Computação em Grelha: Perspectivas de Segurança

Pedro Gama, Paulo Ferreira
INESC-ID/IST
Distributed Systems Group
Rua Alves Redol, nº9, 1000-029 Lisboa
[pedro.gama, paulo.ferreira]@inesc-id.pt

Abstract

A utilização de sistemas de computação em grelha permitiu revolucionar a partilha e optimização de recursos heterogéneos entre entidades distintas.

Em paralelo, estes sistemas introduziram também inúmeras vulnerabilidades de segurança, associadas aos sistemas distribuídos no geral, e aos sistemas em grelha em particular, a que é necessário fazer face.

Apresentamos neste artigo uma análise dos aspectos de segurança que consideramos mais relevantes para sistemas de computação em grelha, como a autenticação dos utilizadores e recursos, a autorização nos acessos, a delegação de privilégios e a comunicação entre nós. Complementamos ainda essa análise com a avaliação das arquitecturas de segurança das plataformas existentes, quando aplicável.

Pretendemos contribuir para a consciencialização da importância da segurança nas plataformas de computação em grelha, contribuindo para a evolução deste tipo de sistemas.

1. Introdução

Os sistemas distribuídos tradicionais, como o AFS (*Andrew File System*)[51], o Sun NFS (*Network File System*) [81] e o Coda[83] são utilizados, entre outros aspectos para interligar recursos fisicamente separados, permitindo fazer face a inúmeras situações em que as aplicações necessitam de aceder a recursos remotos de forma segura. Os conceitos utilizados foram amadurecidos durante largas décadas, desde os artigos embrionários de Licklider[60] e Vyssotsky[102].

No entanto, a necessidade de optimizar os recursos existentes nas organizações obriga à integração de diversos recursos heterogéneos e dinâmicos, envolvendo em simultâneo vários sistemas operativos e respectivas arquitecturas. Visto que estes sistemas podem pertencer a organizações distintas, surgem dificuldades adicionais na disponibilização segura dos recursos. Os recursos e os próprios utilizadores necessitam de ser autenticados e autorizados de forma segura entre as diversas entidades, que podem utilizar sistemas heterogéneos.

Surgem assim os conceitos estreitamente relacionados de computação em grelha (ou *grid*) e organização virtual (*Virtual Organization* ou VO). No âmbito deste artigo utilizaremos com o mesmo significado as expressões "*grid*"e "organização virtual".

A noção de computação em grelha é derivada da analogia com as grelhas de fornecimento de energia eléctrica. Nestes sistemas um utilizador pode decidir em qualquer momento utilizar o serviço (recurso) disponibilizado através de uma interface normalizada[111]. Não necessita de possuir qualquer infraestrutura e/ou plataformas para gestão do serviço (como um servidor de armazenamento por exemplo), bastando-lhe utilizar a interface apresentada. Numa grelha de computação, em particular, os serviços disponibilizados podem ser de múltiplos tipos, não se circunscrevendo à capacidade computacional (CPU) como é vulgar encontrar nos casos de estudo apresentados na literatura [49, 94]. A grelha pode alternativamente disponibilizar capacidade de armazenamento, largura de banda, licenças de software e equipamento especializado, entre outros recursos. [11, 72].

O conceito de organização virtual foi introduzido por Ian Foster[33], no que é considerado um artigo embrionário sobre esta temática[32]. Assim, uma organização virtual é apresentada como um conjunto de entidades distintas que partilham dinâmicamente os seus recursos heterogéneos[37, 11, 41]. Desta forma podem optimizar a utilização dos mesmos recursos, fazendo face a diversos tipos de problemas de larga escala. Como exemplo podemos referir o projecto *Crossgrid*[30], associado ao *Large Hadron Collider* (LHC) do CERN. Este projecto pretende possibilitar em primeiro lugar a partilha de capacidade de armazenamento entre todos os participantes, de forma a possibilitar o armazenamento de cerca de dois Petabytes por experiência, volume de dados dificilmente comportável numa

arquitectura centralizada. Por outro lado pretende-se ainda utilizar os recursos partilhados para facilitar a dispersão (recorrendo a recursos de rede) e processamento (recorrendo a recursos de CPU) de toda a informação obtida.

Os problemas centrais neste tipo de plataformas prendem-se com a necessidade de acomodar um nível elevado de dinamismo e heterogeneidade no que respeita às políticas e mecanismos de segurança. Por um lado, qualquer entidade pode introduzir em qualquer momento novos utilizadores e recursos na organização, devendo estes ser integrados de forma rápida e coerente na rede (com possibilidade de respectivamente aceder aos recursos existentes e por outro lado serem acedidos pelos utilizadores existentes). Por outro lado os diversos mecanismos e políticas de segurança existentes nas diversas entidades devem interoperar de forma consistente, garantindo que o nível de segurança de qualquer entidade não é reduzido pelo facto de estar integrada na organização virtual. Por último deve ser possível estabelecer políticas de segurança eficazes a todos os níveis da organização virtual (desde a organização virtual como um todo até ao gestor de segurança de cada recurso, passando por eventuais *schedulers*, *brokers*, etc).

Existem inúmeras soluções para a disponibilização de recursos em sistemas distribuídos de grande dimensão, associadas a diferentes plataformas com aproximações e mesmo semânticas de funcionamento distintas[7]. Por exemplo a plataforma *peer-to-peer JXTA*[91, 90] e o sistema *European Data grid* (EDG)[28] podem ambos ser encarados como plataformas de grelha. No entanto, o primeiro possibilita a formação de redes ad-hoc com um sistema de segurança bastante informal (baseado por exemplo em auto-assinaturas e redes de confiança), enquanto o segundo é baseado em infraestruturas formais de certificação entre as diversas entidades e processos formais de adesão à organização virtual. Claramente os requisitos de segurança destas plataformas serão diferentes.

Por outro lado, mesmo ao nível de plataformas que explorem a mesma semântica em termos de utilização é possível encontrarmos inúmeros mecanismos distintos com o mesmo objectivo. Enquanto que o sistema *Legion*[109, 47] utiliza as potencialidades de segurança base do *Unix* para proteger a execução de aplicações de *grid* associando-as a contas locais, o sistema Crisis[10] utiliza o sistema de execução remota segura *Janus* para o mesmo efeito.

Organizações como o GGF (*Global grid Forum*)[18] promovem a unificação e interoperabilidade de todos estes mecanismos, apostando fortemente na integração das plataformas de *grid* com os mecanismos de webservices [24, 86, 67], através da adopção da arquitectura OGSA[34]. Efectivamente, os grupos de trabalho de web services estão a trabalhar activamente na definição de arquitecturas de segurança [68, 26, 58] que podem ser utilizadas de uma forma interoperável em diversos aspectos da operação de uma *grid*, como sejam os serviços de informação e submissão de processos. No contexto deste artigo focamos as acções do grupo de trabalho OGSA-SEC-WG (*Open grid Service Architecture Security Working Group*[1]), que constitui um dos grupos de trabalho do GGF. A arquitectura GSI (*Grid Security Infrastructure*) é introduzida no âmbito da OGSA como um modelo de autenticação e autorização para ambientes de *grid*[106], incorporando aspectos como *single sign-on*, delegação e mapeamento de credenciais, entre outros.

Este artigo apresenta uma análise da problemática da segurança nas principais plataformas de computação em grelha existentes actualmente. Pretendemos contribuir para a evolução da segurança associada a este tipo de sistemas através de uma compilação e análise das alternativas existentes para as diversas funcionalidades a incorporar numa arquitectura de segurança.

Este artigo está organizado da seguinte forma: Na próxima secção apresentamos um *overview* geral sobre o universo da computação em grelha. De seguida, a secção 2 abarca diversos aspectos relacionados com a identificação das entidades e sua autenticação num ambiente de *grid*. A secção 3 apresenta as possibilidades de realizar a autorização do utilizador, após este ter sido autenticado. A temática da comunicação entre nós do *grid* é abordado na secção 4 e o isolamento na execução de aplicações remotas na secção 5. Finalmente apresentamos os aspectos relacionados com a delegação e revogação de privilégios nas secções 6 e 7, seguido da seccão 8 relativa à definição e aplicação de políticas de segurança em ambientes de computação em grelha e das conclusões do artigo.

2 Autenticação

A identificação e posterior autenticação dos utilizadores constitui um dois principais desafios em sistemas que envolvam diversas plataformas independentes, como é o caso do *grid* [11]. Devido ao facto da autenticação ser usualmente realizada a nível global e permitir funcionalidades como o *single sign-on*, a sua falha pode comprometer todo o restante sistema.

Por um lado, a flexibilidade na incorporação de novos utilizadores e recursos no âmbito de uma organização virtual dificulta a atribuição de identificadores unívocos[95]. Por outro lado os sistemas constituintes do *grid* pretendem continuar a aplicar os seus próprios mecanismos de segurança em inúmeras situações, necessitando para isso de autenticar o utilizador localmente (por exemplo para aplicação de controle de acesso base nos sistemas *Unix*).

Efectivamente, em sistemas distribuídos de reduzida complexidade, cada utilizador pode ser associado de uma forma estática a uma conta distinta, que utilizará em conjunto com uma password para se identificar perante o sis-

tema [10]. No entanto, não é viável atribuir a cada potencial utilizador de uma organização virtual uma conta local a cada sistema, não só pela enorme carga administrativa que essa tarefa exigiria como também pela redução substancial de flexibilidade no mecanismo de autorização [71].

2.1 Identificação

O facto de novos recursos e utilizadores poderem registar-se dinâmicamente nas organizações virtuais torna impossível para um determinado sistema prever à partida todos os potenciais utilizadores dos serviços disponibilizados. Como tal é necessário introduzir mecanismos de confiança entre as diversas entidades ou entre uma entidade e a superestrutura da organização virtual, no sentido de garantir que um determinado utilizador possui os privilégios necessários para aceder a um recurso.

Como podemos concluir a partir da Tabela 1, a maior parte das plataformas de *grid* analisadas utilizam tecnologia de chave pública[84] (*Public Key Infrastructure* ou PKI) no sentido de identificar univocamente os utilizadores. Os sistemas PKI promovem a interoperabilidade entre diversos sistemas através da utilização de um formato normalizado para descrever a informação relativa a um determinado utilizador. Esta é armazenada num certificado digital, cuja norma mais conhecida é a X.509[61] . A cada utilizador é atribuído um par de chaves digitais, composto por uma chave pública e uma chave privada. A chave pública é inserida, juntamente com as restantes informações do utilizador, no certificado digital, que é de seguida assinado pela respectiva entidade de certificação (*Certification Authority* ou CA). A chave privada associada é utilizada para provar a posse do certificado digital. De forma a aumentar o nível de segurança deste sistema a chave privada é usualmente cifrada com uma senha de acesso, ou armazenada num dispositivo resistente à intrusão, como um cartão inteligente (*smartcard*). Para mais pormenores relativos a certificados digitais e criptografia de chave pública o leitor é direccionado para [84].

