Avaliação Resiliente de Autorização $UCON_{ABC}$ para Computação em Nuvem

Arlindo L. Marcon Jr., Altair O. Santin, Maicon Stihler

Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR) – Curitiba, PR – Brasil

{almjr, santin, stihler}@ppgia.pucpr.br

Abstract. *The cloud services contractor needs a fine grained control of each user's individual consumption to define usage management procedures with higher precision. A $UCON_{ABC}$ approach offers individualized periodic evaluations, performing continuous reevaluations of a user's authorization attributes. However, this approach was not designed for the dynamic context of cloud computing. The work presented in this paper shows that it is possible to provide resilience to the process of reevaluating authorizations of $UCON_{ABC}$. The development of proof of concept shows cloud computing provides elasticity to the entities responsible for attribute accounting and its evaluation, enabling the proposed approach.*

Resumo. *O contratante dos serviços de nuvem necessita de controle fino de consumo individual de seus usuários para definir esquemas de gerenciamento de uso mais precisos. A abordagem de controle $UCON_{ABC}$ oferece avaliações individualizadas periódicas, executando a reavaliação contínua dos atributos de autorização do usuário. Porém, este esquema não foi projetado para o contexto dinâmico da nuvem computacional. O trabalho apresentado no artigo mostra que é possível prover resiliência ao processo de reavaliação de autorização do $UCON_{ABC}$. O desenvolvimento de uma prova de conceito mostra que a computação em nuvem prove elasticidade às entidades que contabilizam e avaliam os atributos do uso, viabilizando o esquema proposto.*

1. Introdução

A arquitetura cliente-servidor e os *softwares* de prateleira ainda dominam o mercado. Porém, a computação em nuvem está alterando este cenário, fornecendo serviços que podem ser contratados de acordo com a necessidade do consumidor. O ônus de gerenciamento do ambiente fica a cargo do provedor de serviços. Esta abordagem afeta todos os níveis do ecossistema computacional (*i.e.*, desde a infraestrutura até os usuários) [Erickson *et al.*, 2009] [Hayes, 2008]. As principais entidades envolvidas nesse contexto podem ser brevemente descritas como: *i) provedores*, fornecem os serviços para a nuvem computacional - *e.g.*, infraestrutura, plataforma; *ii) consumidor*, entidade que contrata os serviços para atender seus usuários; *iii) usuário*, sujeito que utiliza os serviços – o usuário final [Marcon Jr. *et al.*, 2010].

Os provedores que fazem parte da nuvem devem honrar os contratos de nível de serviço (*Service Level Agreements* - *SLAs*) estabelecidos com o consumidor, controlar o acesso aos sistemas e dados em nível de usuário e administrativo (privilegiado). Adicionalmente, os provedores deveriam monitorar os recursos alocados aos

Este trabalho é parcialmente financiado pelo Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC/MCTI), contrato 1313 e CNPq, processos: 478285/2011-6 e 310671/2012-4.

consumidores e oferecer mecanismos de gerenciamento e controle de uso de acordo com as especificidades de cada domínio [Emeakaroha *et al.*, 2011]. O esquema de gerenciamento a ser aplicado deve ser escalável, efetivo e eficiente para provedores e consumidores dos serviços [CSA, 2011].

A demanda gerada pelos usuários do consumidor origina diferentes cenários de carga para os serviços da nuvem (*e.g.*, processamento, armazenamento). Uma única sessão de consumo pode necessitar de recursos de diversos provedores. Neste caso, o emprego da abordagem tradicional – configuração estática de políticas nos serviços – pode caracterizar a subutilização de recursos em um provedor e a sobrecarga destes em outro. Em suma, não há como prever a demanda que um usuário vai gerar no serviço. Em um contexto ideal, as políticas de uso deveriam ser frequentemente avaliadas visando detectar essa disparidade de consumo. A monitoração periódica dos recursos asseguraria a utilização uniforme dos serviços sem prejudicar o acesso do usuário.

Neste trabalho utilizamos agentes de monitoramento para coletar informações de consumo e o controle de uso (*Usage Control* - $UCON_{ABC}$) para reavaliar políticas. O $UCON_{ABC}$ estende o controle de acesso clássico executando a avaliação continua da autorização de um usuário [Park *et al.*, 2004]. A continuidade do uso é autorizada de acordo com a reavaliação das políticas. O uso pode ser entendido como operações de escrita em um objeto (*e.g.*, um arquivo) ou o consumo de recursos (*e.g.*, ciclos de *CPU*). A frequência com que ocorre a coleta de atributos se reflete diretamente no período de tempo em que o usuário pode esta violando uma política, situação esta que caracteriza uma exceção (*i.e.*, disparidade entre a autorização concedida e a política vigente). Evidentemente, é desejável aplicar o menor intervalo de tempo possível para obtenção dos atributos de uso e respectiva reavaliação das políticas. Esta proposta mostra um cenário em que os períodos de inconsistência de autorização (situação na qual o usuário esta em condição de exceção) são mitigados.

O trabalho esta organizado da seguinte maneira: a seção 2 descreve brevemente a nuvem computacional; a seção 3 aborda os modelos de controle de acesso; a seção 4 apresenta trabalhos relacionados; a seção 5 descreve a proposta de controle de uso; a seção 6 apresenta o protótipo e os testes; por fim, a seção 7 aborda as conclusões.

2. Computação em Nuvem

A nuvem (*Cloud Computing*) busca fornecer alta disponibilidade e elasticidade com base nas seguintes características [Mell *et al.*, 2009]: *i) auto-atendimento*: o consumidor contrata serviços computacionais de acordo com sua necessidade; *ii) amplo acesso à rede*: os serviços são disponibilizados na *Internet*; *iii) pool de recursos*: o provedor agrupa seus recursos e os redistribui conforme a necessidade do ambiente; *iv) elasticidade*: a quantidade de recursos fornecidos pode ser alterada dinamicamente; *v) contabilidade*: a nuvem fornece métricas de consumo de acordo com o serviço.

A nuvem pode ser vista como uma pilha de serviços, onde cada serviço é construído utilizando os níveis inferiores. Os provedores que compõem a nuvem podem ser classificados de acordo com o tipo de serviço fornecido [Mell *et al.*, 2009]: *i) Software como um Serviço – SaaS*: oferece o produto final da computação em nuvem, o *software* que o consumidor utiliza; *ii) Plataforma como um Serviço – PaaS*: oferece suporte para o consumidor implantar suas próprias aplicações e gerenciar o sistema que hospeda as configurações do ambiente; *iii) Infraestrutura como um Serviço – IaaS*: provê recursos computacionais básicos (*e.g.*, processamento, armazenamento),

permitindo que o consumidor execute seus próprios sistemas operacionais (*e.g.*, *Windows*, *Linux*) [Grobauer *et al.*, 2010].

