control* (UCON) e os modelos dinâmicos.

ABAC, ou controle de acesso baseado em atributos, é um modelo de controle de acesso que tem como premissa a avaliação de requisições de acesso baseando-se em atributos fornecidos pelo sujeito.

Segundo Damiani, Vimercati e Samarati (2005), o ABAC pode atender aos requisitos de privacidade, anonimidade, expressividade e possíveis restrições.

Nos modelos de controle de acesso tradicionais a autorização é baseada na identidade do sujeito requisitando acesso a um recurso. Isso funciona perfeitamente quando ambos se encontram em um mesmo domínio de

segurança. No entanto, com a evolução dos sistemas distribuídos e, principalmente, com a Internet, atualmente é mais comum que sujeitos e recursos pertençam a domínios diferentes. Por isso, é mais apropriado que a decisão de controle de acesso seja baseada nos atributos, que são propriedades intrínsecas do sujeito. Essa abordagem permite uma maior flexibilidade e escalabilidade.

Nesse modelo um usuário que solicita acesso a um recurso não precisa provar sua identidade para o provedor de serviços, apenas mostrar atributos que comprovem que ele pode acessar aquele recurso, como um *token* assinado por um provedor de identidades confiável (onde o usuário foi previamente autenticado). Por exemplo, em um *site* de comércio eletrônico em que os usuários precisam ser maiores de idade, um usuário que quer acessar o recurso não precisa informar sua identidade ao sistema, apenas mostrar um atributo de idade que venha assinado pelo seu provedor de identidades, considerando que esse provedor é confiado pelo *site*.

Na verdade, nesse modelo o sujeito é substituído por um conjunto de atributos. Para implementar esse modelo, uma ideia que vem sendo bastante utilizada é a de certificado de atributos. Certificados de atributos são certificados digitais que contém informações sobre atributos de um sujeito e são ligados a um certificado de chave pública que identifica esse sujeito (FARRELL; HOUSLEY, 2002).

O ABAC também é conhecido por *Policy-based Access Control* (PBAC), ou controle de acesso baseado em políticas, devido a sua dependência nas políticas de acesso definidas. As políticas de acesso descrevem as regras a partir das quais o sistema toma as suas decisões de controle de acesso. Um exemplo de política seria liberar acesso a todos os usuários que apresentem o atributo idade maior que 18 anos e negar a todos em caso contrário.

UCON, ou controle de uso, é um modelo de controle de acesso e uso de recursos proposto por Park e Sandhu (2002). É um modelo baseado em atributos e tem como objetivo agregar os recursos do controle de acesso tradicional, do gerenciamento de direitos digitais (*Digital Rights Management* - DRM) e do gerenciamento de confiança.

Segundo Park e Sandhu (2004), o termo controle de uso é uma generalização de controle de acesso para englobar autorizações, obrigações, condições, continuidade e mutabilidade. Obrigações são requisitos que devem ser cumpridos pelos sujeitos para a obtenção do acesso; condições são requisitos do sistema ou do ambiente e independentes de sujeitos e objetos; continuidade, ou controle contínuo, é a característica de avaliar a autorização mesmo durante o acesso e mutabilidade representa a característica de que os atributos de um sujeito são mutáveis e podem ser alterados com o acesso.

Um exemplo de controle contínuo é a necessidade de um usuário clicar em uma propaganda em um dado intervalo de tempo para manter seu acesso e um exemplo de mutabilidade é um atributo de contagem de número de acessos, que pode ser atualizado após um acesso.

O modelo UCON trata do problema do acesso à recursos digitais em diferentes sistemas e a partir de diferentes dispositivos, o que acontece com a computação pervasiva e, de acordo com Park e Sandhu (2004), o objetivo do controle de uso é fornecer uma nova base intelectual para o controle de acesso, substituindo as várias extensões ao modelo de matriz de acesso que foram sendo feitas ao longo das décadas.

A expressividade do modelo UCON, porém, vem ao custo de uma alta complexidade, especialmente na implementação (SUHENDRA, 2011).

2.3 CONTROLE DE ACESSO DINÂMICO

Apesar de ameaças como divulgação não autorizada, negação de serviço e alteração de informações ainda serem extremamente críticas, novos tipos de sistemas, caracterizados pela sua distribuição, pela necessidade de reconfiguração automática e pela dinamicidade dos recursos e usuários, apresentam novos desafios para os modelos de controle de acesso. Modelos como computação pervasiva e ubíqua, grades computacionais e computação em nuvem se encaixam nessas necessidades (ZHANG; PARASHAR, 2003, 2004; FALL et al., 2011).

Um dos problemas dos modelos de controle de acesso citados anteriormente é que são estáticos do ponto de vista de autorização, ou seja, todas as decisões já são preestabelecidas, baseadas nas políticas a que o sistema se submete.

A ideia principal dos sistemas de autorização ou controle de acesso dinâmicos é que cada requisição de acesso deve ser analisada no contexto em que ocorre, de maneira dinâmica, levando em conta não somente as políticas pré-definidas do sistema, mas também informações de contexto como o risco da operação, a necessidade do usuário em realizar aquela operação, o benefício da operação para o sistema e para o usuário, entre outras.

Em aplicações reais, muitas vezes situações inesperadas requerem a violação de políticas de acesso. Isso ocorre, entre outros motivos, porque as políticas de acesso podem ser incompletas e não prever situações legítimas ou mesmo serem conflitantes entre si.

Dois exemplos recorrentes na literatura e que exemplificam necessidades excepcionais de acesso ocorrem em ambientes médicos e militares. Na primeira situação, uma enfermeira precisa acessar o histórico de um pa-

ciente, com acesso originalmente restrito a médicos, para salvar a vida do mesmo. Na segunda situação, um soldado em campo de batalha precisa acessar informações restritas a oficiais para a conclusão de uma missão.

Essas situações excepcionais são citadas por alguns autores como *"break the glass"*, em referência ao fato de que, para que sejam aceitas, é necessário burlar o sistema de autorização. Segundo Brucker e Petritsch (2009), o *"break the glass"* é uma abordagem para prover um suporte flexível à políticas, prevenindo uma estagnação do sistema que poderia levar a prejuízos.

Ao analisar as situações de exemplo e outras semelhantes sob a ótica dos modelos de controle de acesso tradicionais, a requisição de acesso seria negada ou, para ser liberada, precisaria da intervenção manual de um administrador do sistema, o que poderia acarretar na liberação de permissões desnecessárias ao usuário ou no aumento da exposição do recurso, através de ações como a diminuição da classificação do recurso em um sistema MAC ou a elevação do papel do usuário em um sistema RBAC.

Em um sistema de controle de acesso dinâmico essa requisição seria excepcionalmente liberada apenas em uma situação, caso o risco fosse aceitável para o sistema, e as ações realizadas pelo usuário durante o acesso excepcional seriam monitoradas.

Brucker e Petritsch (2009) argumentam que utilizar controle de acesso baseado em risco e *"break the glass"* são abordagens independentes e mutuamente benéficas.

Os modelos de controle de acesso dinâmicos são caracterizados pelo uso de uma função de cálculo em "tempo real" para cada requisição de acesso e essa função é a principal característica que diferencia os modelos entre si.

Entre as características mais utilizadas para a tomada de decisão em um modelo de controle de acesso dinâmico estão o risco do acesso, a necessidade do usuário em obter o acesso, o benefício do acesso, a confiança do sistema no usuário e o contexto em que a requisição de acesso ocorre - local, dispositivo, horário, forma de acesso, entre outros.

Nos modelos de controle de acesso dinâmico, quando um sujeito requisita acesso a um recurso, essa requisição é processada por uma função de avaliação que usa todas as informações que julgar necessárias para chegar a uma decisão de acesso, como ilustrado na Figura 1.

A natureza dinâmica do controle de acesso é capturada nesses modelos porque as decisões de acesso podem variar de acordo com informações de contexto avaliadas no momento da requisição.

Nesses modelos, a liberação de um pedido de acesso geralmente envolve alguma forma de monitoração pelo sistema, que pode ser na forma de: obrigações (*obligations*), que são pós-condições que um usuário deve cum-

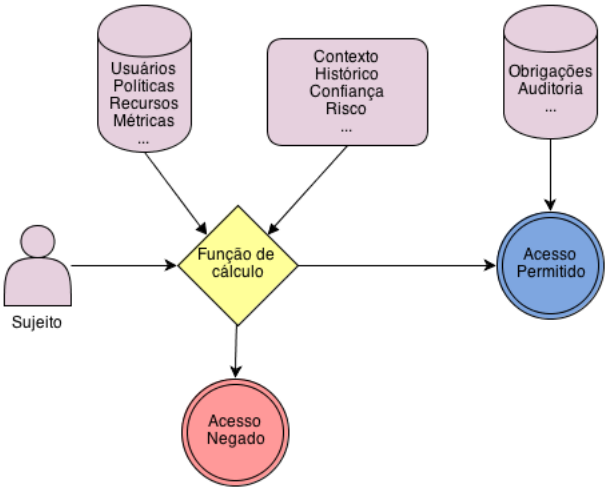

Figura 1: Visão geral dos modelos de controle de acesso dinâmicos

prir a fim de manter seu direito de acesso (SUHENDRA, 2011); um sistema de reputação que registra as ações dos usuários e atribui recompensas e penalidades aos mesmos (SHAIKH; ADI; LOGRIPPO, 2012) ou um sistema de mercado, em que os usuários tem uma quantidade limitada de pontos que podem ser usados para "comprar" acessos excepcionais (MOLLOY; CHENG; ROHATGI, 2008).

2.4 CONTROLE DE ACESSO BASEADO EM RISCO

Risco pode ser considerado como o dano potencial que pode surgir em um processo atual ou futuro e é geralmente representado pela probabilidade de ocorrência de um evento indesejado e o seu impacto resultante (DIEP et al., 2007).

Segundo Peterson (2006), métricas de risco são uma forma de quantificar os ativos, ameaças e vulnerabilidades de um sistema e risco é diferente de incerteza porque pode ser medido e administrado.

Análise de risco é o processo de mapear os riscos relacionados aos ativos considerados no escopo, analisando variáveis como ameaças, probabilidade de ocorrência das ameaças e impacto da ocorrência do risco. Análise de risco é uma parte do processo de gerência de risco e é uma ferramenta fundamental para a tomada de decisões em áreas tão diversas quanto economia e

segurança da informação.

Os modelos de controle de acesso baseado em risco realizam uma análise de risco na requisição de acesso para chegar a uma decisão de acesso.

A análise de risco se divide em duas categorias principais: qualitativa e quantitativa. Nos métodos qualitativos, diferentes escalas de risco são utilizadas e geralmente a valoração se dá através da opinião de um especialista, enquanto os métodos quantitativos utilizam uma maneira de atribuir um valor numérico que representa o risco de uma requisição de acesso e buscam fazê-lo de forma automatizada.

Em métodos quantitativos, o risco de um evento é geralmente representado por $R = P \times I$, onde P é a probabilidade de ocorrência do evento e I é o impacto da ocorrência do evento. Em situações onde há um histórico de acessos e onde o impacto pode ser quantificado, especialmente em valores monetários, o cálculo torna-se mais fácil, mas há situações onde esses itens não são facilmente obtidos ou onde se deseja considerar também outras características e, para isso, existem as diversas propostas de cálculo de risco.

Entre os modelos baseados em risco, o termo *Risk-adaptive Access Control* (RAdAC), ou controle de acesso adaptável ao risco, é bastante utilizado. Esse modelo originou-se na *National Security Agency* (NSA) e é descrito pelo JASON Program Office (2004), em um trabalho pioneiro na área. O trabalho propõe o uso do controle de acesso baseado em risco para um compartilhamento de informações mais eficaz em ambientes militares, descrevendo o cenário de controle de acesso à informações no domínio considerado, explicitando seus principais problemas e propondo a utilização de uma valoração de risco para o controle de acesso. O trabalho, entretanto, não entra em detalhes sobre como realizar a valoração do risco.

O trabalho citado e outros subsequentes utilizam os conceitos de "necessidade operacional" - a necessidade que uma entidade tem de acessar certa informação para poder completar uma missão - e "risco de segurança" - o valor a ser definido que representa o dano ao sistema caso a informação seja liberada - para a decisão de acesso. O acesso é liberado caso a necessidade operacional seja maior que o risco de segurança. Choudhary (2005) propõe que no cálculo do risco entrem componentes como características de pessoas, características de componentes de TI, características dos objetos, fatores ambientais, fatores situacionais e heurísticas.

Britton e Brown (2007) apresentam um método de quantificação para o modelo RAdAC baseado na opinião de especialistas. Uma lista de fatores de risco é compilada e um valor atribuído a cada fator. Posteriormente pesos são atribuídos a cada valor e o resultado final é a combinação de todos os fatores com os seus pesos.

Farroha e Farroha (2012e) descrevem alguns desafios práticos para se

atingir a implementação do modelo RAdAC, entre eles o cálculo em tempo real do risco de segurança para cada decisão de acesso; a determinação do risco operacional; a quantificação do nível de confiança; o uso de heurísticas para atingir as decisões de acesso e a possibilidade de revogação do acesso a qualquer momento.

Cheng et al. (2007) e Ni, Bertino e Lobo (2010) propõem o uso de lógica difusa (*fuzzy*) para o cálculo dos valores de risco. Molloy et al. (2012) sugerem a decisão de acesso baseada em técnicas de aprendizagem de máquina, especificamente classificadores treinados com um histórico de decisões. Shaikh, Adi e Logrippo (2012) demonstram dois métodos de cálculo de risco que consideram o histórico dos usuários. Wang e Jin (2011) apresentam uma aplicação do modelo em um contexto de proteção à privacidade em sistemas de saúde.

Em resumo, o grande problema da valoração do risco é a incerteza das informações e muitos trabalhos apresentam formas de tentar contornar essa incerteza para o cálculo do risco. Além dos trabalhos aqui citados, existem outros que utilizam métodos como redes *Bayesianas*, inferência probabilística, lógica difusa, aprendizagem de máquina e teoria da decisão para a valoração do risco. Isso mostra que existe uma grande diversidade de formas de se abordar essa valoração e cada abordagem, naturalmente, apresenta vantagens e desvantagens.

A principal crítica aos modelos baseados em risco é que a valoração de risco é um processo eminentemente subjetivo e baseado em opiniões, sendo muito afetado por quem realiza a análise.

Alguns trabalhos se preocupam em analisar as relações entre o controle de acesso baseado em risco e outros modelos. Kandala, Sandhu e Bhamidipati (2011) discutem as relações do RAdAC com o UCON. Celikel et al. (2009) realiza uma análise de risco para bancos de dados que utilizam RBAC e Baracaldo e Joshi (2012) estende o RBAC através da incorporação de um processo de análise de risco.

Alguns modelos levam em conta não apenas o risco de uma requisição de acesso, mas também o benefício que esse acesso pode trazer aos usuários e ao sistema. Han et al. (2012), inclusive, argumenta que o risco e o benefício são características intrínsecas de qualquer acesso a informações sensíveis, não importando se o modelo de controle de acesso as considera ou não. No trabalho citado, os autores propõem o uso do cálculo de benefício e risco para o controle de acesso e mostram quatro variantes de aplicações, além de sugerir uma extensão do XACML para suportar a proposta.

O modelo *Benefit And Risk Access Control* (BARAC), ou controle de acesso com benefício e risco, foi proposto por Zhang, Brodsky e Jajodia (2006). Esse modelo associa vetores de risco e benefício para toda ação de

leitura ou escrita em um recurso e o sistema é responsável por manter sempre um estado em que o benefício supere o risco.

O fato de agregar uma nova característica além do risco ao cálculo de controle de acesso permite ao modelo apresentar uma visão mais rica do contexto de acesso, mas adiciona um novo nível de subjetividade e uma nova dificuldade na quantificação e na tomada de decisões de acesso.

Outros modelos utilizam outras características para atingir o controle de acesso dinâmico, sendo as principais a confiança (LI et al., 2008) e o contexto, como nos trabalhos de Diep et al. (2007), Ahmed e Zhang (2010) e Dimmock (2003).

2.5 ARQUITETURAS DE CONTROLE DE ACESSO

Com o aumento da complexidade dos sistemas que utilizam controle de acesso, tornou-se necessário criar arquiteturas de referência para protocolos de autorização, especialmente para sistemas distribuídos.

Uma série de três *Request for Comments* (RFC) da *Internet Engineering Task Force* (IETF) (VOLLBRECHT et al., 2000b, 2000a; FARRELL et al., 2000) define a arquitetura, os requisitos e dá exemplos de autorização para sistemas na Internet. O RFC2904 (VOLLBRECHT et al., 2000b), especialmente, define os pontos principais da arquitetura de autorização, a saber:

***Policy Retrieval Point* (PRP)** O ponto da arquitetura a partir do qual o sistema recupera as políticas de acesso que serão usadas pelo PDP;

***Policy Information Point* (PIP)** O ponto em que o sistema recupera informações relacionadas ao usuários que serão comparadas com as políticas para alcançar as decisões de acesso;

***Policy Decision Point* (PDP)** O ponto que efetivamente realiza a decisão de controle de acesso, com base nas políticas e informações recuperadas; e

***Policy Enforcement Point* (PEP)** O ponto de acesso do usuário ao sistema. É o ponto que protege um recurso sensível, recebendo a requisição de acesso e enviando-a ao PDP.

Baseado nessa arquitetura a *Organization for the Advancement of Structured Information Standards* (OASIS) criou o padrão *eXtensible Access Control Markup Language* (XACML). XACML se encontra atualmente na versão 3.0 e é "uma linguagem de políticas de controle de acesso de propósito geral" (OASIS, 2003).

A linguagem dá suporte à criação tanto de políticas quanto de requisições e respostas de acesso em XML.

O padrão XACML estabelece não somente uma linguagem para definição de políticas de controle de acesso, mas também uma arquitetura distribuída para sistemas de controle de acesso baseada no RFC2904. Essa arquitetura utiliza também quatro pontos, com a diferença de que chama o PRP de *Policy Administration Point* (PAP).

Diversas outras linguagens e arquiteturas de autorização foram propostas e implementadas em cenários específicos, mas o XACML apresenta algumas vantagens, como: ser um padrão internacional desenvolvido por uma comunidade; ser genérico e poder se adaptar a diversas situações diferentes; ser distribuído por natureza; e ser bastante abrangente, com suporte a vários tipos de dados e funções e possibilidade de integração com mecanismos como *Lightweight Directory Access Protocol* (LDAP) e SAML.

Por esses e outros motivos, XACML vem sendo bastante utilizado em aplicações industriais, científicas e acadêmicas e possui muitas implementações livres. Entre essas implementações podemos destacar: Sun XACML, Enterprise Java XACML e HERASAF (OASIS, 2013a).