O sistema *Globus*, sobre o qual assentam inúmeras plataformas de *grid* utiliza certificados X.509 de modo a identificar univocamente cada entidade (seja esta um utilizador, um servidor (ou *container* ou uma aplicação) [35, 16]. De forma a suportar um nível acrescido de segurança na gestão da chave privada de um utilizador, o Globus suporta ainda opcionalmente o armazenamento da informação dentro de um cartão inteligente (*smartcard*). Durante a operação de login o cartão é contactado no sentido de criar um proxy certificate utilizando o protocolo PKCS11[2].

O sistema *Legion*[88, 70] atribui um *Legion Object Identifier (LOID)* unívoco a cada um dos utilizadores do sistema. Este LOID contém diversos campos de informação associados ao utilizador, incluindo ainda um certificado x.509 com uma chave pública RSA. Após autenticar um utilizador, o sistema *Legion* gera uma credencial de curta-duração, que serve para identificar univocamente o utilizador no sistema. Esta credencial é enviada com cada pedido de serviço, devidamente assinada pela chave pública do destinatário.

O sistema Unicore[29] também utiliza PKI e [46] descreve a infraestrutura em maior detalhe, focando o impacto de ter uma única ou múltiplas autoridades de certificação.

O sistema CRISIS atribui a cada um dos utilizadores um certificado digital X.509 com uma assinatura de longa duração. No entanto, para que seja aceite por um interlocutor, este certificado deve ainda ser co-assinado por um Online Agent (OLA), que verifica a validade da assinatura realizada. A assinatura do *Online Agent* é normalmente de curta duração, destinando-se à execução de uma operação. Caso um *Online Agent* seja comprometido e realize um ataque de negação de serviço através da recusa em validar certificados digitais, o utilizador poderá simplesmente utilizar outro OLA. A revogação de certificados digitais comprometidos pode ser realizada facilmente através da comunicação das CRL's aos OLA's, não sendo necessária a distribuição por todos os utilizadores.

A orientação a serviços das novas normas de *grid*, nomeadamente do OGSA, leva a que os próprios mecanismos de autorização presentes numa plataforma de *grid* comecem a ser baseados em serviços [86]. Surge assim o conceito de VOMS (*Virtual Organization Membership Service*), que centraliza todos os mecanismos de autenticação presentes numa organização virtual, permitindo aos diversos sistemas delegarem essa responsabilidade [75].

Por último, poderão ainda existir sistemas em que se pretenda proporcionar uma utilização anónima dos recursos. Neste caso as credenciais de acessos poderão conter não a identidade do utilizador, mas somente um papel (role) gerado dinâmicamente e que permita a identificação dos privilégios a utilizar para a realização da operação pretendida. De notar que uma utilização anónima de um recurso não significa obrigatoriamente que o utilizador não se registe. Este poderá registar-se perante um sistema de autenticação da organização virtual que lhe fornece uma credencial com os seus privilégios. A partir daí o acesso aos recursos é absolutamente anónimo. De momento, e que seja do nosso conhecimento, nenhum sistema de *grid* existente suporta este tipo de funcionalidades.

2.2 Certificação de Identidades

De forma a utilizar um recurso de uma determinada organização virtual, o utilizador apresenta o seu identificador (habitualmente a sua chave pública X.509) de forma a validar os seus privilégios. O gestor de segurança do recurso necessita então de verificar a autenticidade da assinatura no certificado.

Tabela 1. Mecanismos de Identificação

Globus	Certificados x509
Consh	Mecanismos Locais
Gridbank	Certificados x509
EDG	Certificados x509
Nordugrid	Certificados x509
JXTA	Certificados x509
Crisis	Certificados x509
Condor	Certificados x509
NASA IPG	Certificados x509
Legion	Certificados x509
Alchemi	*Username* / *password*
Unicore	Certificados x509
IGENV	Certificados x509

No entanto, visto que uma organização virtual possuirá usualmente uma ou várias autoridades de certificação é necessário garantir que o sistema onde o recurso está localizado reconheça a autoridade de certificação que gerou o certificado digital em causa. Existem diversas aproximações para este objectivo: a existência de uma única entidade de certificação a nível global, a definição de uma hierarquia de entidades de certificação e a federação de entidades de certificação.

A primeira solução está ser utilizada a nível de projectos de pequena e média dimensão, em situações onde é possível formalizar a nível institucional a confiança numa única entidade de certificação. No entanto, em projectos envolvendo a criação de redes *ad-hoc* de recursos e com um alcance a nível planetário, esta aproximação tem uma viabilidade reduzida. Actualmente não se antevê um acordo sobre uma entidade de certificação a nível global, apesar dos esforços de algumas entidades como a Microsoft. O sistema Passport [74] obteve uma adesão razoável graças à inclusão automática de todos os utilizadores do conhecido sistema de correio electrónico Hotmail, mas esta infraestrutura não está actualmente a ser utilizada em nenhum sistema de grelha. Adicionalmente esta aproximação levanta numerosas críticas devido ao facto de uma única entidade reunir informações pessoais sobre um número tão elevado de utilizadores.

Uma solução alternativa é a de constituir uma hierarquia de entidades de certificação que possibilite a um sistema verificar a identidade de um certificado através da pesquisa da sua cadeia de certificação (*certification chain*). No entanto visto que a hierarquia de certificação não tem uma raíz única, existe a possibilidade de um certificado não poder ser verificado devido a não existir nenhuma CA na cadeia de certificação respectiva que seja da confiança do sistema.

Finalmente, a federação de entidades de certificação, composta por diversas hierarquias de entidades de certificação que se autorizam mutuamente é o mecanismo promovido por plataformas como o Liberty Alliance e o WS-Federation [23]. Será razoável assumir que, dentro de uma organização virtual, irão coexistir diversas entidades de certificação, pelo que terão que existir mecanismos pelos quais se possam inter-certificar. Desta forma todos os utilizadores poderão ser identificados de forma segura[46], como no sistema MDS-2[25] incorporado na arquitectura da plataforma *Globus*. Esta alternativa parece efectivamente ser a que melhor se adequa ao ambiente de *grid* [21]. No entanto, a heterogeneidade dos sistemas e a dinâmica na partilha e utilização dos recursos impede a utilização de alternativas que obriguem a um peso administrativo excessivo. Efectivamente a morosidade de inter-certificação institucional impede inúmeras vezes uma eficaz utilização dos recursos.

Um aspecto transversal à certificação de identidades prende-se com o nível de formalização da autoridade de certificação. O facto de se utilizarem certificados digitais não significa obrigatoriamente que se tenha que instalar uma Autoridade de Certificação central. Por exemplo, em comunidades *peer-to-peer* poder-se-ão constituir grupos ad-hoc de utilizadores que designarão uma pseudo-autoridade de certificação, que assina os certificados digitais dos elementos da comunidades. Poderão até existir sistemas em que os utilizadores simplesmente auto-assinam os seus certificados, que ficam assim associados a um nível mínimo de segurança. Por exemplo o sistema Poblano [19], desenvolvido para a plataforma JXTA introduz a noção de espectro de confiança, abrangendo desde certificados assinados por uma CA de confiança, até certificados auto-assinados no outro extremo.

Os certificados podem ser co-assinados pelos contactos do utilizador, criando-se assim uma rede de confinaça similar ao conceito de web of trust existente na plataforma PGP (Pretty Good Privacy) [112]. Esta aproximação baseia-se na existência de relações de proximidade entre utilizadores. Complementarmente podem ser designados diversos co-assinantes dentro de uma comunidade virtual, que passam assim a actuar como pseudo autoridades de certificação.

2.3 Autenticação Global

Devidamente munido da sua identificação, um utilizador começa por se autenticar globalmente na grelha, de forma a beneficiar de mecanismos como o *single sign-on* (que lhe permite utilizar diversos serviços sem ter que se voltar a autenticar).

De forma a autenticar-se, um utilizador apresenta habitualmente o seu certificado digital ao mecanismo de autenticação da organização virtual. A VO utiliza um mecanismo (como o disponibilizado pelo TLS[38]), em que através de uma protocolo *challenge-response* valida a identidade do utilizador. Emite então, dependendo do sistema

específico, uma credencial [63, 88] ou um certificado de representante (proxy certificate)[14] contendo a identidade e/ou os privilégios do utilizador. Este novo certificado é assinado pelo certificado digital do utilizador (utilizando a sua chave), ao invés de pela Autoridade de Certificação.

Apesar deste processo reduzir de forma significativa o número de vezes que um utilizador necessita de aceder à sua chave privada, cria outras vulnerabilidades relacionadas com o facto das credenciais e certificados de representante poderem ser comprometidos. No entanto, visto que a sua validade é normalmente reduzida (na ordem das horas), assume-se este risco como razoável em função dos benefícios alcançados[15].

Outro ponto relevante em termos de autenticação é o que se refere à possibilidade dos interlocutores se autenticarem mutuamente. Apesar de usualmente apenas se considerar a autenticação do utilizador pelo sistema que lhe está a disponibilizar o acesso a um recurso, existem inúmeras situações em que se torna necessário validar previamente a identidade de um recurso antes de lhe fornecer informação de acesso potencialmente sensível. Uma área onde esta preocupação se torna especialmente relevante é a área médica, em que o simples pedido de informação relativamente a determinado assunto poderá interferir na privacidade de um utilizador ao revelar que ele padece de determinada doença.

A autenticação mútua é obtida na plataforma *Legion* através da utilização dos certificados digitais de ambas as partes. O utilizador que pretende aceder a um serviço envia para este uma mensagem contendo credenciais de autorização, devidamente cifradas com a chave pública do destinatário. Visto que só este conhece a chave privada correspondente, necessária para decifrar as credenciais, o utilizador garante que só o recurso poderá aceder a informação potencialmente sensível presente na credencial. A autenticação do utilizador pelo recurso é conseguida pela verificação das credenciais transmitidas por este e da assinatura presentes na mesma, as quais garantem a identidade e os privilégios do mesmo.

Outro aspecto importante a considerar prende-se com a possibilidade de partilhar informação de contexto entre organizações virtuais distintas. Caso um utilizador ou um recurso estejam envolvidos em organizações virtuais distintas, é desejável eliminar a necessidade de replicar as credenciais do utilizador e as políticas de gestão de recursos por todas as organizações virtuais, sob risco de cometer falhas que poderão afectar a integridade do sistema. Caso seja realizada automaticamente, esta operação permitirá não só minimizar o custo administrativo associado, como ainda compatibilizar ou alertar acerca de políticas de segurança incompatíveis entre si, ou com a política global da organização virtual. Neste sentido é necessário encontrar formas de partilhar de uma forma segura a informação de contexto dos utilizadores e dos recursos (chaves privadas, certificados digitais, credenciais, políticas, etc). As infraestruturas de segurança existentes actualmente não possuem, que seja do nosso conhecimento, este tipo de funcionalidades, apesar de diversos grupos como o *IETF's SACRED Working Group* estarem a trabalhar nesse sentido[5].

