risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

Recebido / Recibido: 23/04/2012
Aceitação / Aceptación: 30/05/2012

Modelo de segurança para a composição dinâmica de *workflows* em arquiteturas de *e-government*

Fábio Marques [1,4], Gonçalo Paiva Dias [1,3], André Zúquete [2,4]

fabio@ua.pt, gpd@ua.pt, andre.zuquete@ua.pt

[1] Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro, Rua Comandante Pinho e Freitas, nº28, 3750-127, Águeda, Portugal

[2] Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, Campus Universitário de Santiago, 3810-193, Aveiro, Portugal

[3] Unidade de Investigação em Governança, Competitividade e Políticas Públicas, Campus Universitário de Santiago, 3810-193, Aveiro, Portugal

[4] Instituto de Engenharia Eletrónica e Telemática de Aveiro, Campus Universitário de Santiago, 3810-193, Aveiro, Portugal

DOI: 10.4304/risti.9.15-26

Resumo: As arquiteturas de interoperabilidade permitem a criação de *workflows* transversais na administração pública e a integração de serviços na perspetiva dos cidadãos e empresas. Neste artigo apresentamos um modelo de segurança que visa as questões levantadas por arquiteturas de interoperabilidade baseadas em agentes autónomos que suportam a composição dinâmica de *workflows*. O modelo baseia-se numa infraestrutura de chave pública e num conjunto de estruturas de dados baseadas em normas bem conhecidas (X.509 V3 e WSDL). Este modelo de segurança suporta a identificação, autenticação, acreditação e autorização e garante que os resultados produzidos pelos agentes apenas são entregues aos seus destinatários, mesmo que estes destinatários não sejam conhecidos na altura da produção do resultado.

Palavras-chave: e-government; Segurança; Workflows dinâmicos; Privacidade; Interoperabilidade.

Abstract: Interoperability architectures allow the creation of transversal workflows in the public administration and the integration of services from the perspective of citizens and businesses. In this paper we present a security model to address the security issues that are raised by an interoperability architecture that supports the dynamic composition of e-government workflows by autonomous agents. The model is based in a Public Key Infrastructure and a set of data structures which are supported on well-known standards (X.509 V3 and WSDL). It addresses agent identification, authentication, accreditation and authorization and ensures that results produced by agents are privately delivered to their intended recipients even though those recipients may not be known when the results are produced.

Keywords: e-government; Security; Dynamic workflows; Privacy; Interoperability.

1. Introdução

A Organização para a Cooperação e Desenvolvimento Económico (OCDE) define e-government como "A utilização das Tecnologias da Informação e da Comunicação em atividades de governo" (OECD, 2001). As arquiteturas de interoperabilidade constituem uma das mais importantes aplicações das Tecnologias da Informação e da Comunicação (TIC) no governo. Estas arquiteturas suportam a partilha de informação entre ramos da administração pública, promovendo a eficiência e permitindo a integração de serviços na ótica dos cidadãos e empresas.

Em (Marques, Dias & Zúquete, 2011) foi apresentada uma arquitetura de interoperabilidade para e-government que segue a composição de serviços como abordagem, é mutável, adaptável, versátil e segura. Esta arquitetura é intrinsecamente dinâmica, permite a criação e remoção de novos serviços e de prestadores de serviço em qualquer altura. Devido ao seu dinamismo, não se aplica o paradigma comum de todos os prestadores de serviços serem confiáveis e autorizados para interagir com todos os restantes prestadores de serviços: um sistema de segurança dinâmico é essencial. Neste artigo apresentamos um modelo que, baseando-se na Tecnologia de Infraestrutura de Chave Pública (PKI), permite a criação dinâmica de esquemas de verificação de segurança no *e-government*.

O artigo está organizado da seguinte forma: começamos por introduzir o problema na presente secção; o trabalho relacionado é abordado na secção 2; na secção 3 fazemos uma breve introdução à arquitetura de interoperabilidade baseada em agentes; na secção 4 apresentamos o modelo de segurança; seguindo-se na secção 5 a discussão; o artigo é concluído na secção 6.

2. Trabalho Relacionado

A segurança tem sido uma das maiores preocupações no desenvolvimento de plataformas baseadas em agentes e em sistemas de *workflow*.