A linguagem XACML é composta de diferentes elementos. O elemento raiz de uma política XACML é um *Policy* ou um *PolicySet*, que pode conter outros *Policies*, *PolicySets* ou referências a políticas remotas. Um *Policy* representa uma política de controle de acesso, expressa através de um conjunto de regras, chamadas *Rules* (OASIS, 2003).

Cada *Rule* é avaliado individualmente dentro de um *Policy* e cada *Policy* é avaliado individualmente dentro de um *PolicySet*. O resultado da avaliação de uma regra ou política pelo PDP é *Permit*, *Deny*, *Indeterminate* ou *NotApplicable*. *Permit* representa uma situação em que a permissão de acesso é concedida, *Deny* quando é negada, *Indeterminate* quando ocorre algum erro ou há falta de algum atributo e *NotApplicable* significa que a requisição não pode ser respondida pelo serviço solicitado.

Como um *PolicySet* pode conter múltiplas políticas e um *Policy* pode conter muitas regras que podem ser avaliadas com decisões de acesso diferentes, é necessário que haja um modo de agrupar essas decisões em um resultado único.

Isso é feito através de algoritmos de combinação. Existem algoritmos de combinação de regras e algoritmos de combinação de políticas. Alguns dos algoritmos mais importantes são (IBM, 2012) (OASIS, 2013b):

Deny Overrides Se alguma política for avaliada como *Deny*, o resultado final será *Deny*;

First Applicable O resultado da primeira política avaliada que seja aplicável

ao recurso é o resultado final da combinação;

Only One Applicable Apenas uma política pode ser avaliada aplicável ao recurso e o resultado dessa avaliação é o resultado final da combinação;

Permit Overrides Se alguma política for avaliada como *Deny*, o resultado final será *Deny*.

2.6 IAM EM COMPUTAÇÃO EM NUVEM

Como comentado na seção 2.1.5, gerência de identidades e controle de acesso é um dos domínios fundamentais da segurança em computação em nuvem. Quando se discute controle de acesso em nuvem, é necessário levar em conta o grande número de usuários e recursos disponíveis. O serviço Amazon S3, por exemplo, tem mais de 2 trilhões de objetos armazenados e processa mais de 1 milhão de requisições de acesso por segundo (BARR, 2013).

A pesquisa em controle de acesso e autorização para computação em nuvem tem ganhado bastante força nos últimos anos e muitos modelos e sistemas foram propostos para tratar de diferentes aspectos e características da nuvem (ALMUTAIRI et al., 2012; GHAZIA; MASOOD; SHIBLI, 2012; GOUGLIDIS; MAVRIDIS, 2010). Essa pesquisa é importante porque as principais implementações e os principais provedores de serviços de nuvem ainda utilizam modelos de controle de acesso simples, que não atendem a demandas mais complexas de autorização e federação (BERNABE et al., 2012).

Uma característica bastante explorada sobre autorização em nuvem é a *multi-tenancy*, ou arquitetura multi-inquilinos (CALERO et al., 2010; BERNABE et al., 2012; LEANDRO et al., 2012). Essa arquitetura é uma das bases da computação em nuvem e explora a possibilidade de diversos usuários compartilharem os mesmos equipamentos e, muitas vezes, os mesmos bancos de dados e aplicações.

Em geral os modelos de controle de acesso em nuvem se dividem entre os que dependem de criptografia e os que não dependem. Essa divisão ocorre, entre outros motivos pela modelagem de ameaças. Geralmente modelos que dependem de criptografia consideram o provedor de serviços de nuvem como uma ameaça aos dados dos usuários, enquanto modelos que não dependem de criptografia consideram que o CSP é confiável, especialmente em casos de nuvens privadas (CHADWICK; FATEMA, 2012).

3 TRABALHOS RELACIONADOS

Nesse capítulo são apresentados e discutidos os principais trabalhos relacionados, além de outros que se situam em um contexto próximo, mas apresentam objetivos diferentes.

A descrição dos trabalhos é dividida em duas seções. Primeiramente abordamos os trabalhos que lidam com controle de acesso para computação em nuvem (seção 3.1) e, em seguida, descrevemos brevemente alguns trabalhos relacionados ao conceito de federação de nuvens (seção 3.2).

Uma comparação entre o trabalho desenvolvido e os principais trabalhos relacionados é apresentada no capítulo 6.

3.1 CONTROLE DE ACESSO

Computação em nuvem e controle de acesso são dois temas de pesquisa bastante relevantes e, por isso, na literatura é possível encontrar diversos trabalhos que lidam com controle de acesso para computação em nuvem. Entre esses, podem-se destacar: Almutairi et al. (2012), Chadwick e Fatema (2012), Chow et al. (2009) e Yu et al. (2010).

No entanto, para limitar o número de trabalhos descritos, considera-se como trabalho relacionado nesse capítulo aqueles que utilizam o conceito de controle de acesso baseado em risco para computação em nuvem, descritos nas seções 3.1.1, 3.1.2 e 3.1.3.

A relação entre controle de acesso baseado em risco e computação em nuvem ainda não foi largamente explorada, por isso o número de trabalhos relacionados é restrito. Os trabalhos mais importantes que podemos citar são: Fall et al. (2011), Arias-Cabarcos et al. (2012) e Sharma et al. (2012).

3.1.1 *Toward Quantified Risk-Adaptive Access Control for Multi-tenant Cloud Computing*

O trabalho de Fall et al. (2011) é focado nos problemas para autorização criados pelo uso de arquiteturas multi-inquilino na nuvem. Os autores argumentam que modelos de controle de acesso tradicionais são estáticos e, por isso, não são adequados para o uso em ambientes dinâmicos como a nuvem, enquanto um modelo baseado em risco é dinâmico e naturalmente adaptado a esse tipo de ambiente.

No trabalho citado, é proposta a utilização do modelo RAdAC, com

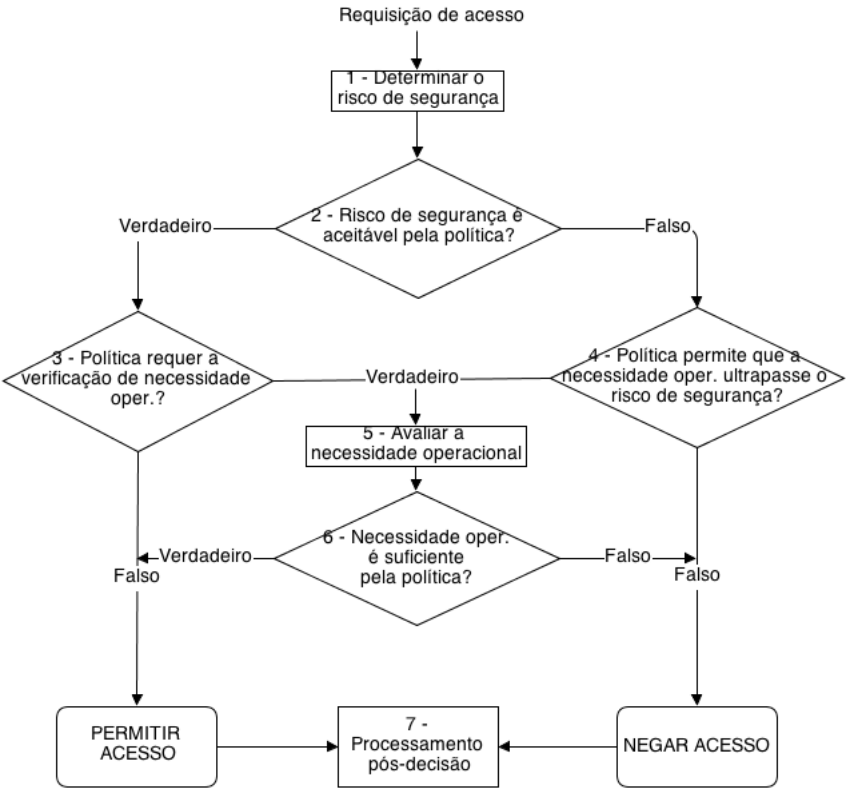

Figura 2: Modelo RAdAC (adaptado de Fall et al. (2011))

a quantificação de alguns componentes de risco através do uso de técnicas de aprendizagem de máquina. Como o trabalho é focado na característica de multi-inquilinos da nuvem, as situações de risco identificadas pelos autores são: o risco de acesso ilegal de dados de um inquilino por outro; o risco de um inquilino expor seus dados inadvertidamente; o risco de administradores acessarem os dados de inquilinos para benefício próprio; e o risco de administradores exporem os dados de inquilinos inadvertidamente.

A Figura 2 mostra o modelo RAdAC como apresentado no artigo descrito.

A principal contribuição do trabalho é introduzir o conceito de controle de acesso baseado em risco para computação em nuvem, no entanto apenas uma discussão da ideia é apresentada, com planos de pesquisa futura. Nenhum tipo de implementação, simulação ou validação é apresentada no

trabalho.

3.1.2 *A Metric-Based Approach to Assess Risk for "On Cloud" Federated Identity Management*

O trabalho de Arias-Cabarcos et al. (2012) descreve as dificuldades atuais em gerência de identidades federadas em nuvem e apresenta como proposta utilizar a avaliação de risco como um método para possibilitar a construção de federações de identidades dinâmicas na nuvem.

De acordo com os autores a computação em nuvem é uma revolução importante na Internet, mas um dos problemas fundamentais para a sua adoção é a necessidade de melhores sistemas de gerência de identidades e controle de acesso, sendo a gerência de identidades federadas identificada como um aspecto essencial da implantação das nuvens.

Os autores identificam a necessidade de acordos de confiança como um obstáculo na formação de federações dinâmicas, necessárias para atingir um aumento de escala e, para contornar essa situação, propõem uma taxonomia para classificação de risco com a função de mitigar ameaças encontradas em decisões acerca de colaboração.

A taxonomia compreende duas fases: pré e pós-federação e, para cada fase, apresenta um conjunto de métricas. As principais métricas de segurança são: confidencialidade, integridade, autenticação, não-repudiação, disponibilidade, responsabilidade e privacidade. A Figura 3 mostra a taxonomia de risco descrita no artigo.

Para cada métrica os autores propõem que seja aplicada uma quantificação básica de risco, onde o valor do risco é igual a probabilidade de ocorrência do evento multiplicado pelo impacto da ocorrência desse evento ($R = P * I$). Depois de todas as métricas quantificadas é realizada uma agregação hierárquica.

Uma arquitetura genérica de agregação hierárquica é apresentada e os autores especificam a arquitetura com o caso de uma função de agregação *fuzzy*, mostrando os passos da sua aplicação.

Apesar de bastante detalhado, o trabalho não apresenta valores numéricos para as métricas ou como esses valores devem ser obtidos, apenas uma descrição semântica de cada métrica. Além disso, de acordo com a proposta apresentada, uma vez escolhidas as métricas e o método de agregação, essa informação se torna fixa no sistema, evitando que usuários possam informar suas próprias funções de cálculo ou agregação.

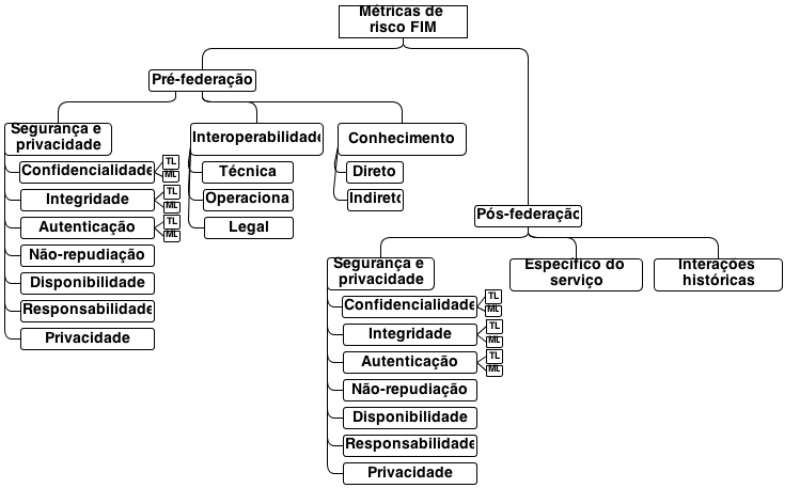

Figura 3: Taxonomia de risco (adaptado de Arias-Cabarcos et al. (2012))

3.1.3 *Using Risk in Access Control for Cloud-Assisted eHealth*

O trabalho de Sharma et al. (2012) apresenta um modelo de controle de acesso baseado em risco para aplicações médicas (*e-health*) na nuvem.

Segundo os autores, aplicações de *e-Health Cloud* são eficientes e econômicas, mas enfrentam problemas de privacidade e segurança devido ao número de plataformas e entidades envolvidas na nuvem. Um desses problemas é o uso do controle de acesso baseado em papéis, que não leva em conta incerteza e risco e não se adapta facilmente à característica dinâmica da nuvem.

O artigo apresenta um protótipo de implementação de um sistema de controle de acesso dinâmico usando risco baseado em requisitos de confidencialidade, integridade e disponibilidade e um protocolo de troca de mensagens para garantir interoperabilidade entre sistemas.

O sistema proposto é um sistema de informações médicas que permite o armazenamento de dados utilizando os serviços de nuvem da Amazon e que, para manter a confidencialidade e integridade dos dados, utiliza um controle de acesso baseado em tarefas e consciente do risco.

A Figura 4 mostra uma visão geral do modelo de controle de acesso proposto no artigo.

Em resumo, toda tarefa a ser realizada no sistema é enviada para a nuvem e um servidor na nuvem calcula um valor de risco associado àquela

Figura 4: Visão geral do modelo de controle de acesso (adaptado de Sharma et al. (2012))

tarefa. Os autores chamam o controle de acesso proposto de *Task-based Availability, Integrity and Confidentiality (AIC)*. O controle de acesso é implementado acima do RBAC e dividido em duas etapas: primeiramente os usuários se autenticam e recebem as permissões do seu papel associado e, em segundo lugar, as requisições são avaliadas com base no risco que apresentam para os dados.

O cálculo do valor de risco é baseado nas ações realizadas e possíveis resultados, probabilidade de risco e custo associado à disponibilidade, integridade e confidencialidade dos dados. Com o valor básico calculado, o sistema então busca o padrão de comportamento anterior do usuário e atribui um valor final de risco, que é comparado a um limite preestabelecido e, se for menor, o acesso é concedido; em caso contrário, o acesso é negado.

O processo de quantificação apresentado é bastante simples. As ações (criar, visualizar, modificar e remover) são classificadas de acordo com o tipo de dado que estão lidando e com o resultado que podem gerar. Três atributos são quantificados: o efeito em disponibilidade, confidencialidade e integridade. Os valores atribuídos são: 1 se a ação tem efeito na característica analisada ou 0 em caso negativo. A Figura 5 apresenta uma tabela com os valores atribuídos pelos autores a cada ação.

O valor de risco (RS - *Risk Score*) é então calculado de acordo com a fórmula:

$$RS = ((c_{aj} * P) + (c_{ij} * P) + (c_{cj} * P)) + RS_{past}$$

Nessa fórmula, c_{aj} é o custo do resultado j da ação a_j em termos de disponibilidade; c_{ij} em termos de integridade; c_{cj} em termos de confidencialidade; e P é a probabilidade de ocorrência do resultado j da ação a_j.

Um protótipo foi implementado em Java utilizando o cálculo de risco descrito e um banco de dados com informações sensíveis armazenado na infraestrutura da Amazon EC2. O protótipo foi avaliado com respeito ao desem-

Ação	Sensibilidade dos dados	Resultado	Valor de risco		
			A	I	C
Criar	Sensível / Não-sensível	Sessão da transação está cheia → nenhum registro é criado	1	1	0
Visualizar	Sensível	Dados não-cifrados → vazamento de informações	0	0	1
	Não-sensível	Registro muito grande → indisponível para visualização	1	0	0
Modificar	Sensível / Não-sensível	Sessão da transação está cheia → nenhuma modificação é realizada	1	1	0
Deletar	Sensível / Não-sensível	Sem backup → nenhuma deleção é feita	1	1	0

Figura 5: Tabela de valores de risco, adaptado de Sharma et al. (2012)

penho e se mostrou cerca de 10 vezes mais lento do que um sistema RBAC simples, mas segundo os autores isso é imperceptível aos usuários.

O artigo apresenta uma forma de quantificação de risco e um protótipo bem descritos. A forma de quantificação parece bastante simples, mas a forma de se obter a probabilidade de um dado resultado de uma ação, por exemplo, não é mostrada (apesar de ser fácil supor que isso possa ser obtido através de um histórico de ações).

3.1.4 Outros

O trabalho de Krautsevich et al. (2010) mostra um uso de análise de risco para a seleção de provedores de serviço no modelo SOA, o que também engloba computação em nuvem.

Vários trabalhos do *Department of Defense* (DoD), como Farroha e Farroha (2011), Farroha e Farroha (2012e), Farroha e Farroha (2012f), Farroha e Farroha (2012b), Farroha e Farroha (2012d), Farroha e Farroha (2012c) e, finalmente, Farroha e Farroha (2012a) citam a relação entre computação em nuvem e controle de acesso baseado em risco, mas sem entrar em detalhes de implementação.

3.2 FEDERAÇÃO DE NUVENS

Em (CELESTI et al., 2010b, 2010c, 2010d) e outros artigos subsequentes, uma arquitetura para federação de nuvens chamada de "Federação Horizontal" é proposta. Nessa arquitetura um componente chamado *Cross-Cloud Federation Manager* (CCFM) se integra em cada provedor de serviços de nuvem, permitindo-o participar da federação. O *Cross-Cloud Federation Manager* (CCFM) é composto de três subcomponentes, responsáveis por funções específicas dentro da federação.

Baseado nessa arquitetura, o trabalho de Celesti et al. (2010a) propõe um mecanismo de gerência de identidades federadas utilizando um provedor de identidades de uma terceira parte.

O trabalho de Bernstein et al. (2009) propõe uma arquitetura básica para a construção de uma Intercloud, baseada em interoperabilidade e com o propósito de atingir uma escala global, como a Internet. Essa arquitetura é composta por três elementos principais: o *Intercloud Root*, os *Intercloud Exchanges* e os *Intercloud Gateways*. O mesmo grupo, em Bernstein e Vij (2010), propõe um mecanismo de autenticação entre nuvens utilizando *Security Assertion Markup Language* (SAML).

O *toolkit* Cloudbus é uma coleção de componentes que visa criar um mercado global de computação em nuvem. Esse mercado é uma combinação de recursos e serviços de nuvens de diferentes provedores. Essa arquitetura é baseada num agenciador (*broker*) chamado *Market Maker* que tem como objetivo mediar os acessos entre recursos da nuvem e aplicações do usuário. O componente responsável pela federação nessa proposta é o Aneka (BUYYA; PANDEY; VECCHIOLA, 2009).