2.4 Autenticação Local

Após realizar uma autenticação global no sistema de *grid*, um utilizador poderá, caso disponha de privilégios suficientes, utilizar os recursos disponíveis. No entanto, um dos princípios base das plataformas de *grid* é permitir aos recursos locais um grau de autonomia considerável de forma a garantir que eles possam aplicar as suas próprias políticas de segurança [69, 50]. Nagaratnam [67] apresenta mesmo como um dos principais desafios dos sistemas de *grid* a integração/interoperabilidade com os mecanismos de segurança existentes localmente.

Alguns sistemas possuem mecanismos rudimentares de mapeamento do identificador global em contas locais. Este é o caso dos sistemas *Globus* e *Legion*, onde ficheiros designados respectivamente por */etc/grid-mapfile* e */etc/-LegionUsers* contém as associações entre as entidades da organização virtual e os utilizadores do sistema local [106, 62, 100]. Apesar de muito simples, esta aproximação peca por obrigar a uma contínua actualização, pelos administradores de sistema, dos potenciais utilizadores dos recursos. No sentido de reduzir este esforço a aproximação adoptada pelos administradores de alguns destes sistemas resume-se a criar uma única conta na máquina para utilizadores do *grid* (e.g. *grid*-user) mapeando todos os utilizadores autenticados globalmente nesta conta. Não é assim possível especificar permissões específicas para cada um dos utilizadores globais.

Outra alternativa, similar à utilizada no *Kerberos*, é a de realizar a autenticação do utilizador num mecanismo global à *grid*, que gera credenciais permitindo ao utilizador comprovar a sua identidade e grupos/papéis em que se posiciona [75]. Esta alternativa permite complementar os mecanismos base do *Globus* e do *Legion*, visto que um mecanismo de autorização como o CAS (ver secção 3) permite delegar as decisões de autorização no acesso aos recursos.

Finalmente, alguns sistemas desenvolveram à medida módulos de interoperação com alguns dos mecanismos de autenticação mais usuais do mercado [66]. Os sistemas *Globus*, *Legion* e JXTA podem assim ser associados a um servidor kerberos no sentido de interoperarem com sistemas locais que utilizem estes mecanismos de segurança [52]. O *Globus*, em particular, possui um centro de distribuição de chaves (*Key Distribution Center* ou KDC) Kerberos modificado, designado por SSLK5D, que lhe permite realizar a interface entre os mecanismos de autenticação do *Globus* e os sistemas de segurança existentes numa organização.

Este KDC verifica num repositório local qual a associação entre a entidade presente no certificado e uma entidade presente no servidor Kerberos, de forma a gerar um bilhete (ticket), que pode ser utilizado como qualquer outro bilhete do Kerberos[66].

A plataforma JXTA contempla ainda em termos de autenticação a utilização de um modelo de confiança ad-hoc como o Poblano [19]. Este permite criar um modelo de confiança distribuído que não depende da actuação de uma entidade central. Funciona de um modo similar ao sistema *Pretty Good Privacy*[112], sendo da responsabilidade de cada utilizador atribuir um determinado grau de confiança aos utilizadores que conhece. A partir do cruzamento da informação dos diversos utilizadores, o Poblano avalia o risco associado à utilização de determinado recurso.

3 Autorização

A autorização no acesso a recursos é um aspecto crucial no âmbito dos sistemas de informação.

No caso particular dos sistemas de computação em grelha, a decisão relativa ao acesso a um recurso é normalmente partilhada por um conjunto diverso de entidades, o que dificulta o seu processamento[53]. Se por um lado o fornecedor de cada recurso quer garantir que as regras de utilização são respeitadas, a organização virtual no seu todo quer também garantir uma eficaz utilização dos recursos. Visto que as diferentes políticas se encontram em sistemas distintos e muitas vezes heterogéneos, a coordenação e aplicação das mesmas não é de forma alguma um aspecto simples.

Adicionalmente, a especificação dos privilégios de acesso a um recurso obriga normalmente à utilização de políticas mais complexas do que a utilização de simples ACLs. Note-se, por exemplo, que inúmeros sistemas de grelha disponibilizam serviços de submissão de pedidos. Estes serviços, apesar de não acederem aos recursos propriamente ditos, necessitam de permissões para obter informação de contexto dos mesmos, como sejam os níveis de utilização actual e máximo, de forma a conseguirem optimizar a utilização da organização virtual. Adicionalmente, após o início da execução de uma tarefa, um administrador da grelha deve ter a possibilidade de desactivar um recurso com comportamento erróneo ou maligno. Isto obriga a que os administradores globais da grelha detenham privilégios sobre cada um dos recursos disponibilizados, o que nem sempre é possível, visto que os gestores dos recursos se encontram distribuídos pelos diversos sistemas.

3.1 Políticas de Segurança Locais

A maior parte dos sistemas de *grid* existentes actualmente, dos quais se destacam o *Globus* e o *Legion*, centram as suas funcionalidades na correcta autenticação do utilizador, relegando a aplicação de uma política de segurança para os recursos locais[78, 92]. Németh[72] formaliza os mecanismos de autorização presentes numa organização virtual, introduzindo a noção de um handler local a cada sistema e que controla o acesso a recursos do *grid*.

O GSI (componente de segurança do *Globus*) realiza a autenticação do utilizador atribuindo-lhe um certificado de representante [14]. Este certificado é então transmitido para o recurso a utilizar, de forma a identificar univocamente o utilizador. A identificação global do utilizador é mapeada, através da utilização do ficheiro *grid*-map numa identificação local, sobre a qual são realizadas as verificações de segurança [37].

Não existem assim limites para os mecanismos de segurança locais a implementar após a obtenção de uma identificação local. Na maior parte das situações, a impossibilidade de registar localmente todos os utilizadores da organização virtual (devido ao seu elevado número), leva a que a maioria dos sistemas opte por mapear todos os utilizadores de uma organização virtual numa única conta local, que é de seguida utilizada para realizar as verificações de segurança [92]. Este método limita de forma substancial a eficácia dos mecanismos de autorização, pois ou se restringe fortemente as permissões de todos os utilizadores associados ao sistema ou se corre um risco acrescido de má utilização dos recursos. Em particular refere-se em [54] a necessidade de associar aos mecanismos de autorização do Globus um mecanismo que suporte *use-conditions and attributes*.

O gestor de segurança local pode realizar diversas verificações ao nível da cadeia de certificados presente no pedido de acesso ao recurso. Assim, pode ser verificada não só a identidade de um utilizador, como também a autoridade de certificação que emitiu o certificado. Isto permite a aplicação de políticas em que seja aceite o certificado do colaborador de uma empresa emitido pela autoridade de certificação da empresa, mas não seja aceite a mesma autoridade de certificação para emitir certificados para os clientes dessa empresa.

O sistema *Legion*, apoiado sobre uma arquitectura baseada em objectos, realiza um procedimento similar através da geração de uma credencial que é depois enviada para os recursos locais [88]. A nível local cada recurso (representado por um objecto) possui o seu gestor de segurança próprio, permitindo assim a aplicação de políticas específicas do recurso [52]. Estas políticas são na versão base do sistema baseadas em ACLs (lista de controlo de acessos). No entanto, o facto de o sistema *Legion* ser baseado em objectos permite também, de uma forma simples e intuitiva, estabelecer políticas ao nível de todo um sistema. Para isso basta a um administrador de sistema modificar a política de segurança de um dos objectos no topo da hierarquia (nor-

malmente o objecto Hosts), acrescentando-lhe a semântica pretendida, como por exemplo impedir o acesso a recursos locais por parte de utilizadores externos à instituição [31]. Da perspectiva do utilizador é possível especificar quais os sistemas confiáveis a utilizar.

O sistema CRISIS constitui a componente de segurança do sistema operativo distribuído WebOS [98]. O modelo de autorização no sistema CRISIS baseia-se na utilização híbrida de ACLs e capacidades (capabilities). Depois de analisar todos os certificados enviados no pedido de um determinado serviço, o sistema verifica quem é o utilizador original (o CRISIS permite delegação de credenciais) utilizando essa identificação para a verificação dos privilégios [10]. Actualmente as ACLs em CRISIS referem apenas quais os utilizadores que podem aceder aos recursos mas prevê-se que em futuras versões se possam também incluir semânticas de consumo máximo na política de utilização do recurso [10]. Um dos problemas do sistema CRISIS para utilização em *grid* é que não é possível realizar a interoperação com sistemas de segurança locais, sendo obrigatório utilizar a infraestrutura CRISIS[36]. Visto que estes mecanismos de autorização baseados em ACLs e capacidades apresentam falhas evidentes na adaptação a ambientes de *grid*, diversos sistemas estenderam a infrastrutura base com políticas de segurança mais avançadas, como o conceito de papéis (*roles*).

O sistema EDG (*European Data Grid*) [28] consiste exactamente num desses casos. Apesar da plataforma base sobre a qual assenta seja o *Globus*, o mecanismo de autorização foi extendido de forma a permitir realizar verificações de segurança globais previamente à invocação dos serviços/recursos (de notar que os próprios recurso realizam também as suas próprias verificações de segurança) [13].

Outro exemplo é o sistema Alchemi [63], que disponibiliza um serviço de autorização baseado em papéis de forma a controlar o acesso aos recursos.

3.2 Políticas Globais à Organização Virtual

Apesar de ser importante que cada um dos participantes da VO possa manter a sua autonomia e políticas de segurança próprias, existem diversas situações em que faz sentido a aplicação a nível global de políticas de segurança [57, 99]. Tal pode ser devido à existência de regras de utilização dos recursos da VO, aceites por todos os participantes, ou simplesmente porque alguns participantes preferem delegar a funcionalidade de autorização num mecanismo disponibilizado pela VO [76, 37, 86].

Um caso de realce é a plataforma *Gridbank*[8]. Este sistema está fortemente associado ao conceito de *grid* economy, pelo qual a disponibilização de recursos na *grid* (p.e. poder computacional) é realizada mediante pagamento de determinado valor monetário ou emissão de créditos para futura utilização de outros recursos. Tendo por base a plataforma *Globus*, o Gridbank funciona como um intermediário entre fornecedores de recursos e utilizadores, negociando os preços a praticar e salvaguardando os interesses de ambas as partes.

O sistema CAS (*Community Authorization Service*) [76, 92] permite a um determinado recurso delegar de forma eficiente a responsabilidade de gestão dos utilizadores e das permissões de acesso para uma autoridade central na organização virtual. O CAS está actualmente integrado com a plataforma *Globus* e, ao invés do GSI que gera certificados de representante com privilégios ilimitados, permite a geração de credenciais contendo somente os privilégios do utilizador ou dos grupos a que ele pertende [77]. Caso a informação de acesso de determinado utilizador seja comprometida basta retirar esse utilizador do sistema CAS, limitando o problema ao período temporal de validade das credenciais, que normalmente ronda um número reduzido de horas. Um administrador central da organização virtual, em conjunto com responsáveis de projecto por ele designados, controla centralmente os privilégios de acesso aos diversos recursos existentes. Este sistema constitui assim uma alternativa ao ficheiro grid-mapfile (ver secção 2) que diminui bastante a carga administrativa que nesta última situação teria que existir em cada um dos sistemas. Uma das desvantagens deste sistema é obrigar à alteração das aplicações que necessitam de validar as credenciais emitidas de forma a validar se a operação pretendida pelo utilizador é compatível com as credenciais que este apresenta. Gannon[43] refere um conceito similar através da descrição do *Grid Authorization Service*.