3. Controle de Acesso

Em ambientes heterogêneos e distribuídos a abordagem mais utilizada para a avaliação das políticas de controle de acesso emprega as seguintes entidades [Yavatkar *et al.*, 2010]: monitor de referência; guardião do serviço; repositório de políticas e repositório de atributos. No modelo baseado em terceirização (*outsourcing*, Figura 1), o guardião do serviço (*Policy Enforcement Point - PEP*) recebe a solicitação de acesso do usuário (*evento o1*) e a encaminha para o monitor de referência (*Policy Decision Point - PDP*; *evento o2*). O *PDP* recupera as políticas do repositório (*Policy Administration Point – PAP*; *evento* $o3_i$) e decide (*evento o4*) se a solicitação de consumo deve ser permitida ou negada. O resultado da decisão é encaminhado para o *PEP* (*evento o5*) para que este execute a ação determinada pelo *PDP* (*evento o6*). No processo de decisão, o *PDP* pode utilizar dados do repositório de atributos (*Policy Information Point – PIP*; *evento* $o3_{ii}$).

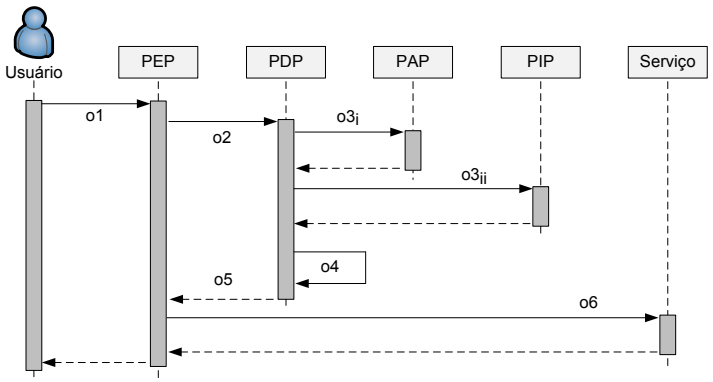

Figura 1. Arquitetura de controle de acesso para ambientes heterogêneos

A abordagem de controle de acesso que melhor contempla as necessidades de ambientes dinâmicos é o controle de uso $UCON_{ABC}$ [Park *et al.*, 2004]. Este modelo reavalia as autorizações periodicamente, levando em consideração a mutabilidade dos atributos [Teigão *et al.*, 2011]. Autorizações podem ser entendidas da maneira tradicional de avaliação e concessão de direitos (*e.g.*, leitura, escrita). Porém, o $UCON_{ABC}$ avalia continuamente os controles, tendo em vista que os atributos do usuário ou objeto (*e.g.*, serviço) podem ser alterados à medida que o acesso é executado. Assim, os atributos mutáveis, referentes ao consumo, podem ser atualizados antes ou durante a utilização do objeto. O modelo de restrições $UCON_{ABC}$ considera que os controles de acesso devem ser avaliados antes (*pre*) e durante (*ongoing*) o uso. A avaliação antes do uso é a clássica, enquanto a avaliação durante o uso é necessária devido à mutabilidade dos atributos.

4. Trabalhos Relacionados

A proposta de Zhang e Zhou [Zhang *et al.*, 2009] auxilia os provedores e consumidores a aumentarem a flexibilidade dos serviços oferecidos na nuvem. A abordagem considera somente a camada *SaaS*, agregando diferentes serviços em uma única interface de acesso. Está proposta adota o esquema de segurança tradicional, desconsiderando a flexibilidade provida pelos demais níveis de serviço (*e.g.* camadas *PaaS*, *IaaS*).

Para Lim e seus colegas [Lim *et al.*, 2009], políticas de controle munidas de dados fornecidos pelo provedor podem auxiliar o consumidor a dimensionar a utilização dos

recursos contratados. O artigo apresenta um modelo que expõe a necessidade que o consumidor tem de gerenciar a cota contratada de acordo com a demanda do ambiente.

Para Bertram [Bertram *et al.*, 2010], a infraestrutura orientada a serviço pode ser utilizada para gerenciar serviços providos na nuvem. A proposta utiliza atributos gerados pelo ambiente, criando e mapeando estes para políticas de controle. Apesar de utilizar uma abordagem interessante, o trabalho não deixa claro como o processo é executado.

A proposta de Tavizi [Tavizi *et al.*, 2012] aplica o tratamento de obrigações $UCON_{ABC}$ em ambientes de nuvem. O trabalho de Danwei [Danwei *et al.*, 2009] apresenta um módulo de negociação que aumenta a flexibilidade do sistema de controle de acesso, permitindo ao usuário executar uma segunda tentativa quando a solicitação de consumo inicial não é compatível com as políticas do serviço. Ambos os trabalhos citados desconsideram a periodicidade do sistema de monitoramento de atributos. A periodicidade reflete diretamente na resposta da reavaliação de autorização executada para o usuário.

Serviços hospedados na nuvem são distribuídos e dinâmicos. Processos de gerenciamento e operação tradicionais não são adequados para esse ambiente. É necessário um esquema de avaliação de políticas de uso que execute a reavaliação continua das regras de acesso no sentido de manter a resiliência do ambiente.

5. Avaliação Resiliente de Autorização $UCON_{ABC}$

A proposta se aplica a uma nuvem computacional formada por provedores e consumidores de serviços, sendo que a comunicação entre as partes é mediada por um domínio intermediário, a federação.

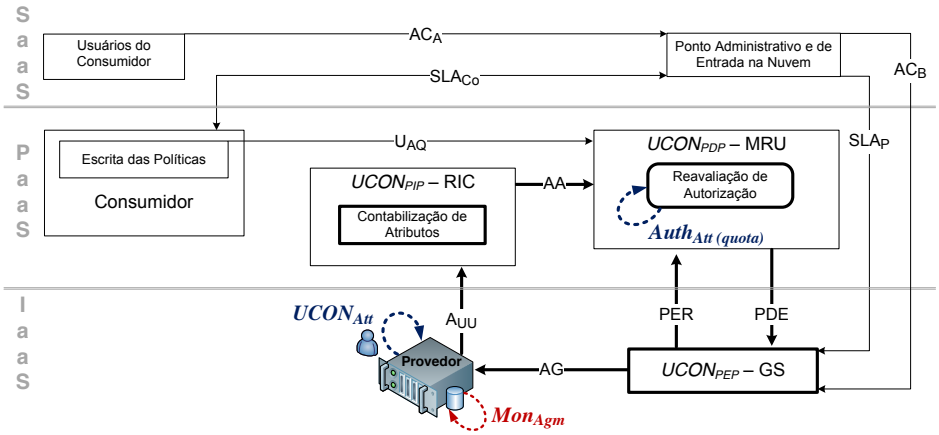

Figura 2. Modelo de autorização resiliente $UCON_{ABC}$.