O trabalho de Coppola et al. (2012) apresenta o projeto Contrail, um arcabouço para a construção de federações de nuvens. O trabalho é focado nas questões de gerência de identidades e controle de acesso do projeto, especialmente autenticação e autorização. Também são descritas extensões para suportar o modelo de controle de acesso *Usage Control* (UCON) na arquitetura proposta.

Outras arquiteturas de federação de nuvens que também podem ser citadas são descritas por Rochwerger et al. (2009) e Buyya, Ranjan e Calheiros (2010)

Para uma visão geral do conceito de gerência de identidades em Intercloud, além de uma proposta de implementação desse conceito, ver Sriram (2013).

4 UM MODELO PARA CONTROLE DE ACESSO DINÂMICO BASEADO EM RISCO

Nesse capítulo é detalhada a proposta do modelo para controle de acesso. Inicialmente o modelo e seus componentes são descritos e depois um estudo de caso da sua aplicação é feito usando uma arquitetura de federação de nuvens computacionais.

A visão completa do modelo proposto, que engloba o modelo para controle de acesso e seu uso nas federações de nuvens é apresentado na Figura 10. Nas seções 4.1 e 4.2 buscamos apresentar os componentes do modelo em detalhes.

4.1 MODELO PARA CONTROLE DE ACESSO

O modelo de controle de acesso dinâmico escolhido para ser utilizado no trabalho é o controle de acesso baseado em risco. A escolha pelo risco ocorre porque a maior parte dos trabalhos que lidam com controle de acesso dinâmico utiliza risco e também porque esse é um conceito extensivamente estudado em segurança da informação e segurança em computação.

No capítulo 2 foram descritas diversas abordagens para o controle de acesso baseado em risco e o objetivo do modelo proposto é encontrar uma forma de agregar as características mais importantes dos diferentes métodos existentes, permitindo que provedores de serviços e usuários da nuvem definam o método que desejam utilizar, sempre respeitando seus requisitos mínimos de segurança.

O modelo utilizado é baseado no cálculo de métricas de risco e na agregação dessas métricas, como no trabalho de Arias-Cabarcos et al. (2012), mas com a inclusão do conceito de políticas de risco, que permitem ao dono do recurso e ao provedor de nuvens um maior controle sobre a flexibilidade da autorização. As políticas de risco são explicadas na seção 4.1.1.

Um ponto importante percebido durante a realização da pesquisa foi que o padrão XACML, bem difundindo tanto na academia quanto na indústria, é bastante eficaz e que é vantajoso mantê-lo como base, desenvolvendo uma extensão que possa lidar com o controle de acesso baseado em risco, adicionando a necessária flexibilidade ao modelo.

O modelo desenvolvido é uma extensão do XACML e a Figura 6 mostra uma visão geral do modelo proposto para controle de acesso. Os seguintes componentes foram adicionados ao padrão XACML:

Risk Engine É o componente chamado pelo PDP para processar o controle

de acesso baseado em risco. O *risk engine* é responsável por analisar e processar as políticas de risco associadas a um recurso e invocar os métodos de quantificação e agregação de risco descritos em cada uma. O *risk engine* é diferente para cada provedor de serviços de nuvem, pois nele estão implementados localmente os métodos de quantificação disponíveis naquele provedor. Caso o usuário deseje utilizar outros métodos, ele deve fornecer uma implementação desses métodos na forma de um *web service*, cujo endereço é informado na política de risco.

Risk Quantification Web Services São os *web services* responsáveis por quantificar o risco de cada requisição de acesso. Esses *web services* são implementados pelo usuário que deseja utilizá-los, seguindo as orientações fornecidas pelo provedor de serviços de nuvem. Cada *web service* é responsável por receber como entrada uma requisição de acesso encaminhada pelo *Risk Engine* e retornar como saída um valor numérico que represente a quantificação da métrica de risco de acordo com os quesitos que julgar necessários.

Risk policies As políticas de risco definem como o acesso baseado em risco deve ser avaliado para cada recurso. As políticas são descritas em detalhes na seção 4.1.1.

Além da adição desses componentes, são necessárias modificações no PAP e no PDP para suportarem o armazenamento e o acesso às novas políticas.

Em resumo, na extensão proposta, ao receber uma requisição de acesso o PDP pode realizar duas verificações em paralelo. Por um lado o PDP realiza a decisão de controle de acesso ABAC, com base nas políticas XACML relacionadas ao recurso. Por outro lado o PDP, em conjunto com o *Risk Engine* realiza uma avaliação de risco da requisição de acesso baseado nas políticas de risco.

Definimos que as avaliações de políticas de risco têm os mesmos quatro possíveis resultados que uma avaliação de política XACML (*Permit*, *Deny*, *NotApplicable* e *Indeterminate*). Ao final da avaliação das políticas, o PDP tem duas decisões de acesso, uma baseada no XACML e uma baseada no risco, e essas decisões podem ser incompatíveis: uma pode ser *Deny* e a outra *Permit*, por exemplo.

Torna-se, então, necessário combinar essas avaliações para se atingir um resultado final. Para isso definimos quatro métodos de combinação de políticas:

Deny Overrides Se alguma das avaliações for *Deny*, o resultado final é *Deny*;

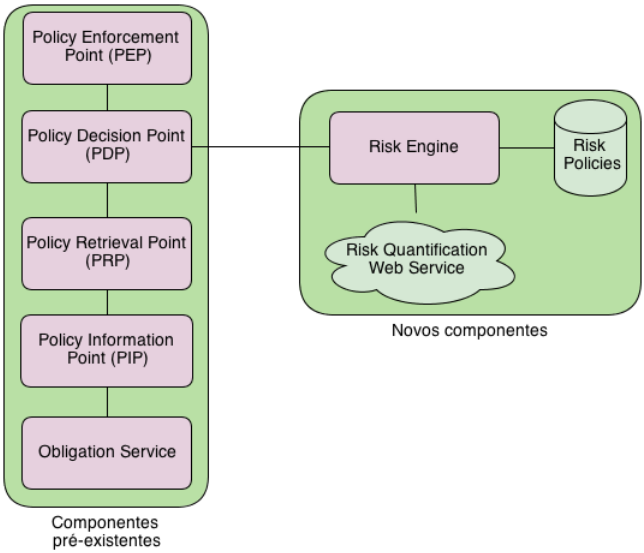

Figura 6: Visão geral do modelo para controle de acesso

Permit Overrides Se alguma das avaliações for *Permit*, o resultado final é *Permit*;

ABAC Precedence A avaliação XACML toma precedência e o resultado final é o mesmo da avaliação XACML;

Risk Precedence A avaliação de risco toma precedência e o resultado final é o mesmo da avaliação de risco;

Os métodos definidos são baseados nos algoritmos de combinação do XACML, os dois primeiros são exatamente iguais a dois métodos descritos no padrão (e comentado na seção 2.5) e os dois últimos, que fazem parte da proposta do modelo desse trabalho, surgem da necessidade de se dar prioridade à análise de risco ou à decisão XACML.

A decisão sobre qual combinação utilizar fica a cargo do CSP, que pode escolher dar mais peso ao risco ou ao XACML. Uma descrição mais completa do processo de decisão encontra-se na seção 4.1.3.

4.1.1 Políticas de risco

Uma política de risco é um arquivo XML que descreve para o provedor de serviços de nuvem como deve ser feito o controle de acesso baseado em risco para um determinado recurso. Esse arquivo é criado pelo usuário dono do recurso e fica armazenado no CSP responsável pelo recurso.

Cada política é composta pela identificação do recurso associado, identificação do dono do recurso, uma série de métricas de risco com suas descrições e métodos de quantificação, um método de agregação de risco e um limite de risco aceitável.

Métodos de quantificação são as funções utilizadas para dar um valor numérico a uma métrica de risco, baseado na requisição de acesso. Um exemplo é uma função que retorna como valor de risco "0" se a requisição utiliza protocolo *Hypertext Transfer Protocol Secure* (HTTPS) e "1" em caso negativo.

Um método de agregação é uma função que recebe os valores de risco calculados para cada métrica e os agrega em um único valor, que representa o resultado final. Alguns exemplos de métodos de agregação são: valor mínimo, valor máximo, média dos valores ou alguma função específica de um modelo, como o método *fuzzy* em Arias-Cabarcos et al. (2012).

Dois tipos de métodos de quantificação de risco são permitidos: locais ou externos. Os métodos locais invocam funções definidas no próprio *Risk Engine*, que processa a requisição de acesso encaminhada pelo PDP, enquanto os métodos externos definem o endereço de um *web service* que, ao ser invocado pelo *Risk Engine*, retorna um valor numérico representando a quantificação do risco.

A Figura 7 apresenta um exemplo de política de risco em que apenas uma métrica externa é calculada e o método de agregação utilizado é a escolha do maior risco.

A política de risco é criada pelo dono do recurso na criação do recurso e processada pelo *Risk Engine* no momento da decisão de acesso. As políticas de risco são administradas e obtidas a partir do PAP, do mesmo modo que as políticas XACML.

É importante salientar que o dono de um recurso pode sempre optar se o recurso pode ser acessado via cálculo de risco ou não, a fim de manter a flexibilidade do controle de acesso. O provedor do serviço de nuvem pode também optar se aceita que seus recursos sejam acessados dessa maneira. Caso o CSP aceite que seus recursos sejam acessados através da análise de risco, mas o usuário seja contrário a essa decisão, a decisão do usuário prevalece.

Além das políticas de risco definidas pelos usuários para cada recurso

```
<risk-ac>
    <resource id="1"/>
    <user id="2"/>
    <metric-set name="transport layer">
        <metric>
            <name>Transport Layer Encryption</name>
            <description>Quantifies the strength of the encryption scheme
            used in the access request</description>
            <quantification>https://example.com/quantify-tl-encryption
            </quantification>
        </metric>
    </metric-set>
    <aggregation-engine>maximum_value<aggregation-engine>
    <risk-threshold>10</risk-threshold>
</risk-ac>
```

Figura 7: Exemplo de política de risco

disponível, o CSP deve disponibilizar uma política básica de risco. A política básica é também um arquivo XML, mas definido pelo próprio CSP, e que contém as métricas mínimas de risco que o sistema exige, bem como o limite mínimo de risco aceitável.

As políticas básicas de cada CSP são avaliadas em todas as requisições de acesso antes das políticas específicas de cada recurso e, caso a política básica seja violada, as políticas específicas de um recurso não são processadas. Por isso, as políticas básicas de risco são uma ferramenta importante para manter os requisitos mínimos de segurança de um CSP, ao mesmo tempo permitindo flexibilidade no controle de acesso.

4.1.2 Especificação das políticas de risco

Nessa seção, buscamos descrever em detalhes o formato de uma política de risco em XML. A criação das políticas de risco é uma forma de tentar suportar o uso de diferentes métricas e métodos de quantificação de risco em um mesmo sistema. A escolha de utilizar XML para a definição das políticas de risco foi feita pelas diversas vantagens que a linguagem apresenta, entre elas:

- A possibilidade de criação de uma linguagem de propósito específico para a descrição das políticas de risco que é extensível para futuras revisões;
- A existência de bibliotecas prontas para o processamento de XML em várias linguagens de programação, muitas dessas bibliotecas eficientes e de código aberto;
- A possibilidade de usar mensagens com suporte a internacionalização

e localização; e

- O fato de ser livre e não necessitar de licenças ou *royalties* para seu uso.

Por padrão, para um arquivo XML ser válido ele deve informar o *Document Type Definition* (DTD) que o define. Entretanto, optou-se por utilizar o padrão *XML Schema* do *World Wide Web Consortium* (W3C) (W3C, 2013), por apresentar vantagens como um sistema de tipos bem definidos, suporte a *namespaces* e ser escrito em XML.

O *schema* deve ser informado em toda política de risco, mas como as políticas de risco são processadas sempre pela mesma aplicação e não há uma troca de mensagens nesse formato, a aplicação pode ser pré-configurada para utilizar esse formato específico, evitando o *overhead* de processar o *schema*.

Toda política de risco, assim como qualquer arquivo XML, deve começar com a declaração XML, informando a versão do padrão XML e, opcionalmente, a codificação de dados utilizada. Assim, a declaração XML de uma política de risco segue o exemplo do Quadro 4.1:

Quadro 4.1: Prólogo do XML

```
1    <?xml version= "1.0" encoding="UTF-8"?>
```

Em seguida, o elemento raiz *risk-policy* é definido e dentro dele são definidos os elementos filhos: *resource*, *user*, *metric-set*, *aggregation-engine* e *risk-threshold*.

O *risk-policy* deve informar a versão do padrão de políticas de risco sendo utilizado. No momento só existe a versão 1.0, mas essa informação é importante para suportar futuras revisões.

Os elementos *resource* e *user* contêm o atributo *id*, que representa a identificação do recurso associado e do usuário criador da política, respectivamente. O elemento *metric-set* representa o conjunto de métricas de risco que aquela política define que seja calculada para um recurso. Esse elemento contém o atributo *name* como forma de identificação.

Os elementos *aggregation-engine* e *risk-threshold* representam o método de agregação de risco utilizado e o limite de risco aceitável pela política, respectivamente. Ambos não contêm atributos, apenas dados. O tipo de dados de *aggregation-engine* é *string* e de *risk-threshold* é número real. O Quadro 4.2 apresenta um exemplo de política com os primeiros elementos descritos.

Quadro 4.2: Exemplo de política com elementos

```
1    <?xml version= "1.0" encoding="UTF-8"?>
2    <risk-policy version="1.0" xmlns:rp="http://inf.ufsc.br/~
         danielrs/risk-policy">
```

```
        <rp:resource id="0"/>
        <rp:user id="0"/>
        <rp:metric-set name="NAME">
        </rp:metric-set>
        <rp:aggregation-engine>ABC</rp:aggregation-engine>
        <rp:risk-threshold>99</rp:risk-threshold>
    </rp:risk-policy>
```

Finalmente, dentro do *metric-set* são adicionados os elementos *metric* e, dentro desses, os elementos *name*, *description* e *quantification*.

Cada elemento *metric* representa uma métrica de risco, com seu nome, descrição e uma identificação do método de quantificação utilizado, que pode ser um nome de função local (suportado pelo *Risk Engine* do CSP) ou um endereço para um *web service* definido pelo usuário.

Nenhum desses elementos contém atributos, apenas dados e todos os dados são do tipo *string*. O Quadro 4.3 apesenta um exemplo de política com todos os elementos descritos.

Quadro 4.3: Exemplo de política finalizada

```
        <risk-policy version="1.0" xmlns:rp="http://inf.ufsc.br/~
            danielrs/risk-policy">
        <rp:resource id="0"/>
        <rp:user id="0"/>
        <rp:metric-set name="NAME">
            <rp:metric>
                <rp:name>NAME</rp:name>
                <rp:description>DESC</rp:description>
                <rp:quantification>QUANTIFICATION</rp:quantification
                    >
            </rp:metric>
        </rp:metric-set>
        <rp:aggregation-engine>ABC</rp:aggregation-engine>
        <rp:risk-threshold>99</rp:risk-threshold>
    </rp:risk-policy>
```

Por fim, para que uma política de risco seja válida, o *schema* apresentado no Quadro 4.4 deve ser seguido:

Quadro 4.4: *XML Schema Definition* das políticas de risco

```
    <?xml version="1.0" encoding="UTF-8" ?>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="risk-policy" minOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="resource" type="xs:string"/ minOccurs=
                "1">
            <xs:element name="user" type="xs:string"/>
            <xs:element name="metric-set" minOccurs="1">
                <xs:complexType>
                    <xs:element name="metric" minOccurs="1">
                        <xs:complexType>
                        <xs:element name="name" type="xs:string"/
                            minOccurs="1">
```

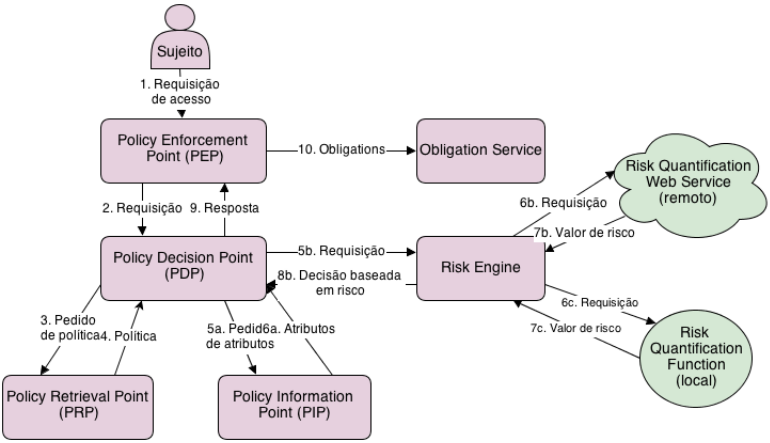

Figura 8: Processo de decisão passo a passo

```
                        <xs:element name="description" type="
                            xs:string"/ minOccurs="1">
                        <xs:element name="quantification" type="
                            xs:string"/ minOccurs="1">
                        </xs:complexType>
                    </xs:element>
                </xs:complexType>
            <xs:element name="aggregation-engine" type="xs:string"/
                minOccurs="1">
            <xs:element name="risk-threshold" type="xs:decimal"/
                minOccurs="1">
        </xs:sequence>
    </xs:complexType>
<xs:attribute name="version" type="xs:string" use="required"/>
</xs:element>
</xs:schema>
```

4.1.3 Processo de decisão

A Figura 8 apresenta, passo a passo, o processo de decisão de controle de acesso proposto.

O passo 1 é a emissão de uma requisição de acesso pelo sujeito para um recurso na nuvem. O PEP recebe essa requisição e a encaminha para o PDP (passo 2). O PDP requisita ao PAP as políticas XACML e as políticas de risco associadas ao recurso (passo 3) e as recebe como resposta (passo 4).

Nesse momento, acontecem as duas decisões de acesso em paralelo. Para a decisão XACML, o PDP requisita os atributos informados na política

ao PIP (passo 5), recebe-os como resultado (passo 6) e o pedido de acesso é então avaliado da maneira tradicional pelo PDP.

Para a decisão de risco, o PDP primeiro verifica se o recurso pode ser avaliado dessa forma. Essa permissão deve ser dada pelo CSP e pelo dono do recurso e é representada pela existência das políticas de risco associadas ao recurso. Caso não haja nenhuma política de risco associada ao recurso, isso significa que o recurso não deve ser avaliado dessa forma e o resultado da decisão será *NotApplicable*.

Caso haja políticas associadas, o PDP encaminha a requisição de acesso ao *risk engine*. O *risk engine* primeiramente analisa a política básica de risco a que o CSP está sujeito. Caso a avaliação da política básica retorne *Permit*, o *Risk Engine* analisa as políticas de risco e realiza as quantificações de acordo com as especificações (passos 6 e 7).