Ramakrishnan [77] apresenta uma plataforma de autorização baseada em componentes para sistemas de *grid*. A plataforma é baseada nas arquitecturas OGSA (*Open Grid Services Architecture*) e CCA (*Common Component Architecture*), permitindo a especificação de políticas de segurança diversas. No protótipo implementado as políticas são constituídas por simples listas de controle de acesso (ACLs), mas os autores referem a possibilidade de incorporação de políticas mais avançadas. Este sistema funciona de forma muito semelhante ao CAS, apresentado anteriormente, na medida em que pode ser utilizado como um serviço de segurança no âmbito de uma organização virtual. No entanto pode também ser utilizado como *middleware* ficando a constituir o módulo de segurança de uma determinada aplicação.

Wasson [103] foca o problema de aplicação coerente das políticas de segurança dentro de uma organização virtual, apresentando uma plataforma para aplicação de políticas de controle de utilização de recursos (*resource provisioning*).

No contexto de uma computação em *grid* a quantidade de dados transferidos entre nós do sistema poderá obrigar a

Tabela 2. Mecanismos de Autorização

Globus	Mecanismos Locais
Consh	*Janus*
Gridbank	Módulo Gridbank
EDG	*Globus* + Mecanismo proprietário
Nordugrid	*Globus* + Mecanismo proprietário
JXTA	Mecanismos Locais, Poblano
Crisis	Mecanismos Locais + *Global Security Manager*
NASA IPG	*Globus* + Motor de Políticas
Legion	Mecanismos Locais
Alchemi	*Role-based Access Control*
Unicore	BD Local
IGENV	Motor de Políticas

replicar e realizar cache dos dados, de forma a aumentar o desempenho [20]. Hoschek [50] refere os problemas relacionados com esta replicação e caching de dados em plataformas de *grid*, visto que o armazenamento de informação confidencial num servidor não seguro permitirá ultrapassar as políticas de segurança estabelecidas, comprometendo assim a segurança do sistema.

4 Comunicações

Num sistema distribuído tradicional o modelo de comunicações desempenha um papel crítico na arquitectura de segurança do sistema. De facto, a própria operacionalidade do sistema baseia-se muitas vezes na própria capacidade de partilhar informação entre os diversos nós. Nos sistemas de computação em grelha, este aspecto é particularmente relevante, pois a execução de uma única operação envolve geralmente a partilha de informação entre múltiplas entidades envolvidas na computação. Imagine-se, por exemplo, o caso de um utilizador que pretende realizar uma simulação oceanográfica. Este poderá contactar um gestor de recursos (*broker*), requisitando-lhe a operação pretendida. O *broker* entrará em contacto com um *scheduler* de forma a validar a disponibilidade dos recursos envolvidos, e o custo da operação. Após validação da configuração pelo utilizador, vários servidores receberão parcelas da computação do utilizador tendo ainda possivelmente que aceder a bases de dados remotas onde se encontram os dados.

Visto que a comunicação poderá envolver uma interacção mais ou menos longa entre as diversas entidades, é desejável que sejam disponibilizadas capacidades de comunicação segura orientadas à sessão e orientadas à mensagem.

4.1 Comunicação baseada em Sessão

No que se refere ao estabelecimento de uma sessão, a maior parte dos sistemas existentes, como o *Globus* e o *Legion*, baseiam a segurança das suas comunicações no estabelecimento de uma ligação SSL ou TLS[38] entre os interlocutores. Esta é efectivamente uma opção interessante, visto que o protocolo http consegue passar facilmente por sistemas de firewalls e combinado com o protocolo SSL consiste num mecanismo de segurança razoavelmente seguro. O protocolo JXTA assenta sobre o mesmo modelo de comunicação, sendo ainda possível o estabelecimento de redes privadas virtuais[65].

No entanto o protocolo SSL não contempla a delegação de credenciais, razão pela qual os criadores do *Globus* adaptaram o protocolo https, criando um novo protocolo designado de httpg. Conseguem desta forma estar alinhados com os objectivos definidos para o módulo GSI[82].

4.2 Comunicação baseada em Mensagem

No que se refere à segurança na troca de mensagens individuais, mais uma vez os mecanismos utilizados são baseados na utilização de tecnologia de chave pública para autenticar e cifrar as mensagens, diferindo apenas o norma utilizado em cada plataforma.

O *Globus* utiliza o GSI XML-signature e o JXTA utiliza o *Cryptographic Message Syntax* (CMS), do *S/MIME Working Group*, de forma a suportar o protocolo *Secure Message Transport and Endpoint Service*.

De forma a facilitar o estabelecimento de um modelo de segurança numa aplicação em grelha, o *Legion* permite definir uma determinada mensagem (que em *Legion* equivale a invocar o método de um objecto), associada a um de três níveis de segurança existentes. Assim, o nível de segurança mínimo consiste em enviar a mensagem em claro e não assinada. Caso se pretenda pode-se seleccionar o modo protegido, que faz com que todas as credenciais enviadas na mensagem sejam cifradas. Este é o modo definido por omissão sempre que uma mensagem contém credenciais. Finalmente, sempre que se pretender cifrar o próprio conteúdo das mensagens deve-se escolher o modo privado. Nos casos em que a invocação de determinado método desencadeia a invocação de outros submétodos, o sistema *Legion* garante que as subinvocações são realizadas pelo menos com o nível de segurança da invocação pai, de forma a prevenir a transmissão acidental de informação sensível.

Por último o sistema *Legion* gera ainda um número aleatório em cada invocação de um método, de forma a permitir agrupar todas as mensagens geradas, e constituir ainda uma chave secreta partilhada entre os vários interlocutores.

Tabela 3. Mecanismos de Comunicação

Globus	SSL e TLS
Consh	Não existente
Gridbank	SSL e TLS
EDG	SSL e TLS
Nordugrid	SSL e TLS
JXTA	TLS
Crisis	SSL
NASA IPG	*Globus*
Legion	Mecanismos de Cifra
Alchemi	Não existente
Unicore	SSL
IGENV	Secure Shell

Tabela 4. Mecanismos de Isolamento

Globus	Contas Locais
Consh	*Janus*
Gridbank	Contas Locais
EDG	Contas Locais
Nordugrid	Contas Locais
JXTA	Não existente
Crisis	*Janus*
NASA IPG	Contas Locais
Legion	Contas Locais
Alchemi	.Net CAS (Code Acess Security)
Unicore	Contas Locais
IGENV	Contas Locais

4.3 Comunicação em Grupo

Uma falta evidente nos protocolos de comunicação utilizadas prende-se com a ausência de mecanismos de comunicação segura em grupo. Efectivamente, só a plataforma JXTA permite enviar uma determinada mensagem cifrada para um conjunto de destinatários [4]. Não permite contudo a especificação de semânticas de correcção associadas à comunicação em grupo [22], como por exemplo especificar garantias de que a mensagem será entregue a todos os destinatários ou a nenhum, que facilitariam bastante a implementação de mecanismos de tolerância a faltas.

5 Isolamento

Mesmo nos casos em que um utilizador está perfeitamente identificado pelo sistema e os seus privilégios de acesso lhe permitem realizar uma determinada operação deve ser considerada a hipótese de que o utilizador, ou um atacante que tenha obtido as suas credenciais, tente utilizar essa funcionalidade para efectuar outras operações não permitidas. Este problema, que é comum nos sistemas distribuídos tradicionais, é aumentado nos sistemas de *grid* que prestam um serviço de CPU, ou seja, que servem de hospedeiros para código remoto.

É impossível na quase totalidade das situações garantir apriori a inocuidade de uma determinada aplicação [3], pelo que se torna necessário identificar mecanismos que limitem a possibilidade de danos no sistema [27]. Cada aplicação ou processo deve ficar assim circunscrita a um perímetro de segurança, usualmente designado de *sandbox*. É este o mecanismo utilizado pela maioria das plataformas de execução segura, das quais a mais vulgar nas plataformas de *grid* é o sistema *Janus*, utilizado no CRISIS [10] e no Consh [3].

As plataformas Java[45] e .Net[64, 59, 104] possuem também algumas das funcionalidades necessárias para assegurar o cumprimentos destes objectivos. Nomeadamente, a plataforma Alchemi, utilizada para construir clusters de sistemas desktop Windows, utiliza a sandbox do .Net para proteger os desktops individuais da execução de código remoto maligno[63].

O sistema *Legion* utiliza os próprios mecanismos de separação de processos do sistema operativo para garantir a segurança na execução dos seus serviços. Como já referimos anteriormente, num sistema *Legion* o administrador cria uma ou várias contas genéricas destinadas à utilização dos serviços. Após autenticação de um determinado utilizador, é-lhe fornecida uma destas contas, que é depois utilizada para invocar os serviços. Um possível problema prende-se com a necessidade de ter no sistema um número de contas igual ou superior ao número máximo de clientes a aceder aos serviços, na forma de uma *pool* de contas a serem atribuídas à medida que os utilizadores acedem. Caso contrário, vários utilizadores teriam que ser associados ao mesmo utilizador, com a possibilidade de acederem a informação confidencial nos processos dos outros utilizadores. E este é efectivamente um dos maiores problemas referidos neste aspecto do *Legion*, devido ao facto da maior parte dos administradores de limitarem a criar uma conta genérica, com todas as desvantagens inerentes a diversos processos correrem sob a mesma identificação local. Adicionalmente existem situações em que alguns serviços têm que correr com os privilégios do administrador do sistema, e nesses casos torna-se ainda necessário impedir a execução simultâneo de vários processos, no sentido de impedir que possam aceder a informação temporária sensível utilizada por outros processos.

Butt[17] apresenta um sistema que permite a execução segura de aplicações em ambientes de *grid*. Isto é conseguido através da utilização conjunta de um ambiente de shell restricto e um monitor de segurança (runtime monitor). A monitorização dos processos é realizada recorrendo à funcionalidade de *ptrace* do *Unix* e ao directório do sistema de ficheiros */proc*, numa aproximação similar à utilizada pelo *Janus*.

6 Delegação

Uma das grandes vantagens do conceito de grelha prende-se com a possibilidade de acesso a um leque alargado de recursos disponibilizados por diversas entidades na organização virtual.