O trabalho estende o modelo de autorização contínua do $UCON_{ABC}$, provendo resiliência ao processo de reavaliação das políticas de uso [Marcon Jr. *et al.*, 2013]. Resiliência, nesta abordagem, significa prover ao modelo a habilidade de tratar algumas situações de exceção que ocorrem com os atributos de autorização do usuário. Porém, mantendo as cotas de consumo do serviço utilizado dentro dos parâmetros estabelecidos no *SLA* (Figura 2; *evento* SLA_{CO}). As entidades envolvidas no suporte a resiliência são destacadas com linhas mais espessas na Figura 2.

O domínio consumidor escreve as políticas de uso para seus usuários (*evento* U_{AQ}), definindo os atributos de autorização individuais (*i.e.*, as cotas de uso de serviço).

A cota é utilizada durante o processo de reavaliação de políticas de controle de uso em substituição aos atributos de autorização. O objetivo deste esquema é flexibilizar as políticas de uso, quando possível, lidando com situações de exceção de autorização.

Atributos que contabilizam o consumo do usuário são obtidos do provedor através de agentes de monitoramento (*Agm*; *evento* A_{UU}; Figura 2). A resiliência para o processo de reavaliação de autorização contínua esta definida somente se o SLA_{CO} menos a soma que contabiliza todos os atributos de uso do usuário é maior que *t*. A constante *t* é uma cota reserva, livremente definida pelo consumidor para um determinado serviço. Isto significa que, quando a soma de consumo total dos usuários (*sctu*) estiver próxima do limiar definido por "SLA_{CO} - *t*", um novo *SLA* deverá ser negociado visando evitar uma violação de contrato.

Para facilitar o entendimento da proposta, um cenário envolvendo um sistema de arquivos é utilizado para explicar como a cota é utilizada para prover resiliência ao processo de autorização $UCON_{ABC}$. Neste contexto (Figura 2) a resiliência está representada como as linhas pontilhadas para os atributos de uso e de autorização. Considerando um consumidor que negocia um contrato (*evento* SLA_{CO}) para um serviço (*e.g.*, espaço de armazenamento de *600GB*, sendo *t = 100GB*). O consumidor escreve políticas de uso que definem *userA: 200GB*, *userB: 200GB* e *userC: 100GB*. Quando o usuário solicita acesso a um serviço da federação (*eventos* AC_A e AC_B), o guardião (*GS*) envia uma solicitação de avaliação de autorização para o monitor de referência $UCON_{ABC}$ (*MRU*; *evento PER*). O *MRU* configura a cota de consumo para o usuário (*i.e.*, inicialmente, o valor é igual ao atributo de autorização predefinido pelo consumidor: *userA: 200GB*, *userB: 200GB* e *userC: 100GB*), fornecendo permissão de consumo para o mecanismo que executa o controle de acesso (*evento PDE*).

Após ter o acesso liberado pelo guardião do serviço (*evento AG*), o usuário começa a utilizar o armazenamento. Posteriormente, o agente de monitoramento (*Agm*) envia os atributos de uso de cada usuário (*e.g.*, *userA: 190GB, userB: 10GB* e *userC: 0GB*) para o repositório de informações de contexto (*RIC*). Passado algum tempo, o guardião (*GS*) solicita uma reavaliação de autorização. Durante a reavaliação (autorização em tempo de execução - *ongoing*), o monitor de referência (*MRU*) compara os atributos de uso de cada usuário (*evento AA*) com a cota (*i.e.*, mecanismo que implementa a resiliência para o processo de autorização). Neste caso, a quantidade de armazenamento utilizada por cada usuário está abaixo das cotas predefinidas.

Um período de tempo depois, o agente envia novamente os atributos de uso de cada usuário (*e.g.*, *userA: 250GB, userB: 20GB* e *userC: 10GB*) para o repositório (*RIC*) e uma reavaliação de autorização é necessária. Durante a reavaliação, o *MRU* percebe que os atributos de uso do *userA* estão excedendo a cota predefinida (*i.e. userA: 200GB*). Neste momento, a condição de soma de consumo para os usuários (*sctu*) é de *280GB*. Como o "SLA_{CO} - *t*" admite *500GB* de armazenamento e o *userA* está excedendo o limite definido no atributo de autorização, a cota para este usuário em situação irregular é automaticamente expandida para *userA: 250GB*.

Com a continuidade da utilização o agente (*Agm*) envia os atributos de uso de cada usuário (*e.g.*, *userA: 240GB, userB: 160GB* e *userC: 100GB*) para o repositório (*RIC*). Enquanto isto, o guardião (*GS*) prossegue solicitando as reavaliações de autorização para o monitor de referência (*MRU*). Neste processo, o *MRU* percebe que os atributos de uso do *userA* estão abaixo da cota expandida, porém, a condição de soma de consumo para os usuários (*sctu*) é de *500GB* de armazenamento. Neste momento, o processo de reavaliação detecta que o espaço de armazenamento, subtraído da cota

reserva (SLA_{CO} - *t*) foi completamente utilizado. Adicionalmente, a cota do *userA* está excedendo a política de uso, sendo que a parcela de armazenamento extra deveria ser equiparada com o atributo de autorização (*i.e.* política de uso), *userA: 200Gb*. Se, na próxima reavaliação os atributos de uso do usuário *userA* se mantiverem além da cota, e a soma dos atributos (*sctu*) estiver próxima do SAL_{CO}, o *userA* vai estar em uma condição de exceção (*i.e.*, o *userA* está autorizado a consumir *200GB* mas os atributos de uso contabilizam um valor excedente, não sendo possível manter o esquema de resiliência). Caso contrário, a cota do *userA* poderia ser alterada novamente.

As condições de exceção ocorrem quando o usuário, durante uma reavaliação, é detectado violando uma cota (*i.e.* atributo de autorização) que autorizava seu acesso em uma avaliação prévia. A exceção acontece porque durante o período de reavaliação (atual e anterior), o usuário continua consumindo o serviço. Além disso, essa situação pode ser desencadeada pela resiliência do modelo, que redefine a cota para o valor original do atributo de autorização. Uma exceção também pode ocorrer se, entre os períodos de reavaliação, houver alguma mudança que torne a política atual mais restritiva do que a política que foi previamente definida. Quando um usuário é detectado em condição de exceção, o consumidor deve reescrever as políticas de uso visando trazer a situação do usuário ao estado normal, antes que a exceção seja alcançada. A condição de exceção pode levar o consumidor a concluir que, quando a *sctu* estiver próxima do limiar ($SLA_{CO} - t$), mais serviços precisam ser contratados (*i.e.* considerando a disponibilidade do provedor; *evento* SLA_P; Figura 2). Adicionalmente, a elasticidade fornecida pela nuvem pode ser utilizada em uma solicitação automática de serviços; está abordagem permite que a resiliência seja fornecida de maneira contínua.