Como anteriormente mencionado, a quantificação pode ser local ou externa. Caso seja local, uma função no próprio *risk engine* é executada (passos 6c e 7b) e, caso seja externa, um *web service* é invocado para realizar o cálculo (passos 6b e 7b).

As métricas de risco são agregadas em um valor único e o *risk engine* retorna uma decisão ao PDP. O PDP, tendo recebido as decisões do XACML e do *risk engine*, aplica um dos algoritmos de combinação de políticas (previamente definido pelo CSP) e decide por liberar ou não o acesso, enviando a resposta ao PEP (passo 9). O PEP então é responsável por analisar e aplicar as obrigações (passo 10).

4.2 FEDERAÇÃO DE NUVENS

Um dos maiores desafios na criação e manutenção de federações de nuvens é a gerência de identidades e controle de acesso (SRIRAM, 2013).

Para implementar autorização utilizando modelos como RBAC ou ABAC, um CSP deve usar informações a respeito de um usuário, que podem ser, por exemplo, a identidade do usuário ou atributos como nome, papel ou data de nascimento.

Para que um CSP confie nas informações de atributos ou identidade de um usuário originário de outro CSP, ambos devem compartilhar algum acordo de confiança e, por isso, esse processo é normalmente mediado por uma federação de identidades.

No entanto, como comentado na seção 2.2.1, a abordagem de federação de identidades apresenta dois problemas principais ao ser implementada em cenários reais: acordos de confiança e interoperabilidade. Nesse estudo de caso, propomos utilizar o controle de acesso baseado em risco para

possibilitar a utilização de federações de nuvens sem a necessidade de acordos para a formação de federações de identidades. Essa parte do modelo foi publicada em Santos, Westphall e Westphall (2013).

Para resolver o problema da necessidade de acordos de confiança em federações de identidades, nos utilizamos do fato de que o estabelecimento da federação de nuvens já envolve um nível de confiança entre as nuvens participantes. No entanto, como comentado na seção 2.2.1, os requisitos de confiança para o estabelecimento de uma federação de nuvens são menores do que os requisitos para o estabelecimento de uma federação de identidades.

Com a federação de nuvens tem-se um agregado de entidades (nuvens computacionais) que desejam compartilhar recursos virtuais entre si, mas não necessariamente confiam que as informações de identidade trocadas entre si são confiáveis. Esse fato pode ser incluído como uma característica de risco no controle de acesso, ou seja, as informações de identidade ou atributos provindas de uma outra nuvem representam um risco em qualquer requisição de acesso se não houver confiança a nível de identidade entre as nuvens.

O desafio de interoperabilidade entre federações não pode ser completamente resolvido com o uso do controle de acesso baseado em risco, mas de certa forma é amenizado. A interoperabilidade precisa se dar de duas formas: a nível de troca de mensagens e a nível de atributos. A nível de troca de mensagens ainda é necessário que as duas entidades se comuniquem através de um protocolo em comum, por exemplo SAML. Já a nível de atributos de usuário, mesmo que as duas partes não concordem sobre quais os atributos que devem ser utilizados, ainda é possível realizar o controle de acesso. Nesse caso, o sistema de controle de acesso baseado em risco trata uma requisição de acesso sem conhecer a identidade do requisitante como uma requisição de acesso excepcional, assim como as descritas no capítulo 2, ou seja, uma requisição de acesso de alguém que, normalmente, não teria permissão para acessar aquele recurso. Por consequência, provavelmente as permissões do usuário serão menores nesse caso, mas isso não impossibilita a utilização da federação.

Em suma, o uso do controle de acesso baseado em risco possibilita o uso de federações de nuvens sem a necessidade de federações de identidades, mas ainda permite que as federações de identidades sejam formadas entre seus membros.

4.2.1 Descrição da federação de nuvens

Para descrever o uso do controle de acesso, primeiramente definimos a arquitetura de federação de nuvens utilizada. Essa arquitetura tem como

motivação principal facilitar a colaboração entre diversas nuvens. A ideia principal é que diversas nuvens, públicas ou privadas, possam se juntar para agregar seus recursos e possibilitar o compartilhamento dos mesmos, além de possibilitar que seus usuários façam a instanciação dos seus recursos virtuais em qualquer ponto da federação.

Os exemplos mais usuais de ambientes onde esse tipo de colaboração é necessário são aplicações médicas, científicas e militares, que requerem processamento intensivo e muito armazenamento, bem como um compartilhamento de informações eficiente.

A arquitetura apresentada é baseada em pontos em comum encontrados nos principais projetos de federação de nuvens atualmente em desenvolvimento, alguns descritos nos capítulos 2 e 3, no entanto ela é simplificada para explicitar as características de autorização, em detrimento das características de interoperabilidade.

A Figura 9 apresenta o diagrama geral da proposta de federação de nuvens.

A arquitetura de federação apresenta os seguintes componentes principais:

CloudProvider É o provedor de serviços de nuvem, seja ela pública ou privada, que efetivamente disponibiliza a infraestrutura sobre a qual os recursos virtuais serão alocados;

CloudManager É o componente da arquitetura responsável por ligar um CloudProvider à federação. Esse componente é bastante modular, de modo a facilitar a interoperabilidade entre diversos sistemas de gerência de nuvens; e

FederationManager Responsável por juntar os CloudManagers em uma federação. Funciona como uma lista de CloudManagers disponíveis e gerencia a troca de mensagens entre os mesmos.

Como descrito, o CloudManager é um componente bastante modular, por isso é dividido em componentes menores, chamados serviços. Os serviços são responsáveis pelas várias funcionalidades do CSP. Os serviços que compõem o CloudManager são:

UserService Responsável por gerenciar os usuários de uma *Cloud*;

ResourceService Responsável por gerenciar os recursos virtuais de uma *Cloud*, como máquinas e discos virtuais;

CloudService Responsável por se comunicar com o *software* gerenciador da nuvem (OpenNebula, OpenStack, Eucalyptus, etc) para que esse possa prover seus serviços à federação;

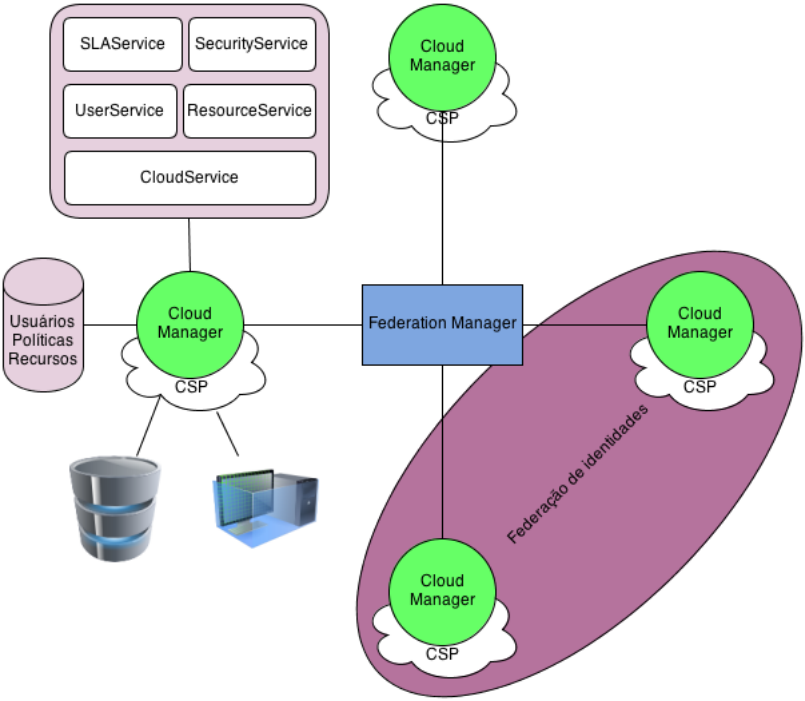

Figura 9: Diagrama geral da arquitetura de federação de nuvens

SLAService Responsável por armazenar os acordos de nível de serviço que devem ser respeitados pelo CSP; e

SecurityService Responsável pelas características de segurança da nuvem, como o controle de acesso e as políticas de segurança. O sistema de controle de acesso baseado em risco é implementado nesse serviço.

Como mostrado na Figura 9, algumas das nuvens participantes da federação podem formar federações de identidades entre si, enquanto outras fazem parte apenas da federação de nuvens.

Sob o ponto de vista de um usuário existem dois tipos de nuvens nessa arquitetura: uma nuvem de origem, que representa o seu CSP de origem, e várias nuvens estrangeiras, que são os outros membros da federação. Usuários podem criar e acessar recursos em ambos os tipos de nuvem, mas essas ações apresentam algumas particularidades.

Nas seções 4.2.2 e 4.2.3 são descritos os dois principais casos de uso da federação de nuvens, considerando o controle de acesso baseado em risco.

4.2.2 Estudo de caso - instanciação de recurso

Ao instanciar um recurso, como uma máquina ou um disco virtual, um usuário de qualquer nuvem participante da federação de nuvens pode escolher se deseja instanciar tal recurso na sua nuvem de origem ou em alguma nuvem estrangeira.

Se o recurso for instanciado em sua nuvem local, ele pode escolher se o mesmo será um recurso privado, visível apenas aos usuários da sua própria nuvem ou um recurso compartilhado, visível aos usuários de todas as nuvens da federação. Se, por sua vez, o recurso for instanciado em uma nuvem estrangeira ele deve ser compartilhado com todos os membros da federação.

No momento da instanciação do recurso, o usuário deve também criar uma política de acesso XACML relacionada ao recurso e, se desejar, uma política de risco. A criação da política de risco representa o desejo do usuário de que aquele recurso seja acessível através do controle de acesso baseado em risco. Se as duas políticas existirem, o usuário deve também optar por qual regra de combinação deseja utilizar (conforme descrito na seção 4.1).

Na criação de uma política de risco associada a um recurso, é apresentada ao usuário uma lista de opções de funções de quantificação disponíveis naquele CSP, bem como uma lista de métodos de agregação. Caso queira utilizar um método de quantificação externo, definido em um *web service*, o usuário tem acesso a uma descrição que representa o formato de dados utilizado para requisições de acesso naquele CSP e que deverá ser tratado pelo

web service, bem como uma descrição de que formato o CSP espera receber como resposta. O nível de risco aceitável também deve ser definido nesse momento.

4.2.3 Estudo de caso - acesso a recursos

Quando um usuário tenta acessar um recurso dentro da sua nuvem de origem, a requisição é tratada normalmente pelo *software* de gerência da nuvem, sem a interferência dos mecanismos criados para a federação. Já quando um usuário tenta acessar um recurso disponível em uma nuvem estrangeira, essa requisição é encaminhada a partir do *CloudManager* de origem, através do *FederationManager* até o *Cloud Manager* da nuvem onde o recurso está instanciado.

A requisição de acesso é finalmente tratada no *SecurityService*, onde estão implementados os componentes do controle de acesso baseado em risco. O processo de decisão de acesso utilizado nesse momento é o mesmo descrito na seção 4.1.2.

A Figura 10 mostra uma visão do controle de acesso baseado em risco inserido na arquitetura de federação de nuvens.

Na figura os componentes da arquitetura de controle de acesso (PEP, PDP, PRP, PIP, *Obligation Service*, *Risk Engine*, *Risk function* e *Risk quantification web service*) aparecem como um detalhe do *SecurityService* que, por sua vez, é um detalhe do *CloudManager*.

4.3 CONSIDERAÇÕES SOBRE A PROPOSTA

O uso de controle de acesso baseado em risco para computação em nuvem oferece uma grande possibilidade de flexibilização no acesso a recursos e informações. Essa utilização, porém, pode também ocasionar problemas, já que, como mencionado no capítulo 2, a análise de risco pode ser um processo muito subjetivo, o que pode acarretar em acessos indevidos. Por isso a opção de utilizar o controle de acesso baseado em risco deve recair primeiramente sobre o CSP e, finalmente, sobre os usuários donos dos recursos. Por isso também o suporte a obrigações, embutido no uso do XACML 3.0 é importante.

O fato de utilizar o cálculo de risco em paralelo com o ABAC e então combinar as duas decisões para uma decisão final cria uma gama de possibilidades de controle de acesso. O controle dinâmico baseado em risco pode ter uma influência maior ou menor na decisão final. Se o controle de acesso

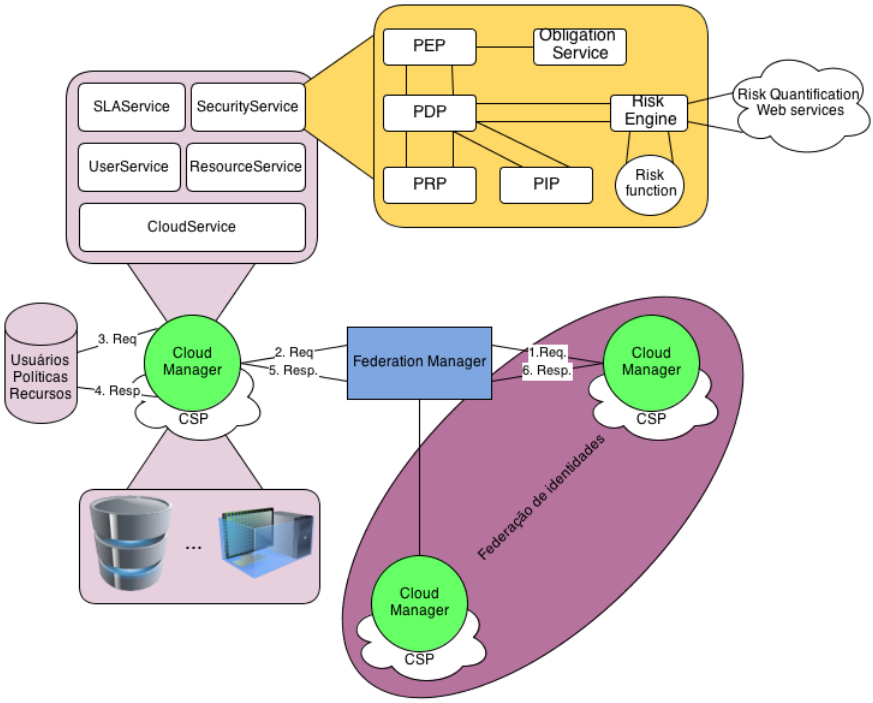

Figura 10: Controle de acesso inserido em uma determinada federação de nuvens

utilizar apenas ABAC, o risco não tem nenhuma influência; se utilizar apenas risco, o ABAC não tem nenhuma influência e, entre os dois extremos é possível dar prioridade a um ou outro ou exigir que ambos permitam o acesso para que esse seja liberado.

A proposta de uso de políticas de risco aqui apresentada tem como vantagens a possibilidade de utilizar diversos métodos de quantificação de riscos, inclusive provenientes de diferentes fontes e também possibilitar aos usuários descreverem e utilizarem seus próprios métodos de quantificação. Além disso, essa proposta garante o cumprimento de requisitos mínimos de segurança através do uso das políticas básicas.

A expressividade da proposta é demonstrada no capítulo 5, com um exemplo da implementação de duas propostas diferentes apresentadas em trabalhos relacionados.

Outra característica importante do modelo proposto é a possibilidade de distribuir os pontos de política. Essa característica é herdada do XACML e mantida na extensão proposta, já que os novos componentes também podem ser distribuídos.

Um exemplo da vantagem de utilizar uma arquitetura distribuída pode ser visualizada em uma pequena modificação do estudo de caso apresentado. Se todos os CSPs que pertencem à federação utilizassem as mesmas políticas de risco, poderia ser utilizado um único PDP para toda a federação, evitando a replicação de componentes em cada provedor individual.

As principais limitações da utilização dessa proposta são o *overhead* de processamento das políticas de risco e especialmente a degradação de desempenho quando os *web services* são utilizados. O desempenho da proposta é analisado em detalhes no capítulo 5.

Questões como troca de mensagens utilizando protocolos seguros, compartilhamento de chaves e autenticação segura não são tratados pela proposta, conforme exposto nas limitações apresentadas no capítulo 1, porque o modelo considera que a autenticação ocorre corretamente e de forma segura e que os provedores de serviços de nuvens são confiáveis. No entanto, aconselha-se que as chamadas aos *web services* utilizem HTTPS, como acontece na implementação detalhada no capítulo 5.

O uso do controle de acesso baseado em risco no cenário de federações de nuvens diminui a necessidade de utilização de federações de identidades nesses ambientes. Isso ocorre porque o próprio estabelecimento de uma federação de nuvens já envolve um nível de confiança entre os participantes e também porque torna-se possível utilizar a autenticação dos usuários provida por cada CSP separadamente.

Utilizar federações de identidades em uma federação de nuvens acarreta a criação de dois níveis de federação, como uma federação vertical, en-

quanto que utilizar a federação de nuvens com o controle de acesso baseado em risco é como utilizar uma federação horizontal, ou seja, que mantém apenas um nível.

Até onde a literatura foi consultada, não há menção a outras alternativas para gerência de identidades em federações de nuvens que não envolvam federações de identidades.

A possibilidade de não utilizar as federações de identidades traz as vantagens de evitar os problemas de escalabilidade e interoperabilidade do FIM. Entretanto, o nível de confiança entre os provedores individuais é diminuído, o que pode levar a acessos mais restritos aos recursos disponíveis e a uma maior necessidade de auditoria e possibilidade de responsabilizar e penalizar usuários por suas ações.

No fim, a escolha entre o uso de federações de identidades e controle de acesso baseado em risco torna-se uma escolha entre um maior nível de confiança ou maiores escalabilidade e interoperabilidade.

5 AMBIENTE E RESULTADOS EXPERIMENTAIS

Nesse capítulo detalha-se a implementação da proposta apresentada no capítulo 5. As ferramentas utilizadas são descritas, bem como a preparação do ambiente de testes, implementação do modelo, experimentos realizados e resultados obtidos.

5.1 DESCRIÇÃO DA IMPLEMENTAÇÃO

A implementação da proposta ocorreu em três partes, primeiramente a implementação do modelo para controle de acesso baseado em risco, em seguida a implementação da arquitetura de federação de nuvens com o controle de acesso, a fim de validar o estudo de caso apresentado no capítulo anterior e, por fim, a implementação dos métodos de quantificação e agregação de risco. A primeira parte é descrita na seção 5.2, a segunda parte na seção 5.3 e a terceira parte na seção 5.4.

As ideias fundamentais das três partes da implementação podem ser resumidas em quatro pontos principais:

Experimentação O objeto sendo desenvolvido é um protótipo e deve ser fácil utilizá-lo para testes, mas não precisa ter todas as características de um produto final. Por esse motivo não foi desenvolvida uma *Graphical User Interface* (GUI) e o tratamento de erros e exceções não é completo;

Modularização A implementação deve ser modular a ponto de permitir o desenvolvimento de novos componentes ou a troca de outros de maneira rápida e fácil;

Reuso A implementação deve reusar bibliotecas prontas e bem documentadas e, na medida do possível, ser também reusável para futuros projetos; e

Velocidade de desenvolvimento O desenvolvimento do protótipo deve ser rápido, utilizando preferencialmente uma linguagem de *scripting* e sendo bastante apoiado no reuso de bibliotecas.