Nesse sentido, uma determinada computação em *grid* pode necessitar da conjugação das funcionalidades presentes em diversos recursos e sistemas. Usualmente seria necessário que um utilizador se autenticasse de cada vez que necessita de utilizar um recurso, de forma a permitir validar os seus privilégios. No entanto verifica-se que uma computação pode ter uma duração elevada (várias horas, dias ou mesmo semanas), pelo que não se torna viável exigir ao utilizador este nível de interacção. Por outro lado, a contínua utilização da chave privada do utilizador aumentaria o risco do seu compromisso, pelo que se tornou necessário encontrar formas alternativas de permitir uma única autenticação [53, 105].

Esta forma de *single sign-on* é normalmente disponibilizada em *grid* através da geração de um certificado de representante, assinado pelo próprio utilizador, e que permite a um determinado processo realizar as operações necessárias em seu lugar [92]. Este certificado é armazenado em claro no sistema do utilizador, mas o facto do certificado ser gerado com um tempo de vida bastante curto minimiza eventuais vulnerabilidades existentes. Mesmo que o certificado seja comprometido, a sua utilidade é temporalmente limitada. O facto de não ser necessário recorrer a uma autoridade de certificação para obter este novo certificado permite uma grande flexibilidade na delegação de privilégios, mas como é óbvio gera algumas vulnerabilidades, pois uma autoridade de certificação tem preocupações globais de gestão da segurança bastante superiores a um vulgar utilizador [61].

A principal questão no que se refere à geração deste novo certificado prende-se com a determinação da quantidade correcta de privilégios a transmitir. Na aproximação mais simplificada o representante possuirá todos os privilégios do utilizador original. Esta foi aliás a única possibilidade nas versões originais dos sistemas *Legion* [88] e *Globus* [35, 76, 88]. No entanto, esta aproximação contradiz o princípio básico de segurança segundo o qual todas as entidades deverão actuar com o nível mínimo de segurança necessário para realizarem as suas acções, e não mais do que isso (*least-privilege principle*) [85].

Para calcular o nível óptimo de privilégios a transferir seria necessário, no limite, analisar todas as possíveis linhas de execução da aplicação. Em particular, num sistema de *grid* a aplicação pode ter que interagir com uma miríade de entidades, desde um *scheduler* que selecciona os recursos a utilizar até aos próprios recursos disponíveis. De notar que os recursos a utilizar podem só ser conhecidos no decorrer da própria computação, impossibilitando processos de obtenção de todas as credenciais apriori. Cada uma destas entidades possui as suas próprias políticas de segurança que, em conjunção com as políticas da organização virtual, constituem o universo sobre o qual deve ser calculado o nível mínimo de privilégios. Devido à complexidade associada a estas decisões, a maior parte dos sistemas opta por realizar uma delegação sem restrições, apesar das vulnerabilidades associadas.

O sistema *Globus* base só permite delegação ilimitada. No entanto diversos sistemas foram já disponibilizados no sentido de incrementar a semântica e facilidade de utilização desta funcionalidade. O sistema CAS (*Community Authorization Service*) [76], também discutido na secção de Autorização, possibilita a restrição dos privilégios transferidos através do *proxy certificate*. Adicionalmente, os certificados de representante do sistema *Globus* já estão baseados na proposta do IETF para *X.509 Proxy Certificates*[96], prevendo a inclusão no futuro de delegação limitada de privilégios.

O sistema EDG, baseado na plataforma *Globus*, também utiliza um mecanismo de delegação ilimitada. Valida no entanto que a designação (*distinguished name*) existente num certificado de representante seja iniciada pela designação existente no certificado original [13]. É afirmado que este mecanismo previne que um utilizador se possa fazer passar por outro através da geração de um certificado de representante fraudulento (e.g. john-doe-proxy-001 gerado a partir de john-doe fornece uma indicação mais fiável acerca da proveniência do certificado do que proxy-247). Na realidade, a utilidade desta modificação é reduzida, visto que os certificados de representantes são validados automaticamente pelos recursos locais, que realizam a validação de toda a cadeia de certificados até um certificado reconhecido, no sentido de se validar a sua autenticidade.

O sistema CRISIS, tal como os sistemas *Globus* e *Legion*, permite também realizar a delegação de privilégios através da criação de um proxy de representante. No entanto, este sistema obriga a que estes proxies sejam validados por um OLA (OnLine Agent) previamente à sua utilização [10]. Desta forma potencia-se a rápida revogação dos certificados comprometidos de um utilizador caso seja necessário. Um dos problemas da plataforma CRISIS é que não se adapta facilmente a sistemas de *grid* destinados a processamento intensivo, pois não permite a criação de subprocessos sobre os quais sejam delegados os privilégios dos processos pai.

Sistemas mais avançados permitem especificar os privilégios a delegar ao certificado de representante. O sistema *Legion* permite restringir a delegação realizada em 3 aspectos essenciais: os métodos que podem ser invocados, os objectos que podem utilizar privilégios delegados ou ser alvo de chamadas desses mesmos objectos, e ainda o alcance

temporal de determinada delegação de privilégios [88].

O primeiro aspecto acima referido, a limitação dos métodos que podem ser invocados, representa uma tarefa incrivelmente complexa, já que todos os métodos e submétodos invocados por uma determinada aplicação têm que ser listados no certificado de delegação. Isto aumenta ainda de forma substancial a dimensão do mesmo certificado. Uma aproximação utilizada tem sido a de automatizar esta tarefa através da utilização de um compilador que navega por todas as execuções possíveis, marcando cada um dos métodos invocados. No entanto, esta aproximação retira ao utilizador a possibilidade de discernir possíveis invocações de métodos que possam comprometer a segurança, e que não sejam visíveis à primeira vista entre largas dezenas de métodos. Outra aproximação seguida tem sido a de classificar os métodos existentes mediante o seu efeito potencial, como por exemplo, método de escrita e métodos de leitura. Esta alternativa está a ser experimentada no sistema *Legion*, em situações onde todos os métodos de leitura são autorizados, mas os métodos de escrita têm que ser explicitamente enumerados. Uma solução que se poderia ainda adoptar seria a de no início da execução impedir a invocação de todos os métodos e, interactivamente, questionar o utilizador acerca da permissão de cada um dos métodos encontrados na linha de execução.

Outras das possibilidades discutidas é a especificação dos objectos que podem invocar um determinado método, por um lado, e que podem ser alvo de invocação, por outro lado. A utilização desta funcionalidade é bastante facilitada pelo facto do sistema *Legion* ser baseado num modelo de objectos, e como tal um utilizador poder especificar as características da delegação socorrendo-se da hierarquia de classes, sem ter que designar individualmente cada um dos objectos. Assim, todos os recursos existentes numa plataforma *Legion* têm que ser registados centralmente, ficando associados ao sistema onde estão hospedados. Quando um utilizador quer utilizar um determinado tipo de recursos, designa quais os sistemas hospedeiros em que tem confiança. Além disso poderá também especificar quais as entidades (objectos) em que confia para realizar a operação.

Finalmente, e tal como nos certificados de representantes gerados no *Globus*, no EDG e no CRISIS, o utilizador pode também no *Legion* especificar o limite temporal de validade da delegação de privilégios. Este aspecto tem grande influência na segurança global do sistema pois um prazo de validade excessivamente longo poderá permitir a um atacante a descodificação do certificado ou mesmo a realização de ataques de repetição, e um prazo demasiado curto obrigará o utilizador a gerar continuamente novos certificados de representante. Por outro lado, um prazo curto permite também a um utilizador um maior controle sobre as operações realizadas, visto que periodicamente terá que gerar novos certificados para a realização das operações pretendidas.

Tabela 5. Mecanismos de Delegação

Globus	*User Proxy*
Consh	Não Existente
Gridbank	*User Proxy*
EDG	*User Proxy*
Nordugrid	*User Proxy*
JXTA	Não Existente
Crisis	Transfer certificates
NASA IPG	*User Proxy*
Legion	Credenciais Restrictas
Alchemi	Não Existente
Unicore	Não Existente
IGENV	Não Existente

O sistema MyProxy disponibiliza um serviço web que permite realizar a delegação dos privilégios sem ter que recorrer a uma linha de comando remota. Isto facilita bastante a implementação de portais web para acesso dos utilizadores a recursos da organização virtual. O serviço basicamente funciona como uma interface web para o GSI[73]. Através desta interface um utilizador pode-se registar no serviço e futuramente obter um novo certificado de representante através da apresentação da sua identificação e uma palavra chave escolhida previamente.

Gannon[43] refere que determinado processo poderá necessitar de transmitir parte dos seus privilégios a outro processo no sentido de executar parte da tarefa que lhe foi incumbida. Ele utiliza assim as funcionalidades de delegação do GSI para realizar essa delegação (neste caso ilimitada visto que o GSI não permite restringir os privilégios a delegar).

7. Revogação de Privilégios

Visto que a tecnologia de certificados digitais, presente na maior parte dos sistemas de *grid*, baseia a sua segurança na confidencialidade da chave privada associada a cada certificado X.509, torna-se necessário a implementação de mecanismos que permitam invalidar um certificado (ou uma credencial) caso a chave privada tenha sido comprometida.

Dois aspectos em particular tornam a revogação de privilégios crítica em sistemas de *grid*, quando comparados com sistemas distribuídos tradicionais. Em primeiro lugar a utilização de certificados de representantes (*proxy certificates*) cria vários certificados em simultâneo para cada utilizador, o que dificulta a sua gestão e controle. Como tal, é difícil para um utilizador verificar se determinado certificado foi comprometido. Em segundo lugar, o facto dos certificados de representação serem assinados pelo próprio utilizador e não por uma autoridade de certificação cria vulnerabilidades adicionais para o certificado gerado, visto

que dificilmente se considerará o computador do utilizador como um local seguro para a realização deste tipo de operações.

Em oposição a estas questões deve-se no entanto também referir que o curto tempo de validade normalmente associado aos certificados de representação diminui bastante as vulnerabilidades existente, contribuindo para que as vantagens na sua utilização ultrapassem geralmente as desvantagens[92].

O modo mais vulgar de revogação de certificados corresponde à emissão de listas de revogação de certificados (*Certificate Revocation List* ou CRLs) que, estando disponíveis a todos os utilizadores em servidores próprios, lhe permitem validar a correcção de determinado certificado que lhes é apresentado [15]. Este é o método utilizado pelo *Globus*, e consequentemente pelo EDG. As CRLs são renovadas periodicamente pelas entidades de certificação, e são automaticamente descarregadas pelos módulos de segurança do *Globus* e do GSI, não afectando a execução das aplicações.

Na plataforma JXTA, o modelo de funcionamento é semelhante, apesar de as entidades responsáveis pelas diversas operações serem diferentes, em virtude das características particulares desta arquitectura. De facto, deve-se notar que as autoridades de certificação referidas no ponto anterior poderão mesmo não existir, visto que os utilizadores têm a hipótese de simplesmente auto-assinar e co-assinar os seus próprios certificados [65]. Assim, existe habitualmente em cada *peer group* um serviço de Anúncio de Revogação (semelhante ao servidor de CRLs), que é usado por qualquer utilizador cujo certificado esteja comprometido. Basta-lhe assinar uma mensagem de revogação com o próprio certificado que está comprometido e enviar para o serviço de revogação, que se encarregará de o distribuir por todos os outros utilizadores do grupo, e em especial pelos nós que realizam co-assinatura dos certificados. Esta é assim uma abordagem pull, em que as CRL's são distribuidas, em lugar de uma abordagem push, em que os utilizadores devem periodicamente consultar os servidores e verificar a validade dos certificados. De notar que devido ao funcionamento ad-hoc da rede, que permite desconexões dos diversos nós, este modelo de funcionamento poderá introduzir algumas vulnerabilidades, permitindo habitualmente a utilização de certificados comprometidos em partições da rede distintas daquela onde a revocação foi realizada.