Na abordagem tradicional de computação em nuvem, quando o usuário alcança o limite definido pelo atributo de autorização, não existe suporte para consumo excedente. Ou seja, o administrador do consumidor precisa pesquisar por usuários que não estão utilizando a cota predefinida e reescrever as políticas. Porém, esta é uma tarefa manual e cansativa, executada sem ter acesso aos atributos de uso reais (*i.e.* não existe suporte para auxiliar o administrador a decidir qual política de usuário poderia ser alterada). A aplicação do modelo de resiliência proposto fornece um equilíbrio automático entre o limite estabelecido na política e a quantidade contratada no *SLA*. Adicionalmente, o administrador precisará agir somente quando a soma dos atributos dos usuários (*sctu*) identificar que o limite definido no SLA_{CO} foi alcançado. O esquema de reavaliação provê resiliência ao processo de autorização, explorando a ociosidade na utilização dos serviços desde que *sctu* esteja abaixo do limiar ($SLA_{CO} - t$).

Nas abordagens tradicionais, a autorização é avaliada somente no inicio da utilização (*pre-authorization* $UCON_{ABC}$), porém isto pode gerar inconsistências entre a autorização concedida e a política de uso corrente. Mesmo no $UCON_{ABC}$ a reavaliação de autorização *ongoing* (durante o acesso) é insuficiente para garantir que nenhum consumo autorizado não esteja violando uma política. Isto pode acontecer porque as condições de exceção ocorrem entre os períodos de reavaliação. Nesta proposta, estes períodos são menores do que nas abordagens tradicionais, dependendo apenas do intervalo entre as reavaliações de autorização (seção 6.1).

A elasticidade fornecida pela nuvem é utilizada para adicionar automaticamente instâncias de monitores de referência e repositórios de informações de contexto. Estas tarefas ocorrem quando as demandas por reavaliações de autorização ou recuperação de atributos aumentam (seção 6.1). A abordagem atua como um *PaaS*, liberando o

consumidor do ônus de gerenciar aspectos de controle do ambiente para se concentrar nos esforços de desenvolvimento de serviços para sua atividade de negócio.

5.1. Arquitetura do Modelo

A arquitetura é baseada em uma nuvem formada por vários provedores e consumidores, sendo que cada um destes pode estar trocando dados com diferentes tipos de serviço (*e.g. SaaS*, *PaaS*). Um ambiente de intermediação (*i.e.* Ambiente Federado - *AF*; Figura 3) é utilizado para facilitar a interação entre as entidades e para prover um *PaaS* seguro e que empregue o modelo de autorização resiliente. O ponto de entrada do ambiente de nuvem para Provedores de Serviço (*PS*) e consumidores é fornecido pelo *Broker*. O *Broker* faz a intermediação da oferta de serviços dos provedores (*IaaS*), negocia os *SLAs* com os consumidores, e executa o redirecionamento dos usuários (usando balanceamento de carga) do Domínio Consumidor (*DC*) para o endereço do provedor definido no *SLA*.

O *PS* filia-se ao ambiente de nuvem federado através do *Broker*. Para cada serviço oferecido o provedor envia ao *Broker*: *a) SLA* – documento que define os itens acordados para o serviço. Este contrato é a fonte de informações para o *Broker* derivar os montantes que podem ser repassados aos consumidores; *b) descrição da interface do serviço* – documento fornecido ao usuário do consumidor para que este implemente seu sistema de acesso conforme as especificidades do serviço disponibilizado no provedor. O consumidor estabelece o *SLA* com o *Broker* e, de acordo com o montante de serviço contratado, define as políticas de uso para seus usuários. As regras de acesso são configuradas no ponto de administração de políticas (*PAP*; Figura 3) do Ambiente Federado (*AF*). Cada *SLA* disponibiliza para o consumidor um conjunto de serviços utilizados para gerenciar o ambiente (*e.g.*, serviços para armazenar atributos (*PIP*) e políticas). A administração do consumidor (*ADMC*) é a entidade responsável por escrever as credenciais de autenticação para seus usuários no Mecanismo de Segurança do Domínio Consumidor (*MSDC*). Estas credenciais permitem aos usuários do domínio interagirem com o *AF* e consumirem os serviços do provedor de nuvem.

Os usuários do *DC* recebem uma cota de uso que define os direitos para o consumo dos serviços. Antes do usuário utilizar sua cota, deverá enviar ao *Broker* sua credencial de autenticação (*i.e.* identificação assinada pelo consumidor). O *Broker*, usando o mecanismo de balanceamento de carga, direciona o usuário para um serviço específico. A solicitação de acesso recebida pelo provedor da federação é interceptada pelo guardião do serviço (*PEP*; Figura 3) e avaliada pelo monitor de referência (*PDP*) instanciado no Ambiente Federado (*AF*). A avaliação de acesso utiliza as informações de contabilização de consumo do usuário armazenadas pelo repositório de informações de contexto (*PIP*). Este serviço é atualizado por agentes de monitoramento (*AGM*) instanciados nos provedores de serviço.

O mecanismo de balanceamento de carga (*load balancer)* é utilizado como um portão (*gate*) de acesso para as instâncias de aplicações ativas no *PS*. A distribuição da carga pode ser feita de um modo simplista (*e.g.*, *round-robin*) ou de maneiras mais elaboradas (*e.g.*, utilizando o tempo de resposta do *host*, carga do sistema). O mecanismo pode detectar aplicações que não estão respondendo, evitando que novas requisições sejam enviadas para a mesma. A aplicação que atende a requisição é mapeada de modo transparente e em tempo de execução.

A atualização da lista de instâncias de aplicações disponíveis para atender requisições do cliente é feita após a inicialização de cada instância. A aplicação, após

ser corretamente configurada e inicializada, envia suas informações de contato (*e.g.*, endereço *IP* e porta de destino) em uma mensagem administrativa ao mecanismo de balanceamento de carga. Este mecanismo adiciona a nova instância ao esquema de controle interno e começa a encaminhar requisições para a mesma. Uma abordagem similar é utilizada no encerramento das instâncias, *i.e.*, uma mensagem é enviada ao balanceador de carga para que este remova a instância de sua lista interna. Neste contexto é possível observar que o *load balancer* pode ser utilizado para prover elasticidade automática ao esquema de provimento de serviços na computação em nuvem.

Após o consumidor configurar as políticas de uso, e o usuário iniciar a utilização do serviço, o *PIP* passa a armazenar informações de consumo referentes a cada usuário acessando o provedor de serviço. Periodicamente os atributos de consumo do consumidor são consolidados no *PIP* para avaliar se o SLA_{CO} do serviço está sendo respeitado. No *DC* os dados de consumo são analisados para decidir se as políticas de uso individuais precisam ser readequadas. Essa abordagem permite ao consumidor redefinir as regras de acesso conforme a necessidade de cada usuário, otimizando a utilização do serviço contratado e minimizar as exceções de autorização.