As características das plataformas de agentes e o seu cariz genérico conduziram ao desenvolvimento de vários modelos de segurança. Por exemplo, em (Stormer, Knorr & Eloff, 2000) os autores propuseram uma aproximação baseada em RBAC (*Role Based Access Control*) para autorização e em SoD (*Separation of Duties*) para a aplicação da integridade. Em (Savarimuthu, Purvis & Oliveira, 2004) os autores utilizaram uma PKI para suportar a autenticação de agentes e impuseram autorização RBAC através da utilização de hierarquias suportadas por PKI (cada Autoridade Certificadora pertence a uma sociedade diferente e cada sociedade representa uma função na plataforma). Em (Kannammal & Iyengar, 2008) é utilizado um *Key Server* que atua como um elemento de confiança comum para armazenar as chaves do *Launcher Agent* e dos agentes móveis. As chaves e o *Key Server* têm um papel central no modelo de segurança, permitindo autenticação aos agentes móveis, autenticação interdomínio e gestão da confiança do domínio.

Diversos modelos de segurança para *workflows* têm sido igualmente propostos ao longo dos anos e, como iremos ver, muitos deles são baseados em RBAC. No entanto isto não é o caso do modelo de autorização apresentado em (Hung & Karlapalem, 2003). Este modelo é suportado por um conjunto de funções de autorização que são executadas em diferentes camadas (*workflow*, controlo e dados) da máquina de estados multicamada que controla o modelo, disponibilizando ou negando o acesso aos diferentes recursos.

Em (Chou & Wu, 2004), Chou e Wu apresentaram um modelo de controlo de acesso baseado em RBAC – WfRBAC (*Role-based access control within workflows*) – que resolve algumas das limitações do modelo RBAC durante o tempo de execução do *workflow* (Chou & Wu, 2004): troca dinâmica de função, gestão de associação de função; e, prevenção indireta de fuga de informação. No modelo WfRBAC, para controlar o acesso à informação do *workflow* durante a sua execução, uma politica de controlo de acesso é embutida no *workflow*. Isto aborda as limitações identificadas do modelo RBAC.

Dois modelos baseados em RBAC são apresentados em (Wainer, Barthelmess & Kumar, 2003), sendo ambos conhecidos por W-RBAC. O primeiro modelo, W0-RBAC junta um serviço de permissões baseado em RBAC e uma componente *workflow*. O serviço de permissões providencia uma linguagem baseada em lógica que permite a definição de utilizadores que podem ser autorizados para realizar tarefas. O segundo modelo – W1-RBAC – adiciona a capacidade de tratamento de exceções ao primeiro modelo.

Em (Atluri & Huang, 1996), o WAM (*Workflow Authorization Model*) é apresentado. Este modelo suporta a especificação de políticas de autorização de acesso que permitem o acesso durante a execução de uma tarefa. A sincronização necessária do fluxo de autorização com o workflow é atingida através da utilização de um modelo de autorização que é associado a cada tarefa que integra o workflow.

Um modelo TBAC (*Task-Based Access Control*) foi apresentado em (Thomas & Sandhu, 1997). Este modelo associa autorização com tarefas. Todas as funções que são aceites para uma tarefa são reunidas num conjunto de confiança. De cada vez que um passo de autorização é executado uma função é escolhida deste conjunto de confiança.

3. A Arquitetura de Interoperabilidade Baseada em Agentes

Nesta secção apresentamos sucintamente a arquitetura baseada em agentes, de forma a contextualizar a nossa contribuição.

3.1. Visão Global

O conceito base da arquitetura é bastante simples: ela é composta por agentes que trabalham em conjunto para prestar serviços. Estes serviços estão registados num repositório de serviços e são publicados pelos agentes que os oferecem. Aplicam-se os seguintes pressupostos:

- A interoperabilidade é conseguida através da composição de serviços simples, que são fornecidos por autoridades públicas, para produzir serviços complexos, que são consumidos por outras autoridades públicas;
- Todos os serviços são prestados por agentes. Os serviços podem ser simples ou complexos. Serviços simples são prestados por agentes associados a um Sistema de Informação Local (SIL) das agências que efetivamente fornecem os serviços. Serviços complexos são prestados por agentes que compõem estes serviços simples;
- A gestão do *workflow* (i.e., o processo de prestação de um serviço) de um serviço complexo não está centralizada em nenhum agente. Os agentes têm a possibilidade de delegar a gestão de partes do *workflow* a outros agentes, invocando novos serviços. Isto implica que os *workflows* são estabelecidos dinamicamente, pelo que não existe um conhecimento prévio dos agentes que participam no *workflow* do serviço complexo;
- O resultado produzido por um agente no decorrer do *workflow* é mantido no agente até ser explicitamente requisitado por outro agente que necessita de o consumir. Este comportamento é imposto por requisitos de confidencialidade. Uma vez que os agentes que necessitam de consumir resultados podem não ser conhecidos na altura da sua produção, os resultados são mantidos pelos agentes que os produzem até que estes sejam requeridos por agentes autorizados para o fazer. De qualquer forma, os agentes têm conhecimento da localização dos resultados de que precisam, uma vez que a informação sobre a disponibilidade do resultado, mas não o seu valor, é transmitida através de todos os agentes que participam no *workflow*.

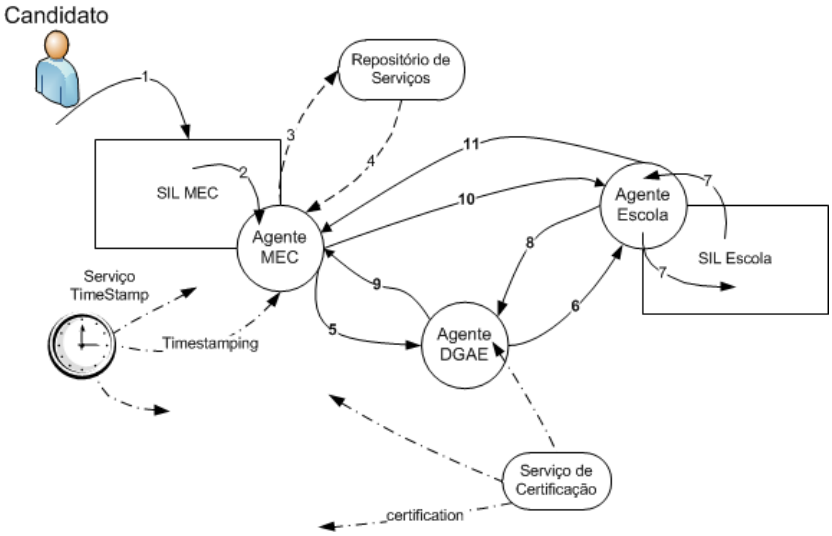

Figura 1 – Representação da arquitetura base e da prestação genérica de serviços

A Figura 1 fornece um exemplo da utilização da arquitetura: candidatura ao ensino superior em Portugal. Ocorrem as seguintes interações:

1. O candidato acede ao Ministério da Educação e Ciência (MEC) para fazer a sua candidatura ao ensino superior e seleciona os cursos e instituições para as quais se quer candidatar;
2. O Sistema de Informação Local do MEC solicita as notas do candidato ao agente que lhe está associado;
3. O agente encontra o serviço da Direção Geral da Administração Escolar (DGAE) que pode fornecer as notas dos alunos;
4. O agente obtém a informação de que necessita para solicitar o serviço;
5. O agente do MEC solicita o serviço ao agente da DGAE;
6. Por sua vez, o agente da DGAE, que não consegue responder diretamente ao serviço mas tem conhecimento do agente que lhe poderá responder, reencaminha o pedido (ou parte dele) para o agente da escola;
7. O agente da escola solicita as notas ao SIL a que está associado;
8. O agente da escola satisfaz o serviço enviando um URI que define a localização das notas do aluno para o agente da DGAE;
9. O agente da DGAE reencaminha a informação para o agente do MEC;
10. Uma vez que o MEC necessita do resultado para concluir o serviço inicialmente solicitado, envia um pedido para o agente da escola a pedir o resultado do serviço;
11. A satisfação deste último pedido conclui o serviço pedido originalmente. De notar que o agente da escola, aquando da produção do resultado, não sabe quem será o destinatário final das notas do aluno. Esse destinatário só fica a ser conhecido quando o agente do MEC solicita explicitamente o resultado produzido.

3.2. Mensagens

Durante a execução do *workflow* de um serviço, várias mensagens são trocadas entre os agentes que estão envolvidos no processo. Os tipos de mensagem são: *Service Request*; *Notification*; *Result Request*; *Result Delivery*.