Toda a implementação, composta pelo controle de acesso, pela federação de nuvens e pela quantificação de risco, foi feita utilizando-se a linguagem Python. A escolha por essa linguagem de programação se deu pela experiência prévia do autor em desenvolvimento utilizando Python, bem

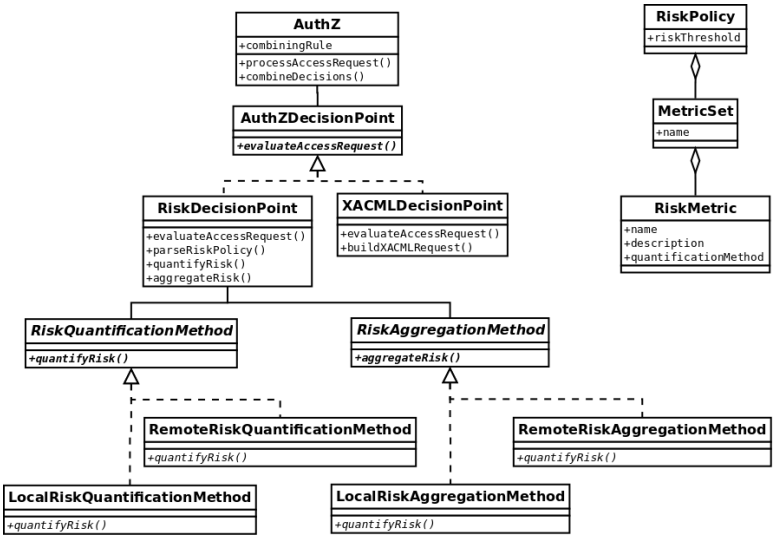

Figura 11: Diagrama de classes da arquitetura de controle de acesso

como pela velocidade de desenvolvimento e facilidade de extensão do código e experimentação, já que Python é uma linguagem interpretada. Todas as bibliotecas utilizadas foram escolhidas levando-se em conta questões como código-fonte disponível, documentação atualizada e exemplos relevantes.

Todos os trechos de código-fonte apresentados nos quadros disponíveis nesse capítulo estão escritos em Python. Apenas o Quadro 5.4 apresenta uma política de risco escrita em XML.

5.2 IMPLEMENTAÇÃO DO CONTROLE DE ACESSO

O sistema de controle de acesso utiliza a biblioteca ndg-xacml[1] para as decisões em XACML e o *framework* web.py[2] para os *web services*.

Para a implementação do controle de acesso foram criadas as classes representadas no diagrama de classes *Unified Modeling Language* (UML) da Figura 11.

A classe principal é a *AuthZ* que, através do método *processAccessRequest* atinge uma decisão de acesso. A classe *AuthZDecisionPoint* é abstrata e realizada pelas classes *RiskDecisionPoint* e *XACMLDecisionPoint*. A classe

[1] https://pypi.python.org/pypi/ndg-xacml

[2] http://webpy.org/

XACMLDecisionPoint utiliza os recursos da biblioteca ndg-xacml para atingir a sua decisão de acesso, enquanto a *RiskDecisionPoint* utiliza os métodos de quantificação e agregação de risco implementados.

Os Quadros 5.1, 5.2 e 5.3 apresentam trechos de código-fonte da implementação dos métodos mais importantes das classes *AuthZ*, *XACMLDecisionPoint* e *RiskDecisionPoint*.

No Quadro 5.1 é apresentado o método *processAccessRequest* da classe *AuthZ*. O objetivo desse método é receber uma requisição de acesso e retornar a decisão correta. Na linha 2, há uma chamada ao método de decisão de acesso baseado em XACML e na linha 3 uma chamada ao método de decisão de acesso baseado em risco. Na linha 4 as duas decisões recebidas são combinadas através de uma chamada ao método *combineDecisions* e na linha 6 a decisão de acesso é retornada.

Quadro 5.1: Método *processAccessRequest* da classe *AuthZ*

```
1 def processAccessRequest(self, request):
2     XACML_decision = self.XACMLDecisionPoint.evaluateAccessRequest(
          request)
3     risk_decision = self.RiskDecisionPoint.evaluateAccessRequest(
          request)
4     response = self.combineDecisions(XACML_decision, risk_decision)
5
6     return response
```

No Quadro 5.2 é apresentado o método *evaluateAccessRequest* da classe *XACMLDecisionPoint*. Na linha 2 é preparado um *policyReader*, que é um objeto da biblioteca *ndg-xacml* usado para processar uma política XACML, na linha 3 a política é lida e na linha 4 carregada no PDP XACML. Na linha 6 uma requicição XACML é montada a partir da requisição recebida e na linha 7 o PDP XACML retorna a sua decisão de acesso, que é então retornada ao *AuthZ* na linha 9.

Quadro 5.2: Método *evaluateAccessRequest* da classe *XACMLDecisionPoint*

```
1 def evaluateAccessRequest(self, request):
2     policyReader = ReaderFactory.getReader(Policy)
3     self.policy = policyReader.parse("policies/xacml.xml")
4     self.pdp = PDP(policy=self.policy)
5
6     XACML_request = self.buildXACMLRequest(request.user, request.
          resource, request.action)
7     response = self.pdp.evaluate(XACML_request)
8
9     return response
```

No Quadro 5.3 é apresentado o método *evaluateAccessRequest* da classe *RiskDecisionPoint*. Na linha 2 a política de risco é processada na forma de um objeto *RiskPolicy*, nas linhas 5 e 6 é realizado um laço de repetição para quantificar todas as métricas presentes na política e salvar os resultados em

uma lista. Na linha 8 os valores de risco armazenados na lista são agregados e nas linhas 9 a 14 a decisão de acesso é calculada e retornada.

Quadro 5.3: Método *evaluateAccessRequest* da classe *RiskDecisionPoint*

```
1  def evaluateAccessRequest(self, request):
2      RiskPolicy = self.parseRiskPolicy("policies/risk.xml")
3
4      risk_values = []
5      for metric in RiskPolicy.risk_metrics:
6          risk_values.append(self.quantifyRisk(metric, request))
7
8      aggregated_risk = self.aggregateRisk(RiskPolicy.
           aggregation_engine, risk_values)
9      response = "DENY"
10
11     if aggregated_risk <= RiskPolicy.risk_threshold:
12         response = "PERMIT"
13
14     return response
```

5.3 IMPLEMENTAÇÃO DA FEDERAÇÃO DE NUVENS

Inicialmente, pensou-se em agregar o sistema de controle de acesso proposto em uma plataforma de federação de nuvens existente. Ao seguir essa abordagem, porém, alguns problemas foram encontrados. Como as propostas de federações de nuvens ainda não estão bem consolidadas, não obtivemos sucesso em montar um ambiente de simulação com as ferramentas encontradas.

A principal ferramenta de federação de nuvens testada foi o projeto Contrail e os problemas encontrados foram documentações incompletas e dificuldade de acoplamento de novas funcionalidades. Por causa da documentação deficiente, não foi possível montar um ambiente de testes com o projeto Contrail. Os outros projetos de federações de nuvens citados nos capítulos 2 e 3 não têm, em sua maioria, implementações disponíveis para testes.

Como o foco do trabalho é o sistema de autorização, decidiu-se, então, fazer uma implementação simplificada de um ambiente de federação de nuvens, que não comprometesse a autorização, mas que pudesse ser implementado em tempo hábil.

O sistema de federação de nuvens implementado utiliza a biblioteca de *sockets* ZeroMQ[3] para a troca de mensagens, e o *framework* peewee[4] e o banco de dados MySQL[5] para a persistência de dados de usuários e re-

[3] http://www.zeromq.org/
[4] https://github.com/coleifer/peewee
[5] http://www.mysql.com/

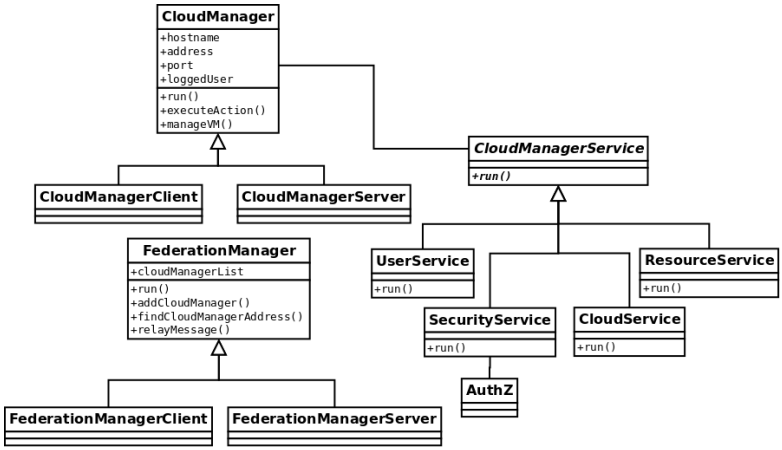

Figura 12: Diagrama de classes da federação de nuvens

cursos virtuais. O sistema de federação foi integrado ao gerenciador de nuvens OpenNebula, devido a sua popularidade e a experiências anteriores bem-sucedidas com a plataforma.

Em termos de funcionalidades foram implementadas apenas chamadas a funções existentes do OpenNebula a partir de outros nós da federação, ou seja, questões como migração de recursos virtuais não foram implementadas.

Para a implementação da federação de nuvens foram criadas as classes representadas no diagrama de classes UML da Figura 12.

A classe AuthZ representada na Figura 12 é a mesma da Figura 11, portanto todas as classes da Figura 11 estão inseridas nessa implementação, mas não foram novamente representadas pelo limite de espaço.

5.4 IMPLEMENTAÇÃO DA QUANTIFICAÇÃO DE RISCO

Para testar a implementação e demonstrar a expressividade das políticas de risco, alguns métodos apresentados em trabalhos relacionados foram implementados. Mais especificamente, detalhamos a implementação das propostas de Sharma et al. (2012) e Britton e Brown (2007), que apresentam propostas bem definidas de quantificação de risco.

O processo de implementação de ambos os exemplos é bastante semelhante. Primeiramente é definida uma política de risco que expressa o modelo e em seguida são implementadas as funções que realizam a quantificação, tanto local quanto externamente. Nenhuma modificação nas classes prontas

é necessária, apenas a adição de novos componentes, através da extensão das classes *LocalRiskQuantificationMethod* e *RemoteRiskQuantificationMethod*.

A implementação da quantificação remota, através de *web services*, foi feita utilizando os protocolos *Simple Object Access Protocol* (SOAP) e HTTPS, de acordo com a recomendação de segurança *Basic Security Profile* da *Web Services Interoperability Organization* (WS-I) (WS-I, 2010).

5.4.1 Sharma et al. (2012)

A definição de uma política de risco baseada no trabalho de Sharma et al. (2012) levou em conta a existência de três métricas: Confidencialidade (*Confidentiality*), Disponibilidade (*Availability*) e Integridade (*Integrity*). O método de agregação recebe o resultado das três quantificações e aplica a fórmula de agregação descrita no trabalho citado.

O Quadro 5.4 apresenta a política definida para essa implementação. Na linha 1 encontra-se o prólogo do XML, definindo a versão da política de risco utilizada e o *namespace* correspondente. Nas linhas 2 e 3 encontram-se a identificação do recurso associado à política e do usuário dono do recurso e na linha 5 inicia-se um conjunto de métricas com o nome do trabalho em que foram propostas (“sharma2012”).

Nas linhas 6 a 10, 12 a 16 e 18 a 22 estão definidas as métricas de risco. Em cada uma há um nome, uma descrição e um método de quantificação. No caso dessa política, todos os métodos de quantificação são externos, para demonstrar a sua implementação em forma de *web services*.

Na linha 26 está definido o método de agregação, também remoto, e na linha 27 o limite de risco aceitável. No trabalho citado não há nenhuma descrição sobre um limite de risco aceitável, mas na implementação foi utilizado um valor de 1.5, para propósitos de experimentação.

Quadro 5.4: Política de risco do método de Sharma et al. (2012)

```
1  <rp:risk-policy version="1.0" xmlns:rp="http://inf.ufsc.br/~danielrs
       ">
2      <rp:resource id="1"/>
3      <rp:user id="1"/>
4
5      <rp:metric-set name="sharma2012">
6          <rp:metric>
7              <rp:name>Confidentiality</rp:name>
8              <rp:description>Confidentiality cost</rp:description>
9              <rp:quantification>https://localhost:8443/quantify-conf<
                   /rp:quantification>
10         </rp:metric>
11
12         <rp:metric>
13             <rp:name>Availability</rp:name>
14             <rp:description>Availability cost</rp:description>
```

```
15              <rp:quantification>https://localhost:8443/quantify-avail
                    </rp:quantification>
16          </rp:metric>
17
18          <rp:metric>
19              <rp:name>Integrity</rp:name>
20              <rp:description>Integrity cost</rp:description>
21              <rp:quantification>https://localhost:8443/quantify-int</
                    rp:quantification>
22          </rp:metric>
23
24      </rp:metric-set>
25
26      <rp:aggregation-engine>https://localhost:8443/aggregate</
            rp:aggregation-engine>
27      <rp:risk-threshold>1.5</rp:risk-threshold>
28  </rp:risk-policy>
```

O Quadro 5.5 apresenta o código-fonte dos *web services* que implementam os métodos de quantificação chamados na política. O método de quantificação utilizado é o definido em Sharma et al. (2012), explicado na seção 3.1.3 dessa dissertação e resumido nas Figuras 4 e 5.

As linhas 1 a 6 são referentes à inicialização dos *web services*, com a importação dos módulos necessários e definição das *Uniform Resource Locators* (URLs) em que respondem. As linhas 10 a 21, 23 a 34 e 36 a 47 apresentam a implementação de cada métrica definida na política.

Todos os *web services* são acessados através do método HTTP GET, recebendo como parâmetros os valores de usuário, ação e recurso relacionados à requisição de acesso. Os possíveis valores de retorno são 1 ou 0, representando o impacto da ação sobre o recurso requisitado. Por exemplo, como pode ser observado nas linhas 42 a 44, a métrica *confidentiality* retorna um valor de impacto 1 caso a ação requisitada seja "VIEW" e o recurso seja considerado sensível; em qualquer outro caso a métrica citada retorna 0.

As linhas 49 a 63 representam o método de agregação. O método recupera os valores de risco calculados para cada uma das métricas, gera valores aleatórios de probabilidades do resultado de cada ação e atribui um valor ao risco anterior associado ao usuário requisitante, que foi fixado em 1. As probabilidades foram definidas aleatoriamente e o risco anterior fixado em 1 pelo fato de não existir um histórico de acessos.

Na linha 60 é aplicada a fórmula de agregação do risco, considerando as probabilidades e o valor de risco anterior.

Quadro 5.5: *Web service* implementando o método de Sharma et al. (2012)

```
1  import random
2  import web
3
4  urls = ('/quantify-conf', 'confidentiality', '/quantify-avail', '
       availability',
5          '/quantify-int', 'integrity', '/aggregate', 'aggregate')
```

```
app = web.application(urls, globals())

sensitive = ('r1')

class availability:
    def GET(self):
        params = web.input(user=None, action=None, resource=None)
        action = params.action
        resource = params.resource

        impact = 1
        if action == "VIEW" and resource not in sensitive:
            impact = 0

        web.setcookie("availability", impact)
        return impact

class integrity:
    def GET(self):
        params = web.input(user=None, action=None, resource=None)
        action = params.action
        resource = params.resource

        impact = 1
        if action == "VIEW":
            impact = 0

        web.setcookie("integrity", impact)
        return impact

class confidentiality:
    def GET(self):
        params = web.input(user=None, action=None, resource=None)
        action = params.action
        resource = params.resource

        impact = 0
        if action == "VIEW" and resource in sensitive:
            impact = 1

        web.setcookie("confidentiality", impact)
        return impact

class aggregate:
    def GET(self):
        params = web.input(user = None, action=None, resource=None)
        user = params.user
        a = float(web.cookies().get("availability"))
        i = float(web.cookies().get("integrity"))
        c = float(web.cookies().get("confidentiality"))
        randoms = (random.random(), random.random(), random.random()
            )
        p1, p2, p3 = randoms
        rs_past = self.risk_from_user(user)

        return ((a*p1) + (i*p2) + (c*p3) + rs_past)

    def risk_from_user(self, user):
        return 1

if __name__ == "__main__":
```

```
66      app.run()
```

5.4.2 Britton e Brown (2007)

O trabalho de Britton e Brown (2007) apresenta um método de quantificação para o modelo RAdAC da NSA. No método proposto, um total de 27 métricas, divididas em 6 grupos são avaliadas para cada requisição de acesso. A proposta dos autores é agregar o risco de cada métrica para atingir uma medida do risco total de segurança.

A definição de risco do trabalho citado considera tanto a probabilidade de ocorrência quanto a consequência da ocorrência de um evento como alta, baixa ou média e utiliza uma distribuição de probabilidades triangular e uma simulação de Monte Carlo para encontrar a probabilidade de cada evento. Essa probabilidade é então multiplicada por um peso atribuído por especialistas a cada uma das métricas.

Por se tratar de um método para um modelo militar, algumas métricas não são muito adequadas para o trabalho aqui desenvolvido, como o nível de ameaça da localização física do sujeito e o papel específico do sujeito na missão atual. No entanto, como o objetivo é demonstrar a expressividade das políticas de risco essas métricas também foram consideradas.

A política de risco definida para esse trabalho se encontra no Apêndice A. Essa política apresenta uma diferença em relação aos exemplos anteriores. Como o modelo considerado no trabalho de Britton e Brown (2007) é o RAdAC, não existe um limite de risco aceitável, mas sim o fato de que o risco de segurança deve ser menor do que a necessidade operacional. Na política de risco a necessidade operacional foi definida como mais uma métrica de risco e o *riskThreshold* foi definido para o valor calculado dinamicamente nessa métrica.

5.5 EXEMPLO DE USO DA IMPLEMENTAÇÃO

Para demonstrar o funcionamento da implementação do modelo para controle de acesso, descrevemos um exemplo de uso. Nesse exemplo considera-se a existência de apenas um CSP que armazena e instancia os recursos de seus usuários.

Suponhamos que Alice instancie uma máquina virtual nesse CSP e decida que o controle de acesso sobre essa máquina seja realizado através do uso de controle de acesso baseado em risco, o que é suportado pelo CSP. Ela define então uma política XACML, uma política de risco e uma regra de

combinação de políticas para essa máquina virtual.