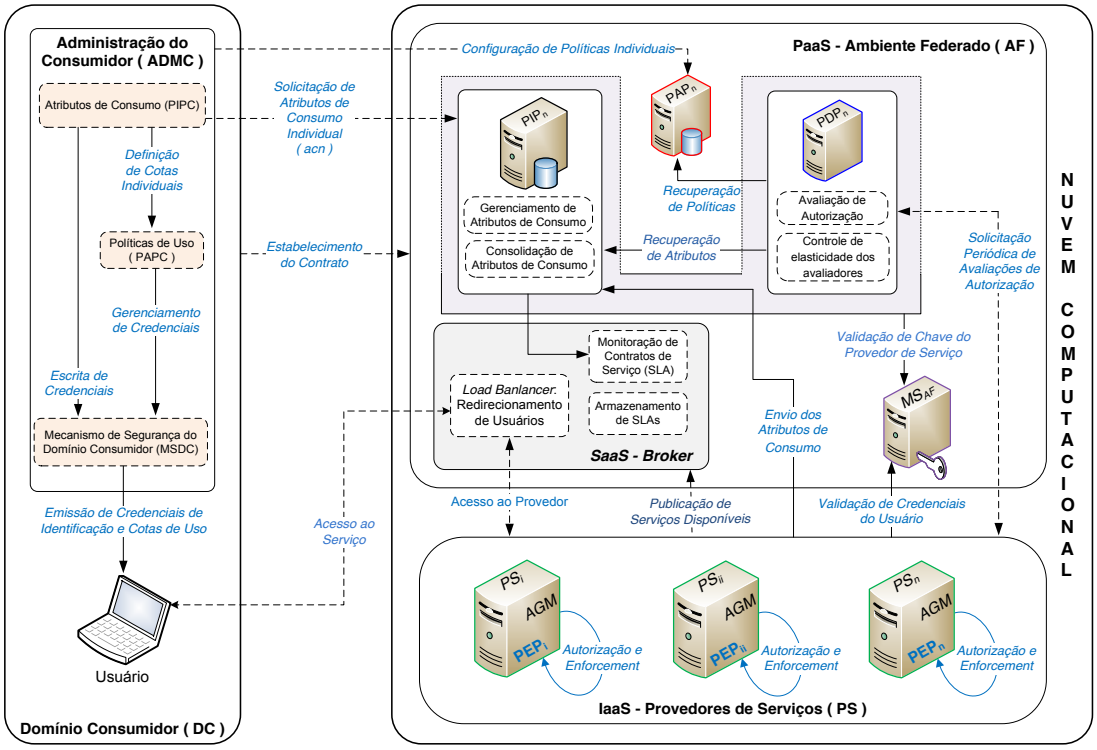

Figura 3. Visão geral da arquitetura proposta

Na abordagem desenvolvida, o ambiente de avaliação de políticas, baseado no modelo *outsourcing*, permite que a avaliação seja terceirizada para o ambiente distribuído externo. Considerando nossas avaliações anteriores, o modelo *outsourcing* se mostra mais adequado ao contexto da computação em nuvem devido à dinamicidade inerente a este ambiente (*i.e.* frequente instanciação e destruição de máquinas virtuais) [Marcon Jr *et al.*, 2009]. Nesta abordagem, nenhum tempo é gasto transferindo políticas e gerenciando sistemas de *cache* no ambiente do provedor para cada serviço instanciado. Pode-se perceber que, para cada serviço, as políticas para os usuários do consumidor deveriam ser configuradas no provedor. Se muitos serviços forem instanciados, os pedidos efêmeros de avaliação de política não justificam o custo de

transferir e gerenciar as políticas no ambiente provedor. Com a abordagem *outsourcing* e o armazenamento das regras no *PAP*, evitam-se as inconsistências causadas pelo *cache* de políticas no domínio do provedor.

O ambiente federado fornece interoperabilidade, alta disponibilidade e elasticidade, aplicando uma abordagem fracamente acoplada baseada em serviços *web*. Estes serviços são acessados por agentes de monitoramento para a escrita dos atributos de consumo referentes aos usuários. Adicionalmente, serviços *web* também são utilizados para armazenar as solicitações de reavaliação de autorização (enviada pelos *PEPs*) e as respostas das avaliações (envidas pelos *PDPs*).

O componente de elasticidade (expansão ou retração) do *framework* de computação em nuvem decide quando se faz necessária a criação de novas instâncias de aplicação, com base em políticas de elasticidade, isto exigirá a criação de uma nova Máquina Virtual (*MV*) para atender esta demanda. Este mesmo componente quando identifica a diminuição na demanda da aplicação monitorada pode, da mesma maneira, reduzir o número de instâncias de *MVs* em execução, a fim de reduzir o consumo de recursos da nuvem. Deste modo a aplicação é redimensionada dinamicamente, provendo a elasticidade característica dos ambientes de computação em nuvem.

Portanto, a elasticidade do ambiente de gerenciamento (*e.g.* *PIP*, *PDP*) é transparente, ocorrendo conforme a demanda do contexto (*i.e.* quantidade de provedores e consumidores fazendo uso do ambiente federado). Tendo em vista que cada consumidor necessita de serviços de gerenciamento para seu próprio domínio, quanto maior for o número de consumidores, maior será o *pool* de servidores para atender a demanda dos serviços *web*. A abordagem utilizada neste artigo fornece alta disponibilidade e elasticidade na parte de gestão de avaliação e contabilização.

5.2. Ambiente Federado e Provedores

O ambiente federado, representado pelo *Broker*, oferece para o usuário um ponto único de entrada para obter acesso aos serviços fornecidos pelos provedores. O *Broker* é a entidade responsável por administrar o *SLA* estabelecido com o consumidor. Quando um usuário necessita de um serviço, o *Broker* seleciona, entre os provedores da federação, aquele que melhor satisfaz as necessidades do consumidor usando o balanceamento de carga. O provedor de serviço é uma entidade associada à federação - o processo de filiação é gerido externamente ao contexto desta proposta. Uma vez associado, o provedor começa a usufruir a infraestrutura de gerenciamento do ambiente federado.

Na camada *PaaS* (Figura 5), os serviços de controle são consumidos pelo nível de *SaaS*, aplicando as políticas para cada pedido enviado por usuários pertencentes ao domínio consumidor. Quando o acesso é concedido na camada *PaaS*, o usuário está autorizado a utilizar os recursos sendo providos (camada *IaaS*). Adicionalmente, o modelo *outsourcing*, adotado nesta proposta, libera o provedor e o consumidor da tarefa de implementar o monitor de referência $UCON_{ABC}$ e o gerenciamento de atributos de consumo (*i.e.* estes serviços estão instanciados na federação). Com esta abordagem, o ponto de execução das políticas (*PEP*) e os agentes de monitoramento (*AGM*) podem ser empacotados juntamente com o sistema desenvolvido pelo consumidor.