Uma mensagem do tipo **Service Request** ((5) e (6) na Figura 1) é uma mensagem que suporta toda a informação necessária para solicitar um serviço. Consiste na descrição do serviço a ser pedido, na identificação do pedido, no timestamp correspondente à hora em que o serviço foi pedido, na assinatura do solicitador, da identificação do destinatário e da identificação do solicitador original. Em determinada altura no tempo, este tipo de mensagem pode conter blocos com o mesmo tipo dele próprio (correspondendo à composição do serviço), ou seja incluindo informação sobre todos os serviços mais simples que foram prestados (e dos agentes que os executaram) até àquele momento.

Toda e qualquer alteração no estado de uma prestação de serviço produz uma mensagem do tipo **Notification** ((8) e (9) na Figura 1). Deste modo, Notifications são enviadas dos agentes que executaram um serviço para os agentes que o tinham solicitado. A mensagem contém informação sobre o serviço que foi pedido, sobre o agente que está a prestar o serviço e sobre o novo estado da prestação do serviço (e.g. a conclusão da prestação do serviço com a produção de um resultado).

Uma mensagem do tipo **Result Request** ((10) na Figura 1) é utilizada quando um agente necessita de aceder a um resultado que foi previamente obtido por outro agente. O valor do resultado é identificado por um URI que foi gerado na altura pelo agente que produziu o resultado. O URI do valor do resultado e a Informação sobre o solicitador, nomeadamente os seus certificados, estão incluídos na mensagem.

Mensagens do tipo **Result Delivery** ((11) na Figura 1) são geradas como resposta ao Result Request. Para além do resultado, contém informação sobre o agente que entrega o resultado e os seus certificados.

3.3. Estruturas de Suporte

De forma a obter-se uma prestação de serviços segura e localizar os serviços disponibilizados, a arquitetura contém algumas infraestruturas de suporte, nomeadamente Repositórios de Serviços e uma Infraestrutura de Chave Pública.

Os Repositórios de Serviços respeitam o padrão *Universal Description, Discovery and Integration* (UDDI) (Clement, Hately, Riegen & Rogers, 2004). Estes repositórios armazenam informação sobre todos os serviços disponibilizados, incluindo a identificação do agente que realiza o serviço e os resultados produzidos. Esta informação é armazenada na estrutura (*Service Description*) baseada na linguagem *Web Services Description Language* (Christensen, Curbera, Meredith & Weerawarana, 2001) (WSDL).

A Infraestrutura de Chave Pública é utilizada para suportar a autorização de agentes (que exige a identificação e autenticação dos agentes), a acreditação dos pares do tipo {agente, serviço} e a assinatura digital das mensagens e dos resultados produzidos.

4. O Modelo de Segurança

Nesta secção identificamos as questões de segurança que resultam da arquitetura e apresentamos o modelo de segurança para os enfrentar.

4.1. Questões de Segurança na arquitetura

A arquitetura baseada em agentes tem um conjunto de questões de segurança que deve ser resolvido.

Primeiro, um agente, como um recetor de pedidos, é abordado de uma de duas formas: através de um *Service Request* ou através de um *Result Request* (ver Figura 2). Ambos os tipos de pedidos podem ser realizados por qualquer agente dentro da arquitetura. Uma vez que não existem restrições no que diz respeito ao acesso à arquitetura, estes pedidos podem também estar acessíveis a qualquer peça de *software* que consiga encontrar os *Web Services* do agente. De forma a proteger o SIL associado ao agente

que disponibiliza o serviço, o agente deve verificar a autorização do solicitador do serviço.

Segundo, o caminho do *workflow* pode forçar algumas limitações à prestação do serviço, por exemplo: conflitos de interesse entre duas entidades que estão envolvidas no mesmo processo de prestação do serviço. Os agentes devem estar preparados para atuar (verificar a autorização do *workflow*) de acordo com estes casos e responder de forma apropriada.

Terceiro, existem algumas preocupações que devem ser abordadas pelo solicitador do serviço ou resultado. Para pedir um serviço, é necessário verificar se o agente a quem o mesmo será solicitado está acreditado para o prestar (o que significa que uma terceira entidade de confiança confirma que um agente não só é capaz de prestar um serviço mas também tem a jurisdição/qualificação necessária para o fazer). Neste caso (solicitar um serviço), o agente solicitador deve ser capaz de identificar qualquer restrição ao nível do *workflow* que possa impedir a participação de outro agente no *workflow*.