Uma revogação eficiente é um dos principais objectivos do sistema CRISIS. Belani [10] refere que a existência dos agentes online (OLA) que co-assinam os certificados com um prazo de validade reduzido permite revogar qualquer certificado com a garantia que este não será utilizado por um período alargado. No entanto, entendemos que esta solução é similar à utilizada no conceito de certificados de representante, visto que o sistema continua a estar vulnerável durante a validade do certificado. Um conceito complementar interessante, mas que não é detalhado em [10] prende-se com a possibilidade de revogar um certificado de representante logo que a sua função seja completa, como é o caso de uma computação que termina.

8 Políticas

A utilização de motores de políticas em sistemas distribuídos permite distinguir de forma eficaz a gestão e a implementação dos serviços disponibilizados[6, 44, 108, 87, 58, 55]. Este é um tópico relativamente pouco explorado na literatura de computação em grelha[101, 92], mas dada a complexidade inerente a estes ambientes, seria de todo desejável a sua incorporação[95].

Visto que na maior dos sistemas actuais a aplicação das políticas é realizada localmente, ou definida de forma estática entre instituições, a gestão integrada das políticas não se tem apresentado como um aspecto crítico. No entanto, à medida que as organizações virtuais se tornam mais complexas, assumindo uma grande dimensão e/ou adoptando um modelo ad-hoc, torna-se necessário garantir que as políticas de segurança das diversas entidades sejam interoperáveis e eficazes[12, 48].

A norma WS-Authorization apresenta-se como um passo seguro neste sentido [107]. Nesta norma, a utilização de SAML (*Security Assertion Markup Language*) permite definir de uma forma normalizada os requisitos de segurança de determinado recurso [9]. Esta definição permite não só a um mecanismo de segurança aplicar as regras necessárias, como a um potencial utilizador avaliar a sua capacidade para utilizar o mesmo recurso.

Galis [40] sugere uma aproximação à gestão de políticas em *grid* através da adopção do conceito de active network management, utilizando na gestão de redes IP. O sistema proposto integra a arquitectura OGSA e o sistema PBM (*Policy Based Management*) no sentido de facilitar a gestão de um sistema de *grid*.

Pearlman [76] não utiliza nenhuma linguagem em particular no sistema CAS preferindo tratar a política como uma caixa preta que cada sistema local deve saber interpretar quando necessário [62].

O sistema IGENV [93] possui um motor de políticas baseado em regras que controla o acesso aos recursos. O motor de regras é alimentado por ficheiros de configuração, sendo a avaliação realizada tendo em conta estas regras e a informação recolhida por agentes de monitorização instalados no sistema. As políticas disponibilizadas dividem-se em 4 tipos principais:

1. As políticas de sessão definem os parâmetros segundo os quais se tem que reger a utilização dos recursos durante uma sessão do utilizador (p.e. a quantidade máxima de recursos que podem ser consumidos durante uma sessão).

2. As políticas de contas-utilizador definem os recursos a que determinado utilizador (que é sempre associado a uma conta individual ou genérica) pode aceder.

3. As políticas aplicacionais definem a configuração das diversas aplicações

4. Finalmente, as políticas de QoS definem as condições associadas à aplicação de métricas de qualidade de serviço no sistema, bem como as penalidades respeitantes a qualquer violação da mesma qualidade de serviço.

Keahey[57] propõe extender o módulo GRAM (*Grid Resource Allocation and Management*) do sistema *Globus* de forma a introduzir uma nova linguagem de políticas. Esta linguagem, que utilizaria o RSL (*Resource Specification Language*) para descrever os recursos e as suas propriedades, permitiria definir políticas de segurança complexas ao nível de toda a organização virtual.

O sistema SESAME [110] introduz o conceito de DRBAC (*Dynamic Role Based Access Control*) no universo da computação em grelha. Através da especificação de máquinas de estado este sistema permite que os utilizadores assumam diferentes roles dependendo do contexto onde se encontram. As políticas são especificadas em XML e definem as permissões de cada role bem como as transições entre os diversos estados. Estas transições utilizam a informação de contexto que pode estar associada ao objecto (localização, data, estado do recurso, etc) ou ao sistema hospedeiro (carga, disponibilidade, etc).

Ryutov e Neuman apresentam uma infraestrutura genérica que permite integrar sistemas de autorização heterogéneos num sistema distribuído [79]. Em particular realizaram um protótipo sobre o sistema *Globus*, de forma a demonstrar que a infrastrutura é aplicável a ambientes de *grid* computing. Introduzem uma nova linguagem designada de EACL (*Extended Access Control List*) de forma a permitir a definição de políticas de uma forma normalizada e que permita definir condições de acesso avançadas (como por exemplo limites na utilização de um serviço). Em paralelo utilizam a interface GAA-API (*Generic Authorization and Access Control API*) para que as diversas aplicações possam realizar pedidos de autorização a este sistema de forma unificada. Assim, qualquer sistema na *grid* pode realizar autonomamente as suas decisões, ou em alternativa utilizar informação integrada ao nível da *grid*. Referem ainda a necessidade de, não só realizar decisões de autorização mas, complementarmente, monitorizar a real utilização dos recursos, de forma a garantir que não são cometidos abusos [80].

Galiasso [39] apresenta uma plataforma de políticas que permite a integração de diversas políticas complexas existentes numa federação de entidades. Apesar de não se referir explicitamente ao *grid*, consideramos que o seu sistema poderia ser aplicado neste tipo de ambientes. A plataforma proposta contém uma linguagem de políticas que pretende representar todas as políticas existentes nos diversos nós da federação. A expressividade desta linguagem permite definir um conjunto elevado de semânticas, incluindo diversas políticas baseadas na história, como separação de deveres e chinese wall.

Kang propõe a utilização de um motor de políticas de segurança denominado SALSA [56] para realizar o controle de workflows entre organizações. Este sistema poderá ser facilmente adoptado a organizações virtuais, permitindo um controle eficaz deste tipo de processo. O motor referido permite a especificação de políticas baseadas na história e dependentes do contexto. Pode ser utilizado complementarmente aos mecanismos de segurança existentes em cada uma das organizações, visto que se baseia na instalação de um gestor de segurança em cada um dos sistemas.

Uszok refere a integração do sistema Kaos [97] com a plataforma *Globus*. Esta integração permitiu a aplicação de políticas complexas sobre os recursos disponibilizados. O próprio motor de autorização é um serviço web ao qual os outros serviços acedem no sentido de se registarem e especificarem as suas políticas de acesso. De seguida, quando um utilizador pretende aceder a um destes recursos, contacta o serviço Kaos, que valida os seus privilégios. Em caso de sucesso emite uma credencial que lhe permite então aceder ao serviço em questão, numa aproximação similar à do sistema CAS.

A arquitectura de controle de acessos proposta pela OASIS permite a aplicação de política de segurança baseada no modelo RBAC [6]. Estas políticas podem ser definidas num contexto interorganizacional, adaptando-se assim às necessidades de uma organização virtual. Ao contrário de outras platformas de políticas, o OASIS não suporta a delegação de privilégios, optando em alternativa por disponibilizar o conceito de appointments (por exemplo um hospital não pode desempenhar o papel de médico, mas pode emitir credenciais para outras entidades certificando-as).

Sundaram [89] desenvolveu um motor de políticas de utilização que incorporou no gestor de alocação (resource broker) EZ*grid*. Desta forma os administradores dos recursos podem especificar os parâmetros de utilização dos recursos que são depois utilizados para realizar decisões de alocação, maximizando a utilização dos recursos e a segurança no acesso aos mesmos.

9. Conclusões

Apresentámos, no decorrer deste artigo, um panorama alargado do que consideramos os aspectos principais de segurança nas plataformas de computação em grelha.

Os certificados X.509 são utilizados por grande parte dos sistemas para identificar univocamente os seus utilizadores.

Efectivamente, constituem uma solução bastante mais sofisticada e escalável para identificação unívoca do que um simples *login* de sistema, além de permitirem, adicionalmente, uma autenticação segura baseada num protocolo como o TLS. Associados aos certificados X.509 estão as autoridades de certificação, actualmente um dos pontos críticos no que se refere à interoperabilidade entre organizações virtuais. A questão da inter-certificação de autoridades de certificação não tem recebido grande atenção por parte da comunidade de *grid*, baseando-se na prática os sistemas na existência de uma autoridade-raíz única. Prevemos no entanto que a disseminação da tecnologia e a constituição de organizações virtuais de grande dimensão ou que adoptem um modelo mais ad-hoc obrigue a avanços nesta área. É comum no entanto encontrarem-se ainda soluções que obrigam a uma carga administrativa bastante elevada, como a actualização manual dos utilizadores de toda a *grid* em cada um dos sistemas, ou que por outro lado limitam apriori a flexibilidade na gestão dos recursos, como a especificação de contas locais únicas para mapeamento dos utilizadores da *grid*.

Em termos de autorização, os mecanismos locais como a aplicação de privilégios de contas *Unix* continuam a constituir a opção mais utilizada, sendo mesmo a única em diversos casos. Identificamos duas razões primordiais para este facto: Em primeiro lugar os administradores de sistema encontram-se ainda renitentes relativamente a cederem controle dos seus recursos a uma terceira entidade, exigindo assim que a palavra final no que respeita a um acesso seja a sua. Esta aproximação tem vindo a ser abandonada em prol de ferramentas como o CAS (Community Authorization Service), que permitem delegar parte da responsabilidade na definição de políticas de acesso para um serviço existente na *grid*. Por outro lado, a heterogeneidade de sistemas encontrados na *grid* leva a que seja relativamente difícil incorporar diversas arquitecturas de segurança numa plataforma única, sendo mais fácil delegar em cada sistema a gestão/controle dos acessos.

A comunicação entre nós dos sistemas de *grid* é baseada por um lado em normas já bastante reconhecidas na área dos sistemas distribuídos, como é o caso do TLS, mas por outro lado introduz também bastantes inovações, mais especificamente a utilização das novas normas de *webservices* para comunicação segura: o *WS-SecureConversation*, o *WS-SecureMessage* e o *XML-Signature*.