5.3. Atributos

Os atributos pertencentes a camadas *IaaS* se referem ao total de recursos utilizados pela máquina virtual ou instância do serviço (Figura 4; *evento rsv*). Estes atributos são

referentes aos recursos alocados para o consumidor (*e.g. CPU-v*; *HD-v*; onde "*v*" representa recurso virtual). O *PaaS* fornece atributos referentes ao usuário que está consumindo o serviço (*e.g. Atr-Usuário*; *evento ua*). Os atributos de consumo manipulados pelo *PIP* são fornecidos por agentes (*AGM*) sendo executados nas máquinas virtuais dos provedores de serviço (*evento tp*). O *PIP* da federação fornece informações de consumo individual, referentes a cada usuário, para o *PIP* do consumidor (*PIPC*; *evento st*). Os atributos de consumo armazenados pelo *PIP* também são disponibilizados de modo consolidado para o *Broker* (*evento ar*) monitorar os *SLAs*.

Considerando os atributos de cada usuário (Figura 4, *evento st*) e o consumo consolidado (*evento cg*) é possível identificar qual usuário está utilizando os serviços no provedor (*evento ua*). Da mesma forma, é possível identificar se existem recursos ociosos no ambiente do provedor (*evento rsv*). Os atributos de consumo são fornecidos para a administração de políticas do consumidor (*PAPC*; *evento np*) para que esta possa gerar novas regras para seus usuários, justificar a necessidade de contratar mais serviços, ou otimizar a taxa de utilização dos recursos alocados no provedor. A partir dos atributos consolidados é possível executar um balanço automático das instâncias do serviço durante a reconfiguração das políticas.

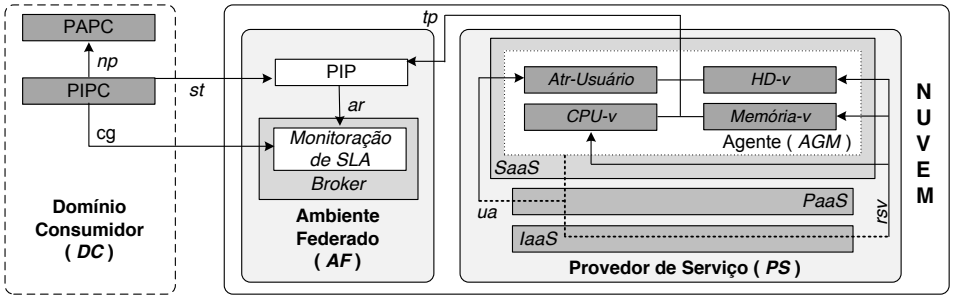

Figura 4. Gerenciamento de atributos de consumo

5.4. Controle de Uso e Gerenciamento de Políticas

A administração de políticas do consumidor (*PAPC*) e o mecanismo de segurança (*MSDC*; Figura 5) transformam o *SLA* em políticas de uso e credenciais de acesso (*evento re*) para os usuários do consumidor (*evento au*). Utilizando as credenciais, os usuários do consumidor acessam o Repositório de Interfaces (*RI*; *evento is*) do *Broker* (*evento* ac_a) e os serviços no provedor (*WS*; *evento* ac_b). As regras, derivadas do *SLA*, são armazenadas no *PAP* do Consumidor (*PAPC*), este serviço é utilizado para gerenciar as políticas de uso internamente ao domínio. Estas regras serão transferidas para o ambiente federado e configuradas no *PAP* da federação (*evento en*). Posteriormente, estas políticas serão utilizadas pelo *PDP* (*evento rp*) para avaliar as solicitações de autorização. Estas solicitações são enviadas pelo *PEP* (*evento av*) sendo executado na máquina virtual do provedor de serviço. As políticas de uso armazenadas no *PAP* também são utilizadas para configurar a cota do usuário.

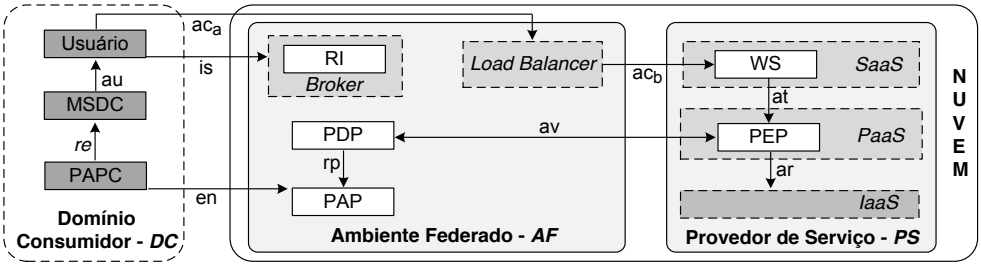

Figura 5. Gerenciamento de políticas de uso

Com a utilização do modelo *outsourcing*, o usuário precisa ser autorizado pelo *PDP* e ter o acesso liberado pelo *PEP* (*evento at*) para poder consumir o serviço da camada *IaaS* (*evento ar*). O processo de avaliação de autorização é executado no *PDP* da federação (*evento av*) utilizando as políticas recuperadas do *PAP* (*evento rp*). Quando a resposta da avaliação é positiva, o usuário está autorizado a consumir o serviço disponibilizado pela camada *IaaS* (*evento ar*), decisão esta que será honrada pelo *PEP*. As solicitações de acesso recebidas pelo guardião do serviço (*evento at*) são encaminhadas para os serviços *web* da federação. Um serviço *PDP* pertencente ao *pool* de servidores avalia a solicitação recebida e responde ao *PEP* solicitante. A avaliação utiliza os atributos de consumo disponíveis no repositório de atributos (*PIP*; Seção 5.3).

6. Protótipo e Testes de Avaliação

Um protótipo foi implementado como prova de conceito utilizando as seguintes tecnologias: *framework* de computação em nuvem *OpenNebula* (*opennebula.org*), versão 4.2 juntamente com o mecanismo de balanceamento de carga – *proxy HTTP Perlbal* (*github.com/perlbal*). Os atributos de consumo foram obtidos com o auxilio dos projetos *JavaSysMon* (*jezhumble.github.com/javasysmon*), *SIGAR* (*www.hyperic.com/support/docs/sigar*) e *API* de acesso a *Java Virtual Machine (JVM - java.lang.management*). O controle de consumo utiliza políticas *XACML* [OASIS, 2005] manipuladas pela *API Sun XACML* (*http://sunxacml.sourceforge.net*). Os serviços fornecidos pelas maquinas virtuais estão hospedados no *Apache TomCat* (*http://tomcat.apache.org*), sendo acessados através da *engine SOAP Axis2* (*http://axis.apache.org*) e módulo *Rampart* (*http://axis.apache.org/axis2/java/rampart*) integrado ao *Axis2* – que protege as mensagens *SOAP* [W3C, 2007] (especificações *WS-Security* [OASIS, 2004] e *WS-Trust* [OASIS, 2007]).