Finalmente, devem ser tidas igualmente algumas precauções aquando da solicitação de um resultado. Uma vez que um agente não sabe para quem está a produzir um resultado no momento em que o produz, então ele é incapaz de explorar as transformações de cifragem/decifragem de forma a assegurar a confidencialidade e a privacidade do mesmo quando em trânsito por outros agentes. Uma vez que este é o caso por omissão na arquitetura, então o agente mantém o resultado produzido até este ser explicitamente requisitado pelo agente que necessita dele. Esta necessidade para solicitar o resultado adiciona algumas preocupações ao agente que dele necessita: primeiro, o resultado é válido? (que quer dizer: o autor do resultado tem jurisdição sobre aquele resultado? Está este agente acreditado para prestar o serviço que deu origem ao resultado?); segundo, se o agente que requer o resultado tem de produzir um novo resultado com base neste resultado anterior, então tem de identificar o agente que originalmente providencia o resultado (é importante para responsabilização).

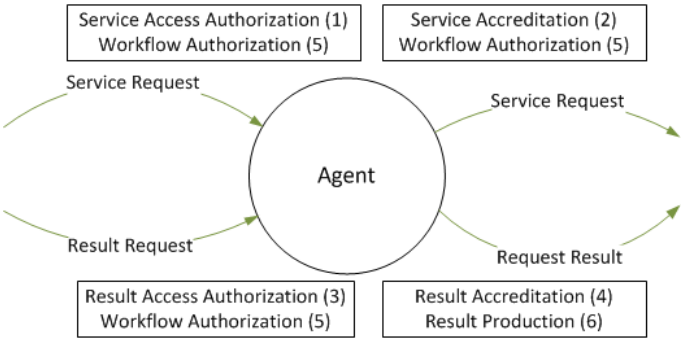

Figura 2 – Tipos de mensagens e preocupações de segurança relacionadas

Resumindo, deve-se assegurar que (ver Figura 2):

1. Um agente que solicita um serviço está autorizado para o fazer.
2. Um agente que disponibiliza um serviço está acreditado para o fazer.

3. Um agente que requer um resultado está autorizado para o obter.
4. Um agente que produz um resultado está acreditado para o oferecer.
5. Um agente que participa no *workflow* está autorizado para participar nesse mesmo *workflow*.
6. Um agente que disponibiliza um resultado é o mesmo agente que o produziu.

4.2. Estruturas de Suporte

A arquitetura deve conter estruturas de dados adequadas para suportar o modelo de segurança. As estruturas de dados seguintes são utilizadas para o efeito: *Certificate* e *Service Description*.

A estrutura **Certificate** baseia-se no RFC 5280. É utilizada com dois propósitos: identificar e autenticar agentes e determinar o nível de autorização para aceder aos serviços disponibilizados. O último é realizado através da utilização das extensões da versão 3 dos certificados X.509.

A estrutura dos certificados (ver Figura 3) contém informação sobre a entidade que certifica, sobre o próprio certificado e sobre todas as classificações de segurança, dado o agente e o tipo de certificado (autorização ou acreditação de serviço). Quando a acreditação do serviço está em causa, o certificado também contém a identificação do serviço.

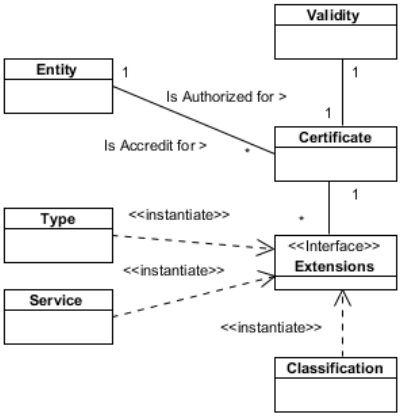

Figura 3 – Estrutura de dados do certificado

A classificação do serviço tem dois significados diferentes, dependendo do tipo de certificado. Nos certificados de autorização, que são utilizados para verificar a classificação de segurança possuída pelos agentes quando atuam como clientes, a classificação deve ser lida como o máximo de autorização que o agente detém. Nos certificados de acreditação de serviços, que são utilizados para acreditar pares {agente, serviço}, define o mínimo de classificação requerido para que um agente possa aceder ao serviço.