A delegação de privilégios é outra das características essenciais de um sistema de *grid* que pretenda suportar computação de grande escala (disponibilizando CPU, armazenamento ou rede). De forma a obstar que um utilizador necessite de autenticar a sua identidade de cada vez que necessita de aceder a um novo recurso da *grid*, sistemas como o Globus e o *Legion* introduzem o conceito de representante do utilizador (*user proxy*). Este é constituído basicamente por um novo certificado digital, gerado na hora, e assinado pelo certificado digital do utilizador. A chave privada é guardada em claro, de forma a permitir autorizar automaticamente novos pedidos e a validade temporal do certificado é de poucas horas, no sentido de limitar o impacto de um eventual compromisso da chave. Sistemas como o CAS (Community Authorization Service) permitem restringir os privilégios especificados nestes certificados, de forma a reduzir ainda mais o impacto de qualquer vulnerabilidade. Por outro lado sistemas como o MyProxy permitem a um utilizador gerar certificados de representante em qualquer local onde se encontrem através do acesso a um site web, ao invés de terem que registar-se no seu sistema de trabalho.

No que se refere a motores de políticas aplicados à *grid*, encontram-se alguns protótipos e referências na literatura mas, que seja do meu conhecimento, não existem motores de políticas suficientemente abrangentes para gerir a complexidade inerente à operação de uma *grid*. Se por um lado diversos sistemas propõem a utilização de políticas para gestão dos recursos, como o CAS, o EZ*grid* e o Gridbank, estas não disponibilizam uma linguagem suficientemente expressiva para suportar um modelo de utilização que se prevê extremamente dinâmico. É necessário levar em linha de conta que num sistema de *grid* totalmente integrado, a maior parte das decisões no que se refere a alocação de recursos será realizada dinâmicamente, podendo a própria informação necessária para o escalonamento ser considerada confidencial, e portanto sujeita a controle de acessos. É necessário não só validar que um recurso se ajusta às necessidades, mas também que todas as tarefas a ele associadas podem ser realizadas mediante as permissões existentes, permissões essas que podem ser alteradas durante a própria alocação do recurso. Entre os sistemas analisados destacamos as plataformas SALSA[56] e a proposta por Galiasso[39], que suportam políticas avançadas, incluindo o conceito de políticas baseadas na história. Não identifiquei no entanto nenhuma plataforma que suporte outro conceito fundamental, o de obrigações[42].

Em suma, a tecnologia de computação em grelha disponibiliza uma evolução necessária aos sistemas distribuídos tradicionais, focando aspectos como a virtualização de recursos, a integração de plataformas heterogéneas e a utilização de recursos 'on-demand'. Encontra-se ainda numa fase de pleno desenvolvimento, apesar da integração recente com a tecnologia de *webservices* (modelo OGSA) lhe ter adicionado uma evidente maturidade e massa crítica. Nota-se também uma grande homogenia na adopção de normas do GGF (*Global grid Forum* e da implementação de referência da arquitectura OGSA, o *Globus Toolkit v4*. De futuro, prevemos que a crescente complexidade no modelo de partilha de recursos obrigará à introdução de plataformas de gestão integradas, possibilitando a definição de re-

gras de utilização e segurança através de políticas de alto nível. Esta evolução permitirá a constituição plena da *Grid Economy*[8], que pretende transformar recursos computacionais em *commodities* como a electricidade e o telefone.

Referências

[1] Ogsa security working group (http://www.cs.virginia.edu/ humphrey/ogsa-sec-wg/).

[2] Pkcs #11 v2.20: Cryptographic token interface norma. Technical report, RSA Laboratories, June 2004.

[3] A. Alexandrov, P. Kmiec, and K. Schauser. Consh: A confined execution environment for internet computations. In *Proceedings of the USENIX Annual Technical Conference*, December 1998.

[4] J. E. Altman. Pki security for jxta overlay networks. Technical report, IAM Consulting,, February 2003.

[5] A. Arsenault and S. Farrell. Securely available credentials - requirements. Technical report, IETF, August 2001.

[6] J. Bacon and K. Moody. Toward open, secure, widely distributed services. *Communications of the ACM*, 45, 2002.

[7] M. Baker, R. Buyya, , and D. Laforenza. Grids and grid technologies for wide-area distributed computing. *Software Practice and Experience*, 32, December 2002.

[8] A. Barmouta and R. Buyya. Gridbank: A grid accounting services architecture (gasa) for distributed systems sharing and integration. In *Proceedings of the 17th Annual International Parallel and Distributed Processing Symposium (IPDPS 2003)*, Nice, France, April 2003.

[9] B. P. Barrett. Introduction to ws authorization - cs551. Technical report, University of Virginia, 2004.

[10] E. Belani, A. Vahdat, T. Anderson, and M. Dahlin. The crisis wide area security architecture. In *Proceedings of the Seventh USENIX Security Symposium*, January 1998.

[11] V. Berstis. Fundamentals of grid computing. Technical report, IBM, 2002.

[12] C. Bidan and V. Issarny. Dealing with multi-policy security in large open distributed systems. In *Proceedings of the 5th European Symposium on Research in Computer Security*, September 1998.

[13] D. Bosio, J. Casey, A. Frohner, and L. G. et al. Next generation eu datagrid data management services. In *Proceedings of Computing in High Energy Physics (CHEP 2003)*, California, USA, March 2003.

[14] I. Boutboul. Manage credentials and access control in a grid application. *IBM developerWorks*, November 2003.

[15] P. J. Broadfoot and A. P. Martin. A critical survey of grid security requirements and technologies. Technical report, Oxford University Computing Laboratory, August 2003.

[16] R. Butler, V. Welch, D. Engert, I. Foster, S. Tuecke, J. Volmer, and C. Kesselman. A national-scale authentication infrastructure. *Computer*, 33(12):60–66, 2000.

[17] A. Butt, S. Adabala, N. Kapadia, R. Figueiredo, and J. Fortes. Fine-grain access control for securing shared resources incomputational grids. In *Proceedings of the Parallel and Distributed Processing Symposium*, Ft. Lauderdale, USA, 2002.

[18] C. Catlett. normas for grid computing: Global grid forum. *Journal of Grid Computing*, 1, March 2003.

[19] R. Chen and W. Yeager. Poblano: A distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems, 2001.

[20] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientic datasets. *Journal of Network and Computer Applications*, 2001.

[21] H. Chivers. Grid security: Problems and potential solutions. Technical report, Department of Computer Science, University of York, 2003.

[22] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems - Concepts and Design (3rd edition)*. Addison-Wesley Publishers, 2001.

[23] J. Crume, A. Buecker, K. Gordon, J. Heid, J. Pannu, J. Sanders, and A. Schmengler. On demand operating environment: Security considerations in an extended enterprise. Technical report, IBM, 2004.

[24] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke. From open grid services infrastructure to wsresource framework: Refactoring & evolution. Technical report, GGF,, December 2004.

[25] K. Czajkowski, C. Kesselman, S. Fitzgerald, and I. Foster. Grid information services for distributed resource sharing. In *Proceedings of The 10th IEEE International Symposium on High Performance Distributed Computing*, 2001.

[26] E. Damiani, S. D. C. di Vimercati, and P. Samarati. Towards securing xml web services. In *Proceedings of the 2002 ACM Workshop on XML Security*, Fairfax VA, USA, 2002.

[27] A. Dan, A. Mohindra, R. Ramaswami, and D. Sitaram. Chakra vyuha (cv): a sandbox operating system environment for controlled execution of alien code. Technical report, IBM, 1997.

[28] M. Draoli, G. Mascari, and R. Puccinelli. Datagrid: Project presentation. Technical report, European Data Grid, July 2001.

[29] D. Erwin. Unicore plus final report - uniform interface to computing resources. Technical report, UNICORE Forum, 2003.

[30] J. G. et al. First prototype of the crossgrid testbed. In *Proceedings of the Grid Computing: First European Across Grids Conference*, Santiago de Compostela, Spain,, February 2004.

[31] A. Ferrari, F. Knabe, M. Humphrey, S. Chapin, and A. Grimshaw. Accountability and control of process creation in metasystems. *Proceedings of the 7th International Conference on High Performance Computing and Networking Europe (HPCN Europe 99)*, April 1999.

[32] I. Foster. The grid: A new infrastructure for 21st century science. *Physics Today*, 54(2), 2002.

[33] I. Foster and C. Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., 2003.

[34] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. Grid services for distributed system integration. *Computer*, 35, June 2002.

[35] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. Technical report, Global Grid Forum, 2002.

[36] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. *Proceedings of the ACM Conference on Computers and Security*, 1998.

[37] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. In *Proceedings of the Intl. J. Supercomputer Applications*, 2001.

[38] A. Freier, P. Karlton, and P. Kocher. Secure socket layer 3.0. *Internet Draft*, March 1996.

[39] P. Galiasso, O. Bremer, J. Hale, S. Shenoi, D. Ferraiola, and V. Hu. Policy mediation for multi-enterprise environments. In *Proceedings of the 16th Annual Computer Security Applications Conference*, New Orleans, LA, USA, December 2000.

[40] A. Galis, J.-P. Gelas, L. Lefèvre, and K. Yang. Active network approach to grid management. *Lecture Notes in Computer Science*, 2659 / 2003, August 2003.

[41] P. Gama. A união faz a força. *Semana Informática*, June 2004.

[42] P. Gama and P. Ferreira. Obligation policies: An enforcement platform. In *Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, Stockholm, Sweden, June 2005.

[43] D. Gannon, R. Bramley, G. Fox, S. Smallen, A. Rossi, R. Ananthakrishnan, F. B. K. Chiu, M. Farrellee, M. Govindaraju, S. Krishnan, L. Ramakrishnan, Y. Simmhan, A. Slominski, Y. Ma, C. Olariu, and N. R.-C. and. Programming the grid: Distributed software components, p2p and grid web services for scientific applications. *Cluster Computing*, 5, July 2002.

[44] J. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of the IEEE Symp. Security and Privacy*, California, USA, 1982.

[45] L. Gong, M. Mueller, H. Prafukhandra, and R. Schemers. Going beyond the sandbox: An overview of the new security architecture in the java development kit 1.2. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, December 1997.

[46] T. Goss-Walter, R. Letz, T. Kentemich, H.-C. Hoppe, and P. Wieder. An analysis of the unicore security model. Technical report, Global Grid Forum, July 2003.

[47] A. S. Grimshaw and W. A. Wulf. Legion: Flexible support for wide-area computing. In *Proceedings of The Seventh ACM SIGOPS European Workshop: Systems Support for Worldwide Applications*, 1996.

[48] J. Hale, P. Galiasso, M. Papa, and S. Shenoi. Security policy coordination for heterogeneous information systems. In *Proceedings of the 15th Annual Computer Security Applications Conference*, Phoenix, Arizona, USA, December 1999.

[49] B. Hayes. Collective wisdom. *American Scientist*, 86, March-April 1998.

[50] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, and K. Stockinger. Data management in an international data grid project. In *Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID 2000)*, Bangalore, India, December 2000.

[51] J. H. Howard. An overview of the andrew file system. Technical report, Carnegie Mellon University, 1988.