Na política XACML, Alice define dois tipos de acesso para seu recurso. Para a ação de visualizar a máquina virtual, ela define que o acesso é permitido a ela ou aos usuários que fazem parte de seu grupo de amigos, ou seja, o provedor de identidades do usuário requisitante deve apresentar na requisição desse usuário um atributo "userId" igual a "Alice" ou um atributo "grupo" que tenha um valor "amigosDeAlice". Para as ações de editar ou excluir a máquina virtual, Alice define que apenas ela mesma tenha permissão. Em todos os outros casos, para qualquer tipo de ação o acesso é sempre negado.

Como política de risco, Alice decide utilizar aquela que foi apresentada na seção 5.4.1, que utiliza as métricas de Confidencialidade, Integridade e Disponibilidade e a quantificação apresentada na Figura 5, conforme o trabalho de Sharma et al. (2012).

Suponhamos que existam mais dois usuários nesse CSP, Bob que faz parte dos grupo de amigos de Alice e Charlie que não faz parte desse grupo.

Ao considerar o controle de acesso XACML, percebemos que quando Alice tenta acessar a máquina para qualquer ação ela tem o acesso liberado; quando Bob tenta acessar a máquina, ele tem o acesso liberado para visualização, mas não para edição ou exclusão; e quando Charlie tenta acessar a máquina, ele tem o acesso negado para qualquer operação.

Considerando o risco, como a política definida por Alice não leva em conta informações dos usuários nas métricas de risco (apenas o risco anterior usado na agregação, que na implementação foi fixado em 1), qualquer um dos três usuários que tentar acessar o recurso terá o mesmo resultado.

Vamos explorar uma decisão de acesso para a requisição de Charlie visualizar a máquina virtual de Alice. A requisição chega ao PEP e é enviada para o PDP, representado pela classe *AuthZ*, nesse momento o método *processAccessRequest* é invocado. Conforme descrito anteriormente, nesse método o PDP XACML e o *risk engine* são invocados.

No caso em análise a decisão do PDP XACML seria "DENY", já que Charlie não tem permissão pela política XACML definida por Alice. O próximo passo seria a decisão baseada em risco.

Como a ação requisitada é visualizar o recurso, considerado sensível, o resultado será impacto 0 para disponibilidade, 0 para integridade e 1 para confidencialidade (conforme a Figura 5). Consideramos que as probabilidades de ocorrência dos resultados sejam todas iguais a 0,33. O risco agregado do acesso seria, então:

$$((a*p1)+(i*p2)+(c*p3)+riscoPassado) =$$
$$((0*0,33)+(0*0,33)+(1*0,33)+1) =$$
$$1,33$$

Como o valor final do risco (1,33) é menor que o limite definido na política (1,5) o acesso seria liberado pela política de risco ("PERMIT").

Nesse momento entra em ação a combinação das políticas de controle de acesso. A Tabela 1 apresenta as possíveis decisões de acesso com base na regra de combinação utilizada e nas decisões do PDP XACML e do *risk engine*.

Tabela 1: Possíveis decisões de acesso

Regra	XACML	Risco	Decisão final
Deny overrides	DENY	PERMIT	DENY
Permit overrides	DENY	PERMIT	PERMIT
ABAC Precedence	DENY	PERMIT	DENY
Risk Precedence	DENY	PERMIT	PERMIT

O resultado final depende, então, da regra de combinação escolhida previamente por Alice. Se a regra escolhida fosse *Permit overrides* ou *Risk Precedence*, o acesso seria liberado e se fosse *Deny overrides* ou *ABAC Precedence* o acesso seria negado.

O exemplo apresentado é apenas didático, porque na implementação os valores de probabilidade são aleatórios e o valor de risco passado foi fixado, o que ignora a influência do usuário requisitante.

5.6 AMBIENTE DE TESTES E EXPERIMENTOS

Com as implementações do sistema de controle de acesso, da federação de nuvens e dos exemplos de políticas concluídas, seguiu-se para a preparação do ambiente de testes. Para os testes foram utilizadas máquinas virtuais instanciadas no serviço de IaaS Amazon EC2 (AMAZON, 2013a).

A Tabela 2 mostra os tipos de máquinas virtuais utilizadas nos experimentos. Na tabela, a unidade utilizada para representar a capacidade de processamento das máquinas virtuais ("EC2 CU") significa *EC2 Compute Unit*. O *compute unit* foi introduzido pela Amazon como medida de processamento

porque o processador físico que executa as máquinas virtuais pode ser diferente em diferentes momentos. Atualmente 1 CU é equivalente a capacidade de processamento de um processador Opteron ou Xeon de 2007 na frequência de 1.0-1.2GHz (AMAZON, 2013b).

Tabela 2: Tipos de máquinas virtuais utilizadas nos experimentos

Nome	Memória	Process.	Armaz.
M1 Small Instance	1,70 GB	1 EC2 CU	160 GB
M1 Medium Instance	3,75 GB	2 EC2 CU	410 GB
M1 Large Instance	7,50 GB	4 EC2 CU	850 GB
M1 Extra Large Instance	15,0 GB	8 EC2 CU	1690 GB

Com o objetivo de medir o desempenho do sistema de controle de acesso, foi implementado um temporizador no método *processAccessRequest*, através do *decorator* Python apresentado no Quadro 5.6:

Quadro 5.6: Temporizador de função implementado

```
1  def print_timing(func):
2      def wrapper(*arg):
3          repetitions = 50
4          times = []
5          for i in xrange(repetitions):
6              t1 = time.clock()
7              res = func(*arg)
8              t2 = time.clock()
9              print 'iteration %d: %s took %0.3f ms' % (i, func.
                   func_name, (t2-t1)*1000.0)
10             times.append((t2-t1)*1000.0)
11             print "%d times (min, max, avg) = (%0.3f ms, %0.3f ms,
                   %0.3f ms)" % (repetitions, min(times), max(times) ,
                   (sum(times) / len(times)))
12         return res
13     return wrapper
```

No Quadro 5.6 é possível observar nas linhas 3 e 5 que todas as chamadas ao método decorado são repetidas 50 vezes. Na linha 6 o tempo antes da chamada do método é obtido, na linha 7 o método decorado é chamado (no caso o método *processAccessRequest* da classe *AuthZ*, apresentado no Quadro 5.1), na linha 8 o tempo depois da chamada é obtido. Na linha 10 o tempo antes da chamada é subtraído do tempo depois da chamada, resultando no tempo de execução do método decorado. Esses tempos são salvos em uma lista e na linha 11 são impressos na tela os resultados, compostos pelo menor tempo de chamada, maior tempo de chamada e tempo médio de chamadas.

5.7 RESULTADOS E DISCUSSÃO

Para obter resultados experimentais do desempenho do modelo foram utilizadas as implementações descritas nas seções 5.2, 5.3 e 5.4 e o ambiente descrito na seção 5.6.

Foram realizados três conjuntos de experimentos, concentrados nas características de desempenho da implementação de controle de acesso. O primeiro conjunto de experimentos foi uma comparação entre diferentes políticas de controle de acesso (seção 5.7.1), o segundo uma avaliação do número de métricas em uma mesma política (seção 5.7.2) e o terceiro um estudo da influência de métricas locais e externas na mesma política (seção 5.7.3).

Todos os tempos apresentados estão em milissegundos (ms).

5.7.1 Comparação entre políticas

A Tabela 3 mostra o tempo gasto para se atingir a decisão de acesso do exemplo descrito na seção 5.5 utilizando três diferentes políticas: (i) apenas acesso XACML; (ii) acesso XACML combinado com o acesso baseado em risco de Sharma et al. (2012); e (iii) acesso XACML combinado com o acesso baseado em risco de Britton e Brown (2007).

Todas as métricas foram quantificadas em métodos locais e o método de agregação também foi implementado dessa maneira.

Tabela 3: Desempenho das políticas de risco

Política	min. (ms)	max. (ms)	média (ms)
XACML	0,925	4,278	1,040
XACML+Sharma et al. (2012)	1,986	11,973	2,436
XACML+Britton e Brown (2007)	4,395	14,234	5,352

Como esperado, o uso do XACML sem outras políticas foi o mais eficiente, seguido pelo uso do XACML com a política de *sharma2012*. A política de Britton e Brown (2007) apresentou o pior desempenho devido ao uso de um maior número de métricas de risco. A relação entre número de métricas e desempenho é explorada na seção 5.7.2.

5.7.2 Comparação entre número de métricas

Nesse conjunto de experimentos foram utilizadas políticas de risco com um número variável de métricas com quantificação local. Todas as métricas retornam valores aleatórios de risco. Essas métricas foram utilizadas para obter um resultado de desempenho baseado no número de métricas e não na complexidade de cada métrica.

A Tabela 4 apresenta os valores mínimo, máximo e médio dos experimentos de controle de acesso para políticas com um número variável de métricas.

Tabela 4: Desempenho com diferentes números de métricas

Número de métricas	min. (ms)	max. (ms)	média (ms)
1	1,832	12,130	2,243
10	2,612	12,876	3,171
100	10,922	60,442	14,030
1000	96,041	175,245	121,383
10000	1168,511	1517,364	1361,025

É possível perceber que o aumento no número de métricas locais diminui o desempenho do controle de acesso. No entanto, essa degradação de desempenho é suave e mesmo para um número muito grande de métricas (10000) a decisão de acesso é atingida em cerca de um 1,5 segundos, o que é bastante razoável.

É importante notar que o aumento no tempo utilizado para uma decisão de acesso acontece principalmente por causa do processamento da política XML e não por causa do tempo de processamento de uma métrica em si.

5.7.3 Uso de *web services*

Nesse conjunto de experimentos foi utilizada uma política de risco com 10 métricas que retornam valores aleatórios de risco, como nos experimentos anteriores. O número de 10 métricas foi utilizado porque a partir desse ponto é possível perceber que o desempenho se degrada de forma considerável e torna o uso de um número maior de métricas inviável para um sistema de controle de acesso.

Esse conjunto de métricas foi testado na implementação da federação de nuvens, mas o tempo gasto em troca de mensagens entre as nuvens foi ignorado, sendo considerado, da mesma forma que nos experimentos anteriores, apenas o tempo utilizado para se atingir a decisão de acesso.

Para esses testes foram definidos quatro casos de políticas de controle de acesso. O caso A representa 10 requisições tratadas apenas pelo XACML local; o caso B representa uma decisão de risco que envolve 10 regras de quantificação de risco realizadas localmente no CSP; o caso C utiliza 5 regras locais e 5 regras remotas (*web services*); e o caso D representa uma política de risco com 10 regras de quantificação remotas. Em todos os casos a regra de agregação utilizada foi implementada localmente. A Tabela 5 mostra os tempos obtidos em cada caso.

Tabela 5: Desempenho com métricas locais e externas

Caso	min. (ms)	max. (ms)	média (ms)
A	1,057	9,372	1,46
B	1,824	15,564	4,574
C	1556,182	2813,56	1726,71
D	3247,563	10350,5	4220,6

É fácil notar que o uso dos *web services* diminui fortemente o desempenho do controle de acesso e que o uso de apenas 10 métricas externas já traz um tempo inadequado para um sistema de controle de acesso (média de 4,2 segundos).

Para notar a diferença no uso de métricas locais e externas, a Figura 13 mostra o crescimento no tempo utilizado para se atingir uma decisão de acesso conforme aumenta-se o número de métricas. Na figura, o eixo X representa o número de métricas utilizado na política e o eixo Y representa o tempo em milissegundos para se atingir a decisão de acesso. A linha contínua e com os marcadores quadrados representa as métricas locais e a linha pontilhada e com marcadores circulares representa as métricas externas.

5.7.4 Discussão

Medir a segurança de um ambiente não é uma tarefa fácil e envolve a definição de métricas precisas, o que para modelos de controle de acesso se torna ainda mais complexo. Geralmente a avaliação de segurança de modelos de controle de acesso envolve a definição de um conjunto de estados possíveis

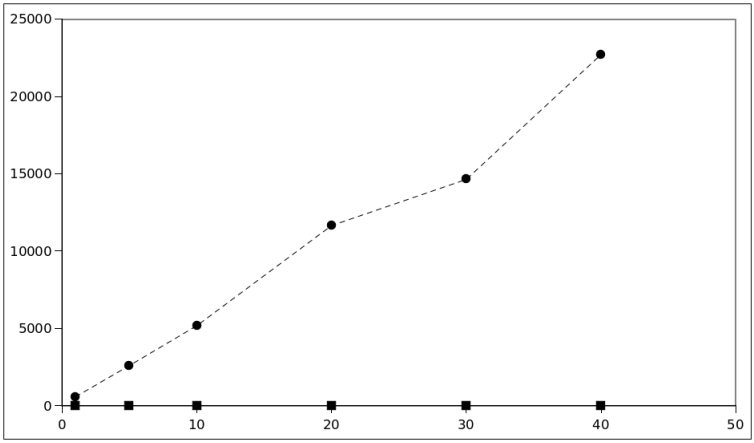

Figura 13: Tempo utilizado para se atingir uma decisão de acesso

e a prova de que em nenhuma configuração de estados há vazamento de permissões para um usuário que não tenha passado pelo controle de acesso (HU; FERRAIOLO; KUHN, 2006).

Como não há uma modelagem formal de modelos de controle de acesso baseados em risco, não é possível provar a sua corretude e, portanto, a sua segurança. Por esses motivos não foi possível realizar experimentos relativos às características de segurança do modelo para controle de acesso proposto.

Hu, Ferraiolo e Kuhn (2006) recomendam o uso de métricas qualitativas para a avaliação de sistemas de controle de acesso, baseadas em capacidades e custos administrativos, cobertura das políticas, extensibilidade e desempenho.

Quanto às questões de capacidades e custos administrativos de políticas, o modelo tem as mesmas características do XACML, adicionando a necessidade de se gerenciar também as políticas de risco. A cobertura das políticas e extensibilidade do modelo podem ser demonstradas através da implementação de métodos descritos em trabalhos relacionados e o desempenho do sistema foi a métrica efetivamente avaliada nos experimentos.

É importante destacar que todos os experimentos ocorreram em máquinas virtuais que estavam alocadas exclusivamente para os testes, ou seja, além do protótipo de controle de acesso, as únicas operações que eram executadas eram do próprio sistema operacional.

Analisando os resultados obtidos é possível tirar algumas conclusões.

Em primeiro lugar é possível perceber que o desempenho ao utilizar métodos de quantificação locais é bastante satisfatório.

Apesar de o tempo para se atingir uma decisão de acesso aumentar com o número de métricas utilizadas, o que é esperado, esse aumento não é muito impactante e ocorre principalmente por causa do processamento da política XML.

O grande problema de desempenho aparece no uso de métodos de quantificação remotos. Isso ocorre por causa do tempo gasto em comunicação *Hypertext Transfer Protocol* (HTTP), que foi considerado nos experimentos.

6 CONCLUSÕES

Nesse capítulo apresentam-se as principais conclusões obtidas com a realização do trabalho, são destacadas as maiores contribuições e algumas oportunidades de trabalhos futuros são descritas.

6.1 CONCLUSÕES

O desenvolvimento de sistemas de controle de acesso para a computação em nuvem é de grande importância, pois tais sistemas são peças fundamentais para garantir a segurança desses ambientes. A gerência de identidades e controle de acesso se torna ainda mais importante quando caminhamos para o uso de nuvens federadas, uma tendência que parece ser a evolução natural do cenário atual.

Os modelos de controle de acesso tradicionais, atualmente implementados na maioria das soluções de computação em nuvem e federações de nuvens, não são suficientes para garantir a segurança desses ambientes quando se torna necessária uma maior flexibilidade no acesso para possibilitar, por exemplo, um compartilhamento eficiente de informações em situações críticas.

Os modelos de controle de acesso baseados em risco se mostram uma alternativa para o uso em computação em nuvem e federações de nuvens. Apesar de haver na literatura algumas propostas para o uso de controle de acesso baseado em risco na nuvem, essas são muito específicas para uma dada situação, não possibilitando a sua aplicação em um contexto mais geral e sem apresentar uma arquitetura de referência que possibilite a sua extensão.

Nesse trabalho foi apresentado uma arquitetura para controle de acesso dinâmico baseado em risco para computação em nuvem, com um estudo de caso em uma federação de nuvens. Essa arquitetura foi construída como uma extensão do XACML, utilizando ABAC, adicionando a flexibilidade necessária para o compartilhamento de recursos e informações nesse ambiente, mantendo as características de distribuição e escalabilidade.

A arquitetura é baseada no uso de políticas de risco, que representam para o sistema de controle de acesso quais características de risco são importantes para o usuário dono de um recurso e como o sistema deve tratar as requisições de acesso àquele recurso.

Em resumo, o uso da arquitetura baseada em políticas de risco permite que os usuários utilizem diferentes métodos de quantificação e agregação de risco, definidos nas propostas apresentadas nos trabalhos relacionados, ou

definam seus próprios métodos, através do uso de *web services*.

Para validar a hipótese formulada no início do trabalho e a proposta apresentada, um protótipo da arquitetura foi implementado, apresentando expressividade suficiente para descrever os modelos de dois trabalhos relacionados. Medições de desempenho foram realizadas sobre esse protótipo e mostraram que o uso da arquitetura proposta é, como esperado, mais lento que o uso de XACML puro, mas com desempenho aceitável para uso de métodos de quantificação locais. O uso de métodos de quantificação remotos tem um desempenho bastante prejudicado por causa da comunicação HTTP envolvida, o que também já era esperado. Por conta disso o uso de métodos de quantificação remotos é possível, mas não é recomendável num ambiente de nuvem, em que as requisições de acesso devem ser respondidas de maneira rápida.

Espera-se que esse modelo seja um auxílio na flexibilização do acesso a informações e recursos em nuvem e na construção de federações de nuvens, fomentando a colaboração entre provedores de serviços de nuvens, para que se possa explorar ainda mais o potencial da computação utilitária.

6.2 CONTRIBUIÇÕES

As principais contribuições do trabalho são:

- A identificação de problemas de controle de acesso em computação em nuvem, advindos do uso de políticas estáticas e modelos tradicionais de controle de acesso;
- A proposta de uso de modelos de controle de acesso dinâmicos baseados em risco para computação em nuvem;
- A definição de uma arquitetura de controle de acesso dinâmico baseado em risco para computação em nuvem. Construída a partir de uma extensão do XACML e do uso de políticas de risco, oferecendo suporte a diferentes métodos de quantificação e agregação, inclusive definidos pelo usuário;
- Um estudo de caso detalhando o uso da proposta em um cenário de federação de nuvens; e
- A validação da arquitetura proposta através da sua implementação, da demonstração do seu uso para implantar outras propostas da literatura e da avaliação de desempenho dessa proposta.