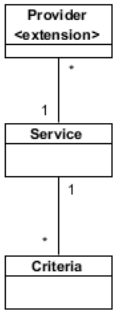

Figura 4 – Estrutura de dados da descrição do serviço

A estrutura **Service Description** baseia-se na especificação WSDL. O seu objetivo principal é disseminar informação sobre os serviços disponíveis na arquitetura. Inclui a descrição do serviço (ver Figura 4), informação sobre quem fornece o serviço (nomeadamente a localização e os seus certificados) e sobre um conjunto de características que são utilizadas pelos agentes para comparar os diferentes fornecedores do mesmo serviço (e.g.: custo; tempo de execução; necessidade de intervenção humana).

Na próxima subsecção expomos de que forma a combinação destas duas estruturas permitem abordar as questões de segurança identificadas.

4.3. A Mecânica do Modelo

Para abordar as questões de segurança identificadas na subsecção 4.1 os agentes utilizam o seguinte processo.

Após a receção de um *Service Request*, o agente que disponibiliza o serviço verifica se o agente que solicitou o serviço tem autorização suficiente para aceder ao serviço. Esta verificação é realizada através da comparação das classificações registadas no certificado de autorização do agente que solicita o serviço, que está disponível na mensagem recebida, com a classificação que é necessária para aceder ao serviço, que está disponível no certificado de acreditação do agente que fornece o serviço. Se o agente que solicita o serviço está autorizado a realizar o pedido, então o agente que fornece o serviço deve ser capaz de verificar se todos os agentes que participaram na prestação do serviço até àquele momento estão autorizados para o fazer de acordo com as políticas do agente que fornece o serviço. Isto é importante uma vez que podem existir políticas de autorização que impeçam que outros agentes participem no *workflow* (e.g.: conflitos de interesse; autorização baseadas no pedido original). Esta verificação também é realizada através da utilização da informação contida na mensagem do pedido do serviço, particularmente nos certificados de autorização e nos seus caminhos de certificação. Com este passo, os itens 1 e 5 estão verificados.

A receção do *Result Request* inicia ações que são similares àquelas que são iniciadas pela receção de um *Service Request*. Neste caso o agente deve verificar a autorização do

agente que solicita o resultado para aceder ao resultado pedido. Isto é feito através da utilização das classificações que estão presentes nos certificados de autorização do solicitador, permitindo que o fornecedor do resultado possa verificar se o agente solicitador está autorizado a aceder ao resultado e se pode participar no *workflow*, abordando os itens 3 e 5.

Para solicitar um serviço um agente deve escolher outro agente que possa fornecer os resultados necessários. Depois de o escolher, determina se este está acreditado para realizar o serviço, baseando-se, para isso, nos certificados e nos respetivos caminhos de certificação associados ao serviço, o que aborda os itens 2 e 5.

O último passo está relacionado com a solicitação de um resultado. Neste ponto, o agente tem um apontador em formato URI que indica a localização do resultado. Para além deste facto, tem também informação sobre o agente que produziu o resultado, nomeadamente os seus certificados. Com esta informação, o agente é capaz de verificar se o agente que fornece o resultado é o mesmo que o produziu, através da verificação da assinatura digital no URI, e de verificar se este está acreditado para produzir aquele resultado. Este passo aborda os itens 4 e 6.

5. Discussão

Como observámos na subsecção 4.1, a possibilidade de um agente delegar partes do *workflow* a outros agentes levanta um conjunto de preocupações de segurança. A nossa abordagem a estas preocupações baseia-se num conjunto de procedimentos, numa PKI e num conjunto de estruturas de dados de suporte.

A arquitetura de interoperabilidade que foi brevemente descrita na secção 3 levanta algumas preocupações de segurança, como mencionado previamente. Devido ao dinamismo da arquitetura, à capacidade que qualquer agente tem de realizar vários papéis simultaneamente e à gestão descentralizada do *workflow*, todos os modelos de segurança existentes se revelam inadequados. Para além disto, o modelo deve ter em consideração o facto de que a verificação da autorização deve ser realizada em diferentes contextos: *Service Request*; *Service Deliver*; e, *Result Request*.

O nosso modelo utiliza uma PKI de forma providenciar autenticação de agentes e acreditação de serviços e autorização.