O protótipo foi avaliado nas seguintes configurações de ambiente: o domínio consumidor foi executado em um *desktop Intel Core i7* de *2Ghz*, *6GB* de *RAM* com *Windows 7 x64*; o servidor que hospeda os provedores de serviço utiliza um *Intel Xeon Dual Processor*, *6 cores* de *2.6Ghz*, *48GB* de *RAM* e *3TB* de disco e o ambiente federado emprega um servidor *Intel Xeon Dual Processor*, *4 cores* de *2.0Ghz*, *16GB* de *RAM* e *3.3 TB* de disco. As máquinas estão conectadas em uma rede local *Gigabit Ethernet*.

6.1. Testes e Avaliação

No *framework OpenNebula* foi utilizado um componente chamado *OneGate*, que permite o monitoramento de máquinas virtuais periodicamente, afim de detectar problemas em aplicações, coleta de métricas e desencadeamento de regras de elasticidade. A elasticidade foi controlada por outro componente, chamado *OneFlow*. As aplicações foram definidas através de um documento que especifica dependências de serviço e regras de elasticidade.

As regras de elasticidade definiram o número mínimo de instâncias que devem ser executadas, assim como o número máximo de instâncias que podem ser criadas. A elasticidade é acionada por expressões que utilizam as métricas fornecidas pelo componente *OneGate*. Um período de tolerância foi aplicado (*i.e.*, duração em que a expressão deve ser verdadeira para que o gatilho seja ativado), evitando instabilidades no sistema decorrente da inicialização ou encerramento de instâncias ao primeiro sinal de mudança no ambiente.

O sistema operacional *linux* da máquina virtual, após seu procedimento de inicialização, executa a aplicação e comunica o balanceador de carga (*Perlbal*), através de sua interface administrativa, o novo endereço disponível. Ao iniciar seu procedimento de encerramento, o sistema operacional comunica ao balanceador a necessidade de remover o endereço da aplicação de sua lista de aplicações disponíveis.

O serviço *web* [W3C, 2004] oferecido ao usuário é provido como um *e-commerce*. Neste cenário, cada agente de monitoramento (*AGM*) enviou *17* tipos de atributos diferentes para o serviço *web*. As principais informações enviadas para o serviço são: endereço de IP (*Internet Protocol*) e o nome da máquina virtual que hospeda o serviço *web* disponibilizado para os usuários; quantidade de memória *heap* gerenciada pela máquina virtual *Java* (*i.e.* valores: inicial, utilizado, reservado e o máximo disponibilizado); identificação do agente de monitoramento e da *thread* responsável por executar o serviço solicitado pelo usuário (*i.e.* nome e *id* da *thread*); tempo gasto pelo agente para monitorar os atributos do sistema; tempo gasto pela *thread* de serviço para atender uma solicitação do usuário e para o sistema responder a solicitação (*overhead*); informações relativas a máquina virtual que hospeda o serviço *web* utilizado nos testes (*e.g.* taxas de utilização referentes ao processador, memória e rede).

Os valores apresentados (Tabelas 1 e 2) foram obtidos executando-se *1000* interações e calculando-se a média entre estes – o coeficiente de variabilidade ficou abaixo de *5%* em todos os casos. Os testes foram executados em ambiente controlado de rede local visando avaliar o comportamento do sistema sem interferências externas. Os testes executados no serviço responsável por armazenar os atributos do ambiente (Tabela 1) visam identificar o tempo médio de resposta e a quantidade de agentes que este serviço *web* suporta atender. Pode-se perceber que o tempo gasto para armazenar estas entradas varia significativamente, apresentando a melhor média para mensagens com tamanho de *2KB* a *16KB* para atributos de consumo [Marcon Jr *et al.*, 2013]. No caso dos testes executados, o agente de monitoramento enviou entradas com cerca de *2KB* (*17* tipos de atributos de consumo).

Tabela 1: Armazenamento de Atributos

Tamanho da entrada enviada para o *WS PIP*	Média de tempo de resposta para armazenar uma entrada no *WS PIP* (*seg*)	Quantidade de agentes atendidos paralelamente pelo *WS PIP* (~)
2 KB	0,75	1030
4 KB	0,76	810
8 KB	0,77	590
16 KB	0,79	510
32 KB	0,85	450
64 KB	0,97	430
128 KB	1,23	370
256 KB	1,51	340
512 KB	1,78	310
1024 KB	1,88	290
2048 KB	2,01	150

Tabela 2: Reavaliação de Políticas

Tamanho da política utilizada pelo *WS*	Média do tempo de resposta para o cliente (*ms*)	Média de tempo para avaliar uma política no *WS PDP* (*ms*)	Quantidade de *WS PEPs* atendidos paralelamente pelo *WS PDP* (~)
2 KB	0,34	0,29	890
4 KB	0,34	0,29	810
8 KB	0,34	0,29	770
16 KB	0,34	0,30	660
32 KB	0,34	0,30	580
64 KB	0,35	0,30	520
128 KB	0,35	0,30	430
256 KB	0,36	0,32	320
512 KB	0,39	0,35	270
1024 KB	0,50	0,45	190
2048 KB	0,58	0,54	140

Um teste análogo foi executado no *PDP* referente a reavaliação de políticas (Tabela 2). Neste cenário, o *PDP* recebe a solicitação de avaliação (com aproximadamente *2KB*, considerando um cabeçalho com cerca de *1KB*) e recupera uma política *XACML* do *PAP* (*i.e.* um arquivo contendo uma regra de autorização com *2KB*). O *PDP* avalia a solicitação e envia a resposta para *PEP* solicitante (tamanho aproximado de *2KB*, considerando um cabeçalho de *1KB*). Este experimento mostra a capacidade do *PDP* para atender solicitações de avaliação de política (Tabela 2;

Quantidade de WS PEPs Atendidos paralelamente pelo WS PDP) e o tempo gasto para executar esta tarefa internamente ao *PDP* (Tabela 2; *Média de tempo para avaliar uma política no WS PDP*). O usuário do consumidor necessita esperar alguns milissegundos a mais para obter a resposta referente a solicitação de acesso interceptada pelo *PEP* (Tabela 2; *Média do tempo de resposta para o cliente*).