As principais vantagens do uso da arquitetura proposta são:

- A possibilidade de definir uma quantidade arbitrária de métricas;
- A possibilidade de utilizar diferentes métodos de quantificação e agregação de risco;
- A possibilidade de o usuário definir novos métodos de quantificação e agregação, através de *web services*; e
- A garantia dos requisitos mínimos de segurança exigidos por provedores através do uso das políticas básicas de risco.

O desenvolvimento desse trabalho resultou na publicação do artigo Santos, Westphall e Westphall (2013), que descreve o estudo de caso. O artigo foi publicado no *The Seventh International Conference on Emerging Security Information, Systems and Technologies* - SECURWARE2013 , avaliado como B3 pelo Qualis da CAPES. Além dessa publicação, o estudo de gerência de identidades e controle de acesso em computação em nuvem, que levou ao seu desenvolvimento, gerou outras publicações, como Leandro et al. (2012), Souza et al. (2013a) e Souza et al. (2013b).

A Tabela 6 apresenta uma comparação entre as principais ideias abordadas nesse trabalho e nos trabalhos mais próximos encontrados na literatura, descritos no capítulo 3.

Os critérios considerados na tabela são: (1) Federação de nuvens; (2) Federação de identidades; (3) Controle de acesso baseado em risco; e (4) Validação. Como é possível observar, esse é o único trabalho que lida com as questões de federação de identidades e federação de nuvens utilizando o conceito de controle de acesso dinâmico baseado em risco.

Cada um dos trabalhos relacionados apresenta sua visão sobre quais características de risco e contexto de acesso são importantes em um ambiente de nuvem. Esse trabalho não teve como objetivo definir um conjunto de métricas de risco e contexto em nuvem, mas a partir da sua realização é possível tirar algumas conclusões a respeito do tema.

Em primeiro lugar, com o uso da nuvem nota-se que características usualmente associadas a contexto como: dispositivo de acesso, horário de acesso e localização do acesso não são mais tão importantes. Isso porque atualmente a quantidade de dispositivos utilizados para acesso, principalmente de dispositivos móveis, é muito grande e esses dispositivos estão sempre conectados e fazendo requisições de acesso, além de estarem em locais diversos conforme o dono do dispositivo se movimenta.

Em computação em nuvem, métricas que dizem respeito aos recursos sendo acessados são mais adequadas do que as que dizem respeito aos usuários requisitando acesso. Isso ocorre porque, conforme já exposto, é

Tabela 6: Comparação entre os trabalhos relacionados

Trabalho	(1)	(2)	(3)	(4)
(FALL et al., 2011)	Não	Não	Sim	Não
(ARIAS-CABARCOS et al., 2012)	Não	Sim	Sim	Sim
(SHARMA et al., 2012)	Não	Não	Sim	Sim
Este trabalho	Sim	Sim	Sim	Sim

muito difícil definir adequadamente o contexto de um usuário, que é muito mais dinâmico que o contexto de um recurso.

Algumas métricas que podem ser citadas como adequadas para um ambiente de nuvem, então, são: tipo de conexão e protocolo criptográfico utilizado; histórico e padrões de acesso ao recurso; impacto das ações em confidencialidade, integridade e disponibilidade (conforme Sharma et al. (2012)); e requisitos de privacidade e sensibilidade do recurso desejado. A principal métrica relacionada aos usuários que pode ser definida é um histórico de violações.

Essas conclusões são fruto de percepções sobre o ambiente de nuvem e a definição de um conjunto completo e coerente de métricas de risco para controle de acesso em computação em nuvem requer a realização de novas pesquisas.

6.3 TRABALHOS FUTUROS

Como trabalhos futuros pretende-se, inicialmente, estudar mais a fundo a relação entre federações de identidade e federações de nuvem e seus níveis de confiança.

Também seria interessante estudar outros modelos de controle de acesso baseado em risco e implementá-los nas políticas de risco, para avaliar se há necessidade da adição de novos componentes.

Gostaríamos de integrar a arquitetura de controle de acesso com um projeto de federação de nuvens existente, como o Contrail, e com um maior número de nuvens na federação, a fim de obter resultados experimentais em um ambiente mais próximo do real.

Também seria muito interessante desenvolver um método de quantificação específico para a computação em nuvem, com a identificação das métricas mais relevantes e uma forma de quantificá-las e agregá-las.

Provavelmente seria possível melhorar o desempenho do protótipo

com relação ao uso de *web services* de diversas formas: juntando todas as requisições para um mesmo servidor e disparando-as como uma única requisição; implementando um esquema de *cache* de respostas das requisições ou realizando várias requisições em paralelo, através do uso de *threads* ou processos concorrentes.

REFERÊNCIAS BIBLIOGRÁFICAS

AHMED, A.; ZHANG, N. An access control architecture for context-risk-aware access control: Architectural design and performance evaluation. In: *Emerging Security Information Systems and Technologies (SECURWARE), 2010 Fourth International Conference on*. [S.l.: s.n.], 2010. p. 251–260.

ALMUTAIRI, A. et al. A distributed access control architecture for cloud computing. *Software, IEEE*, v. 29, n. 2, p. 36–44, 2012. ISSN 0740-7459.

ALZAIN, M. et al. Cloud computing security: From single to multi-clouds. In: *System Science (HICSS), 2012 45th Hawaii International Conference on*. [S.l.: s.n.], 2012. p. 5490 –5499. ISSN 1530-1605.

AMAZON. *Amazon EC2*. 2013a. Disponível em: <http://aws.amazon.com/ec2/>. Acesso em: 01/07/2013.

AMAZON. *Amazon EC2 FAQs*. 2013b. Disponível em: <http://aws.amazon.com/ec2/faqs/>. Acesso em: 01/07/2013.

ARIAS-CABARCOS, P. et al. A metric-based approach to assess risk for "on cloud" federated identity management. *Journal of Network and Systems Management*, Springer New York, v. 20, p. 513–533, 2012. ISSN 1064-7570.

ARMBRUST, M. et al. A view of cloud computing. *Commun. ACM*, ACM, New York, NY, USA, v. 53, n. 4, p. 50–58, abr. 2010. ISSN 0001-0782.

BARACALDO, N.; JOSHI, J. A trust-and-risk aware rbac framework: tackling insider threat. In: *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*. New York, NY, USA: ACM, 2012. (SACMAT '12), p. 167–176. ISBN 978-1-4503-1295-0.

BARR, J. *Amazon S3 - Two Trillion Objects, 1.1 Million Requests / Second*. 2013. Disponível em: <http://aws.typepad.com/aws/2013/04/amazon-s3-two-trillion-objects-11-million-requests-second.html>. Acesso em: 26/06/2013.

BELL, D. E.; LAPADULA, L. J. *Secure Computer Systems: Mathematical Foundations*. [S.l.], mar. 1973. I.

BENANTAR, M. *Access Control Systems: Security, Identity Management and Trust Models*. [S.l.]: Springer, 2006. ISBN 0-387-00445-9.

BERNABE, J. B. et al. Semantic-aware multi-tenancy authorization system for cloud architectures. *Future Generation Computer Systems*, n. 0, p. –, 2012. ISSN 0167-739X.

BERNSTEIN, D. et al. Blueprint for the intercloud - protocols and formats for cloud computing interoperability. In: *Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on*. [S.l.: s.n.], 2009. p. 328 –336.

BERNSTEIN, D.; VIJ, D. Intercloud directory and exchange protocol detail using xmpp and rdf. In: *Services (SERVICES-1), 2010 6th World Congress on*. [S.l.: s.n.], 2010. p. 431 –438.

BERTINO, E.; TAKAHASHI, K. *Identity Management: Concepts, Technologies, and Systems*. [S.l.]: Artech House, 2011. ISBN 9781608070404.

BIBA. Integrity Considerations for Secure Computer Systems. *MITRE Co., technical report ESD-TR 76-372*, 1977.

BOSS, G. et al. *Cloud Computing*. [S.l.], 2007.

BRINHOSA, R. B. et al. A validation model of data input for web services. In: *ICN 2013, The Twelfth International Conference on Networks*. [S.l.: s.n.], 2013.

BRITTON, D.; BROWN, I. *A security risk measurement for the RAdAC model*. [S.l.: s.n.], 2007.

BRUCKER, A. D.; PETRITSCH, H. Extending access control models with break-glass. In: *Proceedings of the 14th ACM symposium on Access control models and technologies*. New York, NY, USA: ACM, 2009. (SACMAT '09), p. 197–206. ISBN 978-1-60558-537-6.

BUYYA, R.; BROBERG, J.; GOSCINSKI, A. M. *Cloud Computing Principles and Paradigms*. [S.l.]: Wiley Publishing, 2011. ISBN 9780470887998.

BUYYA, R.; PANDEY, S.; VECCHIOLA, C. Cloudbus toolkit for market-oriented cloud computing. In: JAATUN, M.; ZHAO, G.; RONG, C. (Ed.). *Cloud Computing*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5931). p. 24–44. ISBN 978-3-642-10664-4.

BUYYA, R.; RANJAN, R.; CALHEIROS, R. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application

services. In: HSU, C.-H. et al. (Ed.). *Algorithms and Architectures for Parallel Processing*. [S.l.]: Springer Berlin / Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6081). p. 13–31. ISBN 978-3-642-13118-9.

CABARCOS, P. A. et al. Enabling saml for dynamic identity federation management. In: WOZNIAK, J. et al. (Ed.). *Wireless and Mobile Networking*. [S.l.]: Springer Berlin Heidelberg, 2009, (IFIP Advances in Information and Communication Technology, v. 308). p. 173–184. ISBN 978-3-642-03840-2.

CALERO, J. et al. Toward a multi-tenancy authorization system for cloud services. *Security Privacy, IEEE*, v. 8, n. 6, p. 48 –55, nov.-dec. 2010. ISSN 1540-7993.

CARLINI, E. et al. Cloud federations in contrail. In: ALEXANDER, M. et al. (Ed.). *Euro-Par 2011: Parallel Processing Workshops*. [S.l.]: Springer Berlin / Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7155). p. 159–168. ISBN 978-3-642-29736-6.

CARROLL, M.; MERWE, A. van der; KOTZE, P. Secure cloud computing: Benefits, risks and controls. In: *Information Security South Africa (ISSA), 2011*. [S.l.: s.n.], 2011. p. 1–9.

CATTEDDU, D.; HOGBEN, G. *Cloud Computing: benefits, risks and recommendations for information security*. [S.l.], 2009.

CELESTI, A. et al. Federation establishment between clever clouds through a saml sso authentication profile. In: . [S.l.: s.n.], 2010a.

CELESTI, A. et al. How to enhance cloud architectures to enable cross-federation. In: *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. [S.l.: s.n.], 2010b. p. 337 –345.

CELESTI, A. et al. Security and cloud computing: Intercloud identity management infrastructure. In: *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on*. [S.l.: s.n.], 2010c. p. 263 –265. ISSN 1524-4547.

CELESTI, A. et al. Three-phase cross-cloud federation model: The cloud sso authentication. In: *Advances in Future Internet (AFIN), 2010 Second International Conference on*. [S.l.: s.n.], 2010d. p. 94 –101.

CELIKEL, E. et al. A risk management approach to rbac. *Risk and Decision Analysis*, v. 1, n. 1, p. 21–33, 2009. ISSN 0740-7459.

CHADWICK, D. W. Federated identity management. In: ALDINI, A.; BARTHE, G.; GORRIERI, R. (Ed.). *Foundations of Security Analysis and Design V*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5705). p. 96–120. ISBN 978-3-642-03828-0.

CHADWICK, D. W.; FATEMA, K. A privacy preserving authorisation system for the cloud. *Journal of Computer and System Sciences*, v. 78, n. 5, p. 1359 – 1373, 2012. ISSN 0022-0000. ¡ce:title¿JCSS Special Issue: Cloud Computing 2011¡/ce:title¿.

CHAVES, S. de; WESTPHALL, C.; LAMIN, F. Sla perspective in security management for cloud computing. In: *Networking and Services (ICNS), 2010 Sixth International Conference on*. [S.l.: s.n.], 2010. p. 212 –217.

CHENG, P. C. et al. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In: *Security and Privacy, 2007. SP '07. IEEE Symposium on*. [S.l.: s.n.], 2007. p. 222–230. ISSN 1081-6011.

CHOUDHARY, R. A policy based architecture for nsa radac model. In: *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*. [S.l.: s.n.], 2005. p. 294–301.

CHOW, R. et al. Controlling data in the cloud: outsourcing computation without outsourcing control. In: *Proceedings of the 2009 ACM workshop on Cloud computing security*. New York, NY, USA: ACM, 2009. (CCSW '09), p. 85–90. ISBN 978-1-60558-784-4.

CLARK, D. D.; WILSON, D. R. A Comparison of Commercial and Military Computer Security Policies. In: *Proceedings of the 1987 IEEE Symposium on Security and Privacy*. [S.l.]: IEEE Computer Society Press, 1987. p. 184–194.

Cloud Security Alliance. *Security Guidance for Critical Areas of Focus in Cloud Computing v3.0*. 2011.

CLOUDS360. *The Top 20 Infrastructure as a Service Vendors*. 2013a. Disponível em: <http://www.clouds360.com/iaas.php>. Acesso em: 25/06/2013.

CLOUDS360. *The Top 20 Platform as a Service Vendors*. 2013b. Disponível em: <http://www.clouds360.com/paas.php>. Acesso em: 25/06/2013.

CLOUDS360. *The Top 20 Software as a Service Vendors*. 2013c. Disponível em: <http://www.clouds360.com/saas.php>. Acesso em: 25/06/2013.

COPPOLA, M. et al. The contrail approach to cloud federations. In: *Proceedings of the International Symposium on Grids and Clouds (ISGC'12)*. [S.l.: s.n.], 2012.

DAMIANI, E.; VIMERCATI, S. D. C. di; SAMARATI, P. *New Paradigms for Access Control in Open Environments*. 2005.

Department of Defense. *The NIST Definition of Cloud Computing*. 1985. Disponível em: <http://csrc.nist.gov/publications/history/dod85.pdf>. Acesso em: 15/05/2012.

DIAMOND, S. *Standard for Intercloud Interoperability and Federation (SIIF)*. 2012. Disponível em: <https://development.standards.ieee.org/get-file/P2302.pdf>. Acesso em: 25/06/2013.

DIEP, N. N. et al. Contextual risk-based access control. In: *Security and Management*. [S.l.: s.n.], 2007. p. 406–412.

DIMMOCK, N. How much is "enough"? risk in trust-based access control. In: *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*. [S.l.: s.n.], 2003. p. 281–282. ISSN 1080-1383.

EGI. *Federated Clouds Task Force*. 2012. Disponível em: <https://wiki.egi.eu/wiki/Fedcloud-tf:FederatedCloudsTaskForce>.

F5 Networks. *Cloud Computing - Survey Results*. 2009. Disponível em: <http://www.f5.com/pdf/reports/cloud-computing-survey-results-2009.pdf>.

FALL, D. et al. Toward Quantified Risk-Adaptive Access Control for Multi-tenant Cloud Computing. In: *Proceedings of the 6th Joint Workshop on Information Security (JWIS2011)*. [S.l.: s.n.], 2011.

FARRELL, S.; HOUSLEY, R. *An Internet Attribute Certificate Profile for Authorization*. IETF, abr. 2002. RFC 3281 (Informational). (Request for Comments, 3281). Disponível em: <http://www.ietf.org/rfc/rfc3281.txt>.

FARRELL, S. et al. *AAA Authorization Requirements*. IETF, ago. 2000. RFC 2906 (Informational). (Request for Comments, 2906). Disponível em: <http://www.ietf.org/rfc/rfc2906.txt>.

FARROHA, B.; FARROHA, D. An investigative analysis into security in the clouds and the impact of virtualization on the security architecture. In: *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011*. [S.l.: s.n.], 2011. p. 1369–1374. ISSN 2155-7578.

FARROHA, B.; FARROHA, D. An adaptive sos framework for integrating dynamic cyber defense. In: *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*. [S.l.: s.n.], 2012a. p. 1–6. ISSN 2155-7578.

FARROHA, B.; FARROHA, D. Architecting dynamic cyber defense for a secure multi-tenant cloud services environment. In: *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*. [S.l.: s.n.], 2012b. p. 1–6. ISSN 2155-7578.

FARROHA, B.; FARROHA, D. Architecting dynamic privileges in protected systems through hardening identity and access management. In: *Systems Conference (SysCon), 2012 IEEE International*. [S.l.: s.n.], 2012c. p. 1–6.

FARROHA, B.; FARROHA, D. Architecting security into the clouds: An enterprise security model. In: *Systems Conference (SysCon), 2012 IEEE International*. [S.l.: s.n.], 2012d. p. 1–7.

FARROHA, B.; FARROHA, D. Challenges of operationalizing dynamic system access control: Transitioning from abac to radac. In: *Systems Conference (SysCon), 2012 IEEE International*. [S.l.: s.n.], 2012e. p. 1–7.

FARROHA, B. S.; FARROHA, D. L. Securing services in the cloud: an investigation of the threats and the mitigations. p. 840508–840508–11, 2012f.

FERRAIOLO, D. F.; KUHN, D. R. *Role-Based Access Controls*. 1992.

FOSTER, I. et al. Cloud computing and grid computing 360-degree compared. In: *Grid Computing Environments Workshop, 2008. GCE '08*. [S.l.: s.n.], 2008. p. 1 –10.

GHAZIA, U. e; MASOOD, R.; SHIBLI, M. Comparative analysis of access control systems on cloud. In: *Software Engineering, Artificial Intelligence, Networking and Parallel Distributed Computing (SNPD), 2012 13th ACIS International Conference on*. [S.l.: s.n.], 2012. p. 41–46.

GOIRI, I.; GUITART, J.; TORRES, J. Characterizing cloud federation for enhancing providers' profit. In: *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. [S.l.: s.n.], 2010. p. 123 –130.

GOUGLIDIS, A.; MAVRIDIS, I. On the definition of access control requirements for grid and cloud computing systems. In: DOULAMIS, A. et al. (Ed.). *Networks for Grid Applications*. [S.l.]: Springer Berlin Heidelberg, 2010, (Lecture Notes of the Institute for Computer Sciences,

Social Informatics and Telecommunications Engineering, v. 25). p. 19–26. ISBN 978-3-642-11733-6.

GROBAUER, B.; WALLOSCHEK, T.; STOCKER, E. Understanding cloud computing vulnerabilities. *Security Privacy, IEEE*, v. 9, n. 2, p. 50 –57, march-april 2011. ISSN 1540-7993.

GROZEV, N.; BUYYA, R. Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, John Wiley & Sons, Ltd, 2012. ISSN 1097-024X.

HAN, W. et al. *A Framework for Quantified Risk and Benefit Adaptive Access Control*. 2012. Disponível em: <http://crypto.fudan.edu.cn/people/weili/papers/han-QSBAC.pdf>. Acesso em: 15/05/2012.

HARRIS, S. *CISSP All-in-One Exam Guide, 6th edition*. [S.l.]: McGraw-Hill, 2013. ISBN 978-0-07-178173-2.