Em termos de verificação de autorização, o objetivo do nosso modelo de segurança não é lidar com restrições de acesso específicas (que, argumentamos, são deveres do SIL) mas definir restrições de acesso gerais ao SIL. Com isto em mente, a nossa abordagem apenas requer a utilização das extensões de certificados apresentadas anteriormente. Mais, estas extensões e o conjunto de procedimentos que foram definidos permitem abordar todas as preocupações identificadas ao nível da autorização.

Mais, os modelos de segurança previamente referenciados definem uma função específica para cada agente, o que não se verifica na nossa arquitetura. Um agente pode prestar ou solicitar diversos serviços dentro do *workflow*, podendo, portanto, atuar com diferentes funções.

Então, para utilizar o RBAC seria necessário um grande número de funções e o necessário dinamismo para suportar a sua criação, manutenção e remoção em qualquer momento.

Finalmente, ao contrário de todos os outros trabalhos referenciados, a nossa proposta também aumenta a confidencialidade dos dados e a privacidade através da execução de *workflows*, impondo a solicitação obrigatória de resultados (que são mantidos pelos agentes que os produziram até serem explicitamente solicitados pelos seus destinatários finais).

6. Conclusões e trabalho futuro

Neste artigo apresentamos um modelo de segurança baseado em PKI para adicionar segurança a uma arquitetura de interoperabilidade baseada em agentes. Esta arquitetura foi igualmente sucintamente descrita, para apresentar as questões de segurança que levanta e que se devem essencialmente ao seu inerente dinamismo.

O modelo permite que agentes verifiquem as permissões de outros agentes para participar no *workflow* associado à prestação de um serviço, para verificar a autorização de outros agentes, para prestar e solicitar serviços e resultados e para verificar se os agentes que entregam resultados são os mesmos que originalmente os produzem. Mais, assegura a privacidade do conteúdo através da garantia de que os agentes solicitam explicitamente os resultados que lhes são destinados. A definição de políticas que possam lidar com restrições de acesso específicas ao SIL está fora do âmbito deste modelo.

No futuro é nossa intenção validar o modelo proposto através da exploração de um protótipo entretanto desenvolvido recorrendo a casos de utilização, os quais irão envolver o desenvolvimento de diversos tipos de agentes.

Referências

Atluri, V., & Huang, W.-kuang. (1996). An authorization model for workflows. Computer Security—ESORICS 96 (pp. 44–64). Springer.

Chou, S. C., & Wu, C. J. (2004). An Access Control Model for Workflows Offering Dynamic Features and Interoperability Ability. Int. Computer Symposium, Dec (pp. 15–17).

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). Web Services Description Language (WSDL) 1.1.

Clement, L., Hately, A., Riegen, C. von, & Rogers, T. (Eds.). (2004). UDDI Version 3.0.2 Specification. Retrieved from uddi.org/pubs/uddi_v3.htm

Hung, P. C. K., & Karlapalem, K. (2003). A secure workflow model. Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21 (Vol. 21, pp. 33–41). Australian Computer Society, Inc.

Kannammal, A., & Iyengar, N. C. S. N. (2008). A Framework for Mobile Agent Security in Distributed Agent Based E-Business Systems. International Journal of Business and Information, 3(1), 129-143.

Marques, F., Dias, G. P., & Zúquete, A. (2011). A General Interoperability Architecture for e-Government based on Agents and Web Services. 6a Conferência Ibérica de Sistemas e Tecnologias de Informação (pp. 338-343).

OECD. (2001). E-Government: Analysis Framework and Methodology. Group, (November), 1-10.

Savarimuthu, B. T. R., Purvis, M., & De Oliveira, M. (2004). Towards Secure Interactions In Agent Societies. Citeseer, 143-148.

Stormer, H., Knorr, K., & Eloff, J. (2000). A model for security in agent-based workflows. Informatik Informatique, 6, 24–29.

Thomas, R., Sandhu, R. (1997). Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management. Proceedings of the IFIP WG11.3 Workshop on Database Security, Lake Tahoe, California.

Wainer, J., Barthelmess, P., & Kumar, A. (2003). W-RBAC-a workflow security model incorporating controlled overriding of constraints. International Journal of Cooperative Information Systems, 12(4), 455–485. Citeseer.