As medidas apresentadas nas Tabelas 1 e 2 fornecem uma noção quanto ao "gatilho de elasticidade" da proposta, indicando que o número de servidores no *pool* deveria ser incrementado quando a demanda estiver próxima de provocar uma parada nos serviços. A abordagem de serviços *web*, juntamente com o esquema de avaliação distribuído (*i.e.* vários *PDPs* instanciados conforme a demanda) fornece elasticidade ao ambiente de avaliação $UCON_{ABC}$. O espaço de gerenciamento de atributos, responsabilidade do *PIP*, segue a mesma abordagem, fornecendo elasticidade ao sistema de contabilização de consumo acessado pelo $UCON_{ABC}$.

7. Conclusões

O trabalho apresentou uma abordagem inovadora para a reavaliação de autorização contínua em controle de uso. O monitoramento constante de serviços e a reavaliação dos atributos de autorização permitem a identificação de disparidades entre autorizações e políticas. Adicionalmente, o esquema provê resiliência (*i.e.*, relaxamento das regras da política) para os atributos de autorização em algumas circunstâncias, sem qualquer perda para o consumidor (*e.g.*, violação de *SLA*). Quando uma disparidade é identificada e o esquema de resiliência não é possível, o usuário estará em condição de exceção. Neste caso, o consumidor possui algumas alternativas para reparar a situação, o que não acontece nas abordagens tradicionais apresentadas pela literatura.

O serviço de contabilização proposto e a reavaliação contínua, para cada usuário, provê um esquema de monitoramento e controle de acesso com fina granularidade, utilizados em ambiente de nuvem. A resiliência aplicada neste trabalho tornou os atributos de autorização (cotas) definidos para cada usuário do consumidor mais flexíveis. Adicionalmente, violações em políticas de controle são monitoradas e tratadas pelo ambiente de gerenciamento em nível de federação (*SLAs*) e de consumidor (condições de exceção). Este esquema permite a utilização flexível dos recursos computacionais sem desperdício, ociosidade ou abuso dos serviços contratados.

A abordagem proposta mostrou que é possível executar o gerenciamento e a consolidação de atributos utilizando padrões abertos (*e.g.*, Serviços *Web*). Adicionalmente, o esquema é adequado para o nível de acesso permitido atualmente para a camada de infraestrutura (*IaaS*), não necessitando de mudanças neste ambiente. O gerenciamento é executado por serviços que trabalham de acordo com a demanda do consumidor. Isto significa que, sem o auxílio apresentado, o consumidor não poderia controlar o uso dos serviços e recursos na nuvem, tendo em vista que nenhum provedor de *IaaS* oferece um serviço similar.

Referências

Bertram S., Boniface M., Surridge M., Briscombe N. e Hall-May M. (2010). *On-Demand Dynamic Security for Risk-Based Secure Collaboration in Clouds*. 3rd CLOUD, pg. 518-525.

CSA (2011). *Security Guidance for Critical Areas of Focus in Cloud Computing v3.0*. Disponível em: *cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf*. Acesso: Mar. 2012.

Danwei C., Xiuli H. e Xunyi R., (2009). *Access Control of Cloud Service Based on UCON*. *1st CloudCom* 2009. LNCS. pg. 559-564.

Emeakaroha V. C., Netto M. A. S., Calheiros R. N., Brandic I., Buyya R. e Rose C. A. F. De. (2011). *Towards Autonomic Detection of SLA Violations in Cloud Infrastructures*. *Elsevier* FGCS, pg. 1-13.

Erickson J., Spence S., Rhodes M., Banks D., Rutherford J., Simpson E., Belrose, G. e Perry R. (2009). *Content-Centered Collaboration Spaces in the Cloud.* IEEE *Internet Computing*, pg. 34-42.

Grobauer B., Walloschek T. e Stöcker E. (2010). *Understanding Cloud-Computing Vulnerabilities. IEEE S&P*, pg. 50-57.

Hayes B. (2008). *Cloud computing*. *Communications* ACM, vol. 51, no. 7, pg. 9-11.

Lim H. C., Babu S., Chase J. S. e Parekh S. S. (2009). *Automated Control in Cloud Computing: Challenges and Opportunities. 1st* ACDC, pg. 13-18.

Marcon Jr. A. L., Santin A. O., Lima Jr. L. A. de P., e Stihler M. (2009). *Policy Management Architecture Based on Provisioning Model and Authorization Certificates*. *ACM* SAC, pg. 1594-1598.

Marcon Jr, A. L., Laureano, M., Santin, A. O. e Maziero, C. A. (2010). *Aspectos de Segurança e Privacidade em Ambientes de Computação em Nuvem*. *Anais de MiniCursos do SBSeg 2010, SBC,* pg. 53-102.

Marcon Jr, A. L., Santin, A. O., Stihler, M. e Bachtold, J. (2013). *A UCONabc Resilient Authorization Evaluation for Cloud Computing*, *IEEE Transactions on Parallel and Distributed Systems, 11 April 2013. IEEE computer Society Digital Library*, *Preprint.*

Mell P. e Grance T. (2009). *The NIST Definition of Cloud Computing. Special Publication* 800-145. *National Institute of Standards and Technology* (NIST)*, Information Technology Laboratory*. Disponível em: *http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf*. Acesso: Jan. 2013.

OASIS (2004). *Web Services Security SOAP Message Security 1.1*. Disponível em: *docs.oasis-open.org/wss/v1.1*. Acesso: Nov. 2012.

OASIS (2005). *eXtensible Access Control Markup Language v 2.0*. Disponível em: *www.oasis-open.org/committees/xacml*. Acesso: Out. 2011.

OASIS (2007). *WS-Trust 1.3*. Disponível em: *www.oasis-open.org/standards#wstrustv1.3*. Acesso: Set. 2012.

Park J. e Sandhu R. (2004). *The $UCON_{ABC}$ Usage Control Model*. *ACM TISSEC*, vol. 7, no. 1, pg. 128-174.

Yavatkar R., Pendarakis D. e Guerin R. (2000). *A Framework for Policy-based Admission Control*, RFC 2753.

Tavizi T., Shajari M. e Dodangeh P., (2012). *A Usage Control Based Architecture for Cloud Environments*. *IEEE IPDPSW 2012*, *pg. 1534-1539*.

Teigão R., Maziero C. e Santin A. O. (2011). *Applying a Usage Control Model in an Operating System Kernel*. *Elsevier Journal of Network and Computer Applications*, pg. 1342-1352.

W3C (2004). *Web Services Architecture*. Disponível em: *www.w3.org/TR/ws-arch*. Acesso: Jun. 2011.

W3C (2007) *SOAP Version 1.2*. Disponível em: *www.w3.org/TR/soap*. Acesso: Set. 2012.

Zhang L. J. e Zhang J. (2009). *An Integrated Service Model Approach for Enabling SOA*. *IEEE* IT Pro. pg. 28-33.