HARSH, P. et al. Contrail virtual execution platform challenges in being part of a cloud federation. In: *Proceedings of the 4th European conference on Towards a service-based internet*. Berlin, Heidelberg: Springer-Verlag, 2011. (ServiceWave'11), p. 50–61. ISBN 978-3-642-24754-5.

HU, V.; FERRAIOLO, D.; KUHN, D. R. *Assessment of Access Control Systems, Interagency Report 7316*. [S.l.], 2006.

IBM. *XACML Policy Decision Point*. 2012. Disponível em: <http://publib.boulder.ibm.com/infocenter/wsdatap/v3r8m1/index.jsp?topic=%2Fxs40% Acesso em: 04/06/2013.

ITU-T. *Baseline capabilities for enhanced global identity management and interoperability*. 2009a. Http://www.itu.int/rec/T-REC-Y.2720-200901-I. Disponível em: <http://www.itu.int/rec/http://www.itu.int/rec/T-REC-X.1250-200909-I>.

ITU-T. *NGN identity management framework*. 2009b. Http://www.itu.int/rec/T-REC-Y.2720-200901-I. Disponível em: <http://www.itu.int/rec/T-REC-Y.2720-200901-I>.

JASON Program Office. *Horizontal Integration: Broader Access Models for Realizing Information Dominance*. [S.l.], 12 2004.

KANDALA, S.; SANDHU, R.; BHAMIDIPATI, V. An attribute based framework for risk-adaptive access control models. In: *Proceedings of the*

2011 Sixth International Conference on Availability, Reliability and Security. Washington, DC, USA: IEEE Computer Society, 2011. (ARES '11), p. 236–241. ISBN 978-0-7695-4485-4.

KRAUTSEVICH, L. et al. Risk-based usage control for service oriented architecture. In: *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on*. [S.l.: s.n.], 2010. p. 641–648. ISSN 1066-6192.

KURZE, T. et al. Cloud federation. In: *CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization*. [S.l.: s.n.], 2011. p. 32–38. ISBN 978-1-61208-153-3.

LAMPROPOULOS, K.; DENAZIS, S. Identity management directions in future internet. *Communications Magazine, IEEE*, v. 49, n. 12, p. 74–83, december 2012. ISSN 0163-6804.

LAMPSON, B. W. Protection. *SIGOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 8, n. 1, p. 18–24, jan. 1974. ISSN 0163-5980.

LASZEWSKI, G. von et al. Comparison of multiple cloud frameworks. In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. [S.l.: s.n.], 2012. p. 734–741. ISSN 2159-6182.

LEANDRO, M. A. P. et al. Multi-tenancy authorization system with federated identity for cloud environments using shibboleth. In: *ICN 2012, The Eleventh International Conference on Networks*. [S.l.: s.n.], 2012.

LEE, H.; JEUN, I.; JUNG, H. Criteria for evaluating the privacy protection level of identity management services. *Emerging Security Information, Systems, and Technologies, The International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 155–160, 2009.

LI, Y. et al. Using trust and risk in access control for grid environment. In: *Proceedings of the 2008 International Conference on Security Technology*. Washington, DC, USA: IEEE Computer Society, 2008. (SECTECH '08), p. 13–16. ISBN 978-0-7695-3486-2.

MATHER, T.; KUMARASWAMY, S.; LATIF, S. *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 0596802765, 9780596802769.

MELL, P.; GRANCE, T. *The NIST Definition of Cloud Computing*. 2011. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Acesso em: 15/05/2012.

MOLLOY, I.; CHENG, P.-C.; ROHATGI, P. Trading in risk: using markets to improve access control. In: *Proceedings of the 2008 workshop on New security paradigms*. New York, NY, USA: ACM, 2008. (NSPW '08), p. 107–125. ISBN 978-1-60558-341-9.

MOLLOY, I. et al. Risk-based security decisions under uncertainty. In: *Proceedings of the second ACM conference on Data and Application Security and Privacy*. New York, NY, USA: ACM, 2012. (CODASPY '12), p. 157–168. ISBN 978-1-4503-1091-8.

NI, Q.; BERTINO, E.; LOBO, J. Risk-based access control systems built on fuzzy inferences. In: *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. New York, NY, USA: ACM, 2010. (ASIACCS '10), p. 250–260. ISBN 978-1-60558-936-7.

OASIS. *A Brief Introduction to XACML*. 2003. Disponível em: <https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html>. Acesso em: 19/02/2013.

OASIS. *Available XACML implementations*. 2013a. Disponível em: <https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml#other>. Acesso em: 26/06/2013.

OASIS. *eXtensible Access Control Markup Language (XACML) Version 3.0*. 2013b. Disponível em: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>. Acesso em: 01/07/2013.

OLDEN, E. Architecting a cloud-scale identity fabric. *Computer*, v. 44, n. 3, p. 52–59, 2011. ISSN 0018-9162.

OLSHANSKY, S. *Distributed Dynamic SAML*. 2008. Disponível em: <https://spaces.internet2.edu/display/dsaml/Distributed+Dynamic+SAML>. Acesso em: 11/06/2013.

ORBEGOZO, I. S. A. et al. Cloud capacity reservation for optimal service deployment. In: *CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization*. [S.l.: s.n.], 2011.

PARK, J.; SANDHU, R. Towards usage control models: beyond traditional access control. In: *Proceedings of the seventh ACM symposium on Access control models and technologies*. New York, NY, USA: ACM, 2002. (SACMAT '02), p. 57–64. ISBN 1-58113-496-7.

PARK, J.; SANDHU, R. The uconabc usage control model. *ACM Trans. Inf. Syst. Secur.*, ACM, New York, NY, USA, v. 7, n. 1, p. 128–174, fev. 2004. ISSN 1094-9224.

PETERSON, G. Introduction to identity management risk metrics. *Security Privacy, IEEE*, v. 4, n. 4, p. 88–91, 2006. ISSN 1540-7993.

PFITZMANN, A.; HANSEN, M. *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*. ago. 2010. Http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf. V0.34. Disponível em: <http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf>.

REN, K.; WANG, C.; WANG, Q. Security challenges for the public cloud. *Internet Computing, IEEE*, v. 16, n. 1, p. 69 –73, jan.-feb. 2012. ISSN 1089-7801.

ROCHWERGER, B. et al. The reservoir model and architecture for open federated cloud computing. *IBM J. Res. Dev.*, IBM Corp., Riverton, NJ, USA, v. 53, n. 4, p. 535–545, jul. 2009. ISSN 0018-8646.

SAMARATI, P.; VIMERCATI, S. de. Access control: Policies, models, and mechanisms. In: FOCARDI, R.; GORRIERI, R. (Ed.). *Foundations of Security Analysis and Design*. [S.l.: s.n.], 2001, (Lecture Notes in Computer Science, v. 2171). p. 137–196.

SANDHU, R.; FERRAIOLO, D.; KUHN, R. The nist model for role-based access control: towards a unified standard. In: *Proceedings of the fifth ACM workshop on Role-based access control*. New York, NY, USA: ACM, 2000. (RBAC '00), p. 47–63. ISBN 1-58113-259-X.

SANTOS, D. R. dos; WESTPHALL, C. M. Uma aplicação de privacidade no gerenciamento de identidades em nuvem com uapprove. In: *Anais do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*. [S.l.: s.n.], 2011.

SANTOS, D. R. dos; WESTPHALL, C. M.; WESTPHALL, C. B. Risk-based dynamic access control for a highly scalable cloud federation. In: *Emerging Security Information Systems and Technologies (SECURWARE), 2013 Seventh International Conference on*. [S.l.: s.n.], 2013.

SCHAFFER, H. X as a service, cloud computing, and the need for good judgment. *IT Professional*, v. 11, n. 5, p. 4–5, 2009. ISSN 1520-9202.

SHAIKH, R. A.; ADI, K.; LOGRIPPO, L. Dynamic risk-based decision methods for access control systems. *Computers & Security*, v. 31, n. 4, p. 447–464, 2012.

SHARMA, M. et al. Using risk in access control for cloud-assisted ehealth. In: *High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*. [S.l.: s.n.], 2012. p. 1047–1052.

SOUZA, R. F. de et al. Challenges of operationalizing pacs on cloud over wireless networks. In: *ICWMC 2013, The Ninth International Conference on Wireless and Mobile Communications*. [S.l.: s.n.], 2013a.

SOUZA, R. F. de et al. A review of pacs on cloud for archiving secure medical images. *International Journal of Privacy and Health Information Management (IJPHIM)*, v. 1, n. 1, p. 53–62, 2013b.

SRIRAM, D. N. Dissertação (Mestrado) — Der Technishcen Universitat Munchen, january 2013.

SUHENDRA, V. A survey on access control deployment. In: *FGIT-SecTech*. [S.l.: s.n.], 2011. p. 11–20.

TAKABI, H.; JOSHI, J. B.; AHN, G.-J. Security and privacy challenges in cloud computing environments. *IEEE Security and Privacy*, IEEE Computer Society, Los Alamitos, CA, USA, v. 8, p. 24–31, 2010. ISSN 1540-7993.

VOLLBRECHT, J. et al. *AAA Authorization Application Examples*. IETF, ago. 2000a. RFC 2905 (Informational). (Request for Comments, 2905). Disponível em: <http://www.ietf.org/rfc/rfc2905.txt>.

VOLLBRECHT, J. et al. *AAA Authorization Framework*. IETF, ago. 2000b. RFC 2904 (Informational). (Request for Comments, 2904). Disponível em: <http://www.ietf.org/rfc/rfc2904.txt>.

VUKOLIć, M. The byzantine empire in the intercloud. *SIGACT News*, ACM, New York, NY, USA, v. 41, n. 3, p. 105–111, set. 2010. ISSN 0163-5700.

W3C. *XML Schema*. 2013. Disponível em: <http://www.w3.org/XML/Schema>. Acesso em: 26/06/2013.

WANG, Q.; JIN, H. Quantified risk-adaptive access control for patient privacy protection in health information systems. In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*.

New York, NY, USA: ACM, 2011. (ASIACCS '11), p. 406–410. ISBN 978-1-4503-0564-8.

WS-I. *Basic Security Profile Version 1.1*. 2010. Disponível em: <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.1.html>. Acesso em: 27/06/2013.

YU, S. et al. Achieving secure, scalable, and fine-grained data access control in cloud computing. In: *INFOCOM, 2010 Proceedings IEEE*. [S.l.: s.n.], 2010. p. 1–9. ISSN 0743-166X.

ZHANG, G.; PARASHAR, M. Dynamic context-aware access control for grid applications. In: *Grid Computing, 2003. Proceedings. Fourth International Workshop on*. [S.l.: s.n.], 2003. p. 101 – 108.

ZHANG, G.; PARASHAR, M. Context-aware dynamic access control for pervasive applications. In: *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*. [S.l.: s.n.], 2004. p. 21–30.

ZHANG, L.; BRODSKY, A.; JAJODIA, S. Toward information sharing: benefit and risk access control (barac). In: *Policies for Distributed Systems and Networks, 2006. Policy 2006. Seventh IEEE International Workshop on*. [S.l.: s.n.], 2006. p. 9 pp.–53.

ZISSIS, D.; LEKKAS, D. Addressing cloud computing security issues. *Future Generation Computer Systems*, v. 28, n. 3, p. 583 – 592, 2012. ISSN 0167-739X.

APÊNDICE A – Política de risco de Britton e Brown (2007)

Quadro A.1: Política de risco de Britton e Brown (2007)

```
<rp:risk-policy version="1.0" xmlns:rp="http://inf.ufsc.br/~danielrs
    ">
    <rp:resource id="1"/>

    <rp:user id="1"/>

    <rp:metric-set name="Characteristics of requester">
        <rp:metric>
            <rp:name>Role</rp:name><rp:description>Papel do sujeito
                na organizacao</rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

        <rp:metric>
            <rp:name>Rank</rp:name><rp:description>Posicao relativa
                do sujeito na organizacao</rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

        <rp:metric>
            <rp:name>Clearance Level</rp:name><rp:description>Nivel
                de habilitacao do sujeito</rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

        <rp:metric>
            <rp:name>Access Level</rp:name><rp:description>Liberacao
                 previa da informacao para o sujeito</rp:description
                >
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

        <rp:metric>
            <rp:name>Previous violations</rp:name><rp:description>
                Violacoes previas do sujeito</rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

        <rp:metric>
            <rp:name>Education Level</rp:name><rp:description>Nivel
                de conhecimento em seguranca do sujeito</
                rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>
    </rp:metric-set>

    <rp:metric-set name="Characteristics of IT Components">
        <rp:metric>
            <rp:name>Machine Type</rp:name><rp:description>Tipo de
                maquina envolvida no acesso</rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

        <rp:metric>
            <rp:name>Application</rp:name><rp:description>Risco
                associado a aplicÃ§Ã£o usada no acesso</
                rp:description>
            <rp:quantification>brittonQuantify</rp:quantification>
        </rp:metric>

```

```
49          <rp:metric>
50              <rp:name>Connection Type</rp:name><rp:description>Tipo
                    de conexao usado no acesso (por exemplo, cabeada ou
                    sem fio)</rp:description>
51              <rp:quantification>brittonQuantify</rp:quantification>
52          </rp:metric>
53
54          <rp:metric>
55              <rp:name>Authentication Type</rp:name><rp:description>
                    Risco associado ao metodo de autenticacao utilizado<
                    /rp:description>
56              <rp:quantification>brittonQuantify</rp:quantification>
57          </rp:metric>
58
59          <rp:metric>
60              <rp:name>Network</rp:name><rp:description>TIpo de rede
                    utilizada no acesso</rp:description>
61              <rp:quantification>brittonQuantify</rp:quantification>
62          </rp:metric>
63
64          <rp:metric>
65              <rp:name>QoP/Encryption Level</rp:name><rp:description>
                    Nivel de cifragem usada para proteger a informacao
                    durante a transmissao</rp:description>
66              <rp:quantification>brittonQuantify</rp:quantification>
67          </rp:metric>
68
69          <rp:metric>
70              <rp:name>Distance from requester to source</rp:name><
                    rp:description>Distancia fisica entre o sujeito e a
                    informacao</rp:description>
71              <rp:quantification>brittonQuantify</rp:quantification>
72          </rp:metric>
73      </rp:metric-set>
74
75      <rp:metric-set name="Heuristics">
76          <rp:metric>
77              <rp:name>Risk Knowledge</rp:name><rp:description>Risco
                    referente a violacoes previas associadas a
                    informacao</rp:description>
78              <rp:quantification>brittonQuantify</rp:quantification>
79          </rp:metric>
80
81          <rp:metric>
82              <rp:name>Trust Level</rp:name><rp:description>Risco
                    associado a um historico de transacoes que ocorreram
                     com sucesso</rp:description>
83              <rp:quantification>brittonQuantify</rp:quantification>
84          </rp:metric>
85      </rp:metric-set>
86
87      <rp:metric-set name="Situational Factors">
88          <rp:metric>
89              <rp:name>Specific Mission Role</rp:name><rp:description>
                    Papel do sujeito na missao atualmente desempenhada</
                    rp:description>
90              <rp:quantification>brittonQuantify</rp:quantification>
91          </rp:metric>
92
93          <rp:metric>
94              <rp:name>Time sensitivity of information</rp:name><
```

```
                rp:description>Quando a informacao e necessaria</
                rp:description>
95          <rp:quantification>brittonQuantify</rp:quantification>
96      </rp:metric>
97
98      <rp:metric>
99          <rp:name>Transaction Type</rp:name><rp:description>Tipo
                de operacao aplicado sobre a informacao</
                rp:description>
100         <rp:quantification>brittonQuantify</rp:quantification>
101     </rp:metric>
102
103     <rp:metric>
104         <rp:name>Auditable or Non-auditable</rp:name><
                rp:description>Indica se o acesso estara disponivel
                para auditoria</rp:description>
105         <rp:quantification>brittonQuantify</rp:quantification>
106     </rp:metric>
107
108     <rp:metric>
109         <rp:name>Audience size</rp:name><rp:description>Numero
                esperado de individuos ou maquinas que terao acesso
                a informacao</rp:description>
110         <rp:quantification>brittonQuantify</rp:quantification>
111     </rp:metric>
112 </rp:metric-set>
113
114 <rp:metric-set name="Environmental Factors">
115     <rp:metric>
116         <rp:name>Current Location</rp:name><rp:description>Nivel
                 de seguranca fisica do local onde o sujeito se
                encontra</rp:description>
117         <rp:quantification>brittonQuantify</rp:quantification>
118     </rp:metric>
119
120     <rp:metric>
121         <rp:name>Operational Environment Threat Level</rp:name><
                rp:description>Nivel de ameaca do local onde o
                sujeito se encontra</rp:description>
122         <rp:quantification>brittonQuantify</rp:quantification>
123     </rp:metric>
124 </rp:metric-set>
125
126 <rp:metric-set name="Characteristics of Information Requested">
127     <rp:metric>
128         <rp:name>Classification Level</rp:name><rp:description>
                Nivel de sensibilidade da informacao</rp:description
                >
129         <rp:quantification>brittonQuantify</rp:quantification>
130     </rp:metric>
131
132     <rp:metric>
133         <rp:name>Encryption Level</rp:name><rp:description>Nivel
                 de cifragem requerido para acesso a informacao</
                rp:description>
134         <rp:quantification>brittonQuantify</rp:quantification>
135     </rp:metric>
136
137     <rp:metric>
138         <rp:name>Network Classification Level</rp:name><
                rp:description>Nivel requerido de classificacao da
```

```
                rede usada no acesso</rp:description>
139             <rp:quantification>brittonQuantify</rp:quantification>
140         </rp:metric>
141
142         <rp:metric>
143             <rp:name>Permission Level</rp:name><rp:description>Risco
                     associado as permissoes da informacao</
                    rp:description>
144             <rp:quantification>brittonQuantify</rp:quantification>
145         </rp:metric>
146
147         <rp:metric>
148             <rp:name>Perishable/Non-Perishable</rp:name><
                    rp:description>Indica se os dados sÃ£o uteis ao
                    longo do tempo</rp:description>
149             <rp:quantification>brittonQuantify</rp:quantification>
150         </rp:metric>
151     </rp:metric-set>
152
153     <rp:metric-set name="operationalNeed">
154         <rp:metric>
155             <rp:name>Operational need</rp:name><rp:description>
                    Necessidade operacional de o sujeito acessar a
                    informacao</rp:description>
156             <rp:quantification>brittonQuantify</rp:quantification>
157         </rp:metric>
158     </rp:metric-set>
159
160     <rp:aggregation-engine>brittonAggregate</rp:aggregation-engine>
161
162     <rp:risk-threshold>operationalNeed</rp:risk-threshold>
163
164 </rp:risk-policy>
```