[52] M. Humphrey, F. Knabe, A. Ferrari, and A. Grimshaw. A flexible security system for metacomputing environments. *Proceedings of the Network and Distributed System Security Symposium*, 2000.

[53] M. Humphrey and M. R. Thompson. Security implications of typical grid computing usage scenarios. In *Proceedings of the 10th IEEE Int. Symp. on High Performance Distributed Computing*, San Francisco, USA, 2001.

[54] W. E. Johnston, D. Gannon, and B. Nitzberg. Grids as production computing environments: The engineering aspects of nasa's information power grid. In *Proceedings of The 8th IEEE Symp. on High Performance Distributed Computing*, 1999.

[55] J. Joshi, A. Ghafoor, W. G. Aref, and E. H. Spafford. Digital government security infrastructure design challenges. *Computer*, 34, February 2001.

[56] M. H. Kang, J. S. Park, and J. N. Froscher. Access control mechanisms for inter-organizational workflow. In *Proceedings of the sixth ACM Symposium on Access Control Models and Technologies*, Chantilly, Virginia, USA, May 2001.

[57] K. Keahey and V. Welch. Fine-grain authorization for resource management in the grid environment. In *Proceedings of the Grid 2002 Workshop*, Ft. Lauderdale, USA, 2002.

[58] P. Kearney, J. Chapman, N. Edwards, M. Gifford, and L. He. An overview of web services security. *BT Technology Journal*, 22, March 2004.

[59] W. Kelly, P. Roe, and J. Sumitomo. G2: A grid middleware for cycle donation using .net. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, 2002.

[60] J. Licklider, R. Taylor, and E. Herbert. The computer as a communication device. *Science and Technology: For the Technical Men in Management*, April 1968.

[61] R. Lock and I. Sommerville. Grid security and its use of x.509 certificates. Technical report, Lancaster University, 2002.

[62] M. Lorch and D. Kafura. Supporting secure ad-hoc user collaboration in grid environments. In *Proceedings of The 3rd Int. Workshop on Grid Computing*, 2002.

[63] A. Luther, R. Buyya, R. Ranjan, and S. Venugopal. *Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework*. Wiley Press, New Jersey, USA, 2005 (in print).

[64] J. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla, and A. Murukan. Improving web application security: Threats and countermeasures. Technical report, Microsoft Corporation, June 2003.

[65] S. Microsystems. Security and project jxta. Technical report, Sun Microsystems, 2002.

[66] C. Moore, R. Johnson, and J. Detry. Adapting globus and kerberos for a secure asci grid. In *Proceedings of SC2001*, Denver, USA, November 2001.

[67] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, I. Foster, and S. Tuecke. The security architecture for open grid services. Technical report, Global Grid Forum, July 2002.

[68] Y. Nakamur, S. Hada, and R. Neyama. Towards the integration of web services security on enterprise environments. In *Proceedings of the Seventh ACM symposium on Access control models and technologies*, Narar City, Nara, Japan, January 2002.

[69] A. Natrajan, M. A. Humphrey, and A. S. Grimshaw. Grids: Harnessing geographically-separated resources in a multi-organisational context. In *Proceedings of The 15th Annual International Symposium on High Performance Computing Systems and Applications*, June 2001.

[70] A. Natrajan, A. Nguyen-Tuong, M. Humphrey, and A. Grimshaw. The legion grid portal, 2002.

[71] M. Niinimaki and V. Sivunen. Applying grid security and virtual organization tools in distributed publication databases. In *Proceedings of the 1st international symposium on Information and communication technologies*, September 2003.

[72] Z. Németh and V. Sunderam. Characterizing grids: Attributes, definitions, and formalisms. *Journal of Grid Computing*, 1, March 2003.

[73] J. Novotny, S. Tuecke, and V. Welch. An online credential repository for the grid: Myproxy. *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01)*, August 2001.

[74] R. Oppliger. Microsoft .net passport: A security analysis. *IEEE Computer*, July 2003.

[75] L. Pearlman, C. Kesselman, V. Welch, I. Foster, and S. Tuecke. The community authorization service: Status and future. *Proceedings of the 2003 Conference for Computing in High Energy and Nuclear Physics*, March 2003.

[76] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, page 50, 2002.

[77] L. Ramakrishnan, H. Rehn, J. Alameda, R. Ananthakrishnan, M. Govindaraju, A. Slominski, K. Connelly, V. Welch, D. Gannon, R. Bramley, and S. Hampton. An authorization framework for a grid based component architecture. In *Proceedings of the Third International Workshop on Grid Computing - GRID 2002*, Baltimore, MD, USA, November 2002.

[78] D. D. Roure, M. A. Baker, N. R. Jennings, and N. R. Shadbolt. The evolution of the grid. *Concurrency and Computation: Practice & Experience*, 2003.

[79] T. Ryutov and C. Neuman. Representation and evaluation of security policies for distributed system services. In *Proceedings of the DARPA Information Survivability Conference & Exposition*, Hilton Head, South Carolina, USA, January 2000.

[80] T. Ryutov and C. Neuman. The specification and enforcement of advanced security policies. In *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 02)*, Monterey, California, USA, June 2002.

[81] R. Sandberg, D. Golgberg, S. Kleiman, D. Walsh, and B. Lyon. Design and implementation of the sun network filesystem. pages 379–390, 1988.

[82] T. Sandholm and J. Gawor. Globus toolkit 3 core: A grid service container framework. Technical report, Global Grid Forum, May 2003.

[83] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere. Coda: A highly available file system for a distributed workstation environment. *IEEE Transactions on Computers*, 39, April 1990.

[84] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, 1995.

[85] B. Schneier. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.

[86] F. Siebenlist, V. Welch, S. Tuecke, I. Foster, N. Nagaratnam, P. Janson, J. Dayka, and A. Nadalin. Ogsa security roadmap. Technical report, Global Grid Forum, July 2002.

[87] E. G. Sirer and K. Wang. An access control language for web services. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, Monterey, California, USA, 2002.

[88] G. Stoker, B. White., E. Stackpole, T. Highley, and M. Humphrey. Toward realizable restricted delegation in computational grids. *Proceedings of the International Conference on High Performance Computing and Networking Europe (HPCN Europe 2001)*, June 2001.

[89] B. Sundaram and B. Chapman. Policy engine: A framework for authorization, accounting policy specification and evaluation in grids. In *Proceedings of the 2nd International Conference on Grid Computing*, Denver, Colorado, USA, Nov 2001.

[90] T. Sunsted. The practice of peer-to-peer computing: Introduction and history. *IBM developerWorks*, March 2001.

[91] T. Sunsted. The practice of peer-to-peer computing: Trust and security in p2p networks. *IBM developerWorks*, July 2001.

[92] M. Surridge. A rough guide to grid security v1.1. Technical report, National e-Science Centre, 2002.

[93] V. Talwar, S. Basu, and R. Kumar. Architecture and environment for enabling interactive grids. *Journal of Grid Computing*, 1, 2003.

[94] D. Thain, T. Tannenbaum, and M. Livny. *Condor and the Grid*. John Wiley and Sons Inc., 2002.

[95] A. Tripathi. Challenges designing next-generation middleware systems. *Communications of the ACM*, 45, 2002.

[96] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet x.509 public key infrastructure (pki) proxy certificate profile. Technical report, IETF, June 2004.

[97] A. Uszok, J. Bradshaw, R. Jeffers, M. Johnson, T. A., J. Dalton, and S. Aitken. Policy and contract management for semantic web services. In *Proceedings of the AAAI Spring Symposium on Semantic Web Services*, 2004.

[98] A. Vahdat, E. Belani, P. Eastham, and C. Yoshikawa. Webos: Operating system services for wide area applications. *Proceedings of the 7th Symp. on High Performance Distributed Computing*, July 1998.

[99] D. C. Verma, S. Sahu, S. B. Calo, M. Beigi, and I. Chang. A policy service for grid computing. In *Proceedings of the Third International Workshop on Grid Computing*, Baltimore, MD, USA, November 2002.

[100] G. von Laszewski, I. Foster, J. Gawor, P. Lane, N. Rehn, and M. Russell. Designing grid-based problem solving environments and portals. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, Hawaii, USA, 2001.

[101] N. N. Vuong, G. S. Smith, and Y. Deng. Managing security policies in a distributed environment using extensible markup language (xml). In *Proceedings of the 2001 ACM symposium on Applied computing*, Las Vegas, Nevada, USA, 2001.

[102] V. A. Vyssotsky, F. J. Corbató, and R. M. Graham. Structure of the multics supervisor. In *Proceedings of the AFIPS Conference*, 1965.

[103] G. Wasson and M. Humphrey. Policy and enforcement in virtual organizations. In *Proceedings of the 4th International Workshop on Grid Computing (Grid2003) (associated with Supercomputing 2003)*, Phoenix, USA, Nov 2003.

[104] D. Watkins and S. Lange. An overview of security in the .net framework. Technical report, Microsoft, January 2002.
[105] P. Watson. Databases and the grid. Technical Report CS-TR-755, Newcastle University, 2002.
[106] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, C. K., J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for grid services. *Proceedings of the 12th International Symposium on High Performance Distributed Computing (HPDC-12)*, June 2003.
[107] A. Wesley and al. Security in a web services world: A proposed architecture and roadmap. Technical report, IBM and Microsoft, April 2002.
[108] T. Y. C. WOO and S. S. Lam. Authorizations in distributed systems: A new approach. *Journal of Computer Security*, 2(2,3):107–136, 1993.
[109] W. A. Wulf, C. Wang, and D. Kienzle. A new model of security for distributed systems. Technical Report CS-95-34, University of Virginia, 1997.
[110] G. Zhang and M. Parashar. Dynamic context-aware access control for grid applications. In *Proceedings of the Fourth International Workshop on Grid Computing (GRID 03)*, 2003.
[111] L.-J. Zhang, H. Li, and H. Lam. Toward a business process grid for utility computing. *IT Professional*, 6, September/October 2004.
[112] P. Zimmermann. *The Official PGP Users Guide*. MIT Press Trade, 1995.

Glossary

A

ACL Access Control List.

C

CA Certification Authority.

CAS Community Authorization Service.

CPU Central Processing Unit.

CRL Certificate Revocation List.

D

DRBAC Dynamic Role Based Access Control.

E

EACL Extended Access Control List.

EDG European Data Grid.

G

GAA-API Generic Authorization and Access Control API.

GRAM Grid Resource Allocation and Management.

GSI Grid Security Infrastructure (no sistema Globus).

I

IETF Internet Engineering Task Force.

J

JXTA ”JXTA is short for Juxtapose, as in side by side. It is a recognition that peer to peer is juxtapose to client server or Web based computing – what is considered today’s traditional computing model.”retirado de www.jxta.org.

L

LOID Legion Object Identifier.

O

OASIS Organization for the Advancement of Structured Information Standards.

OGSA Open Grid Services Architecture.

OLA Online Agent (no sistema Crisis)Central Processing Unit.

P

PKCS Public Key Cryptography Standards.

PKI Public Key Infrastructure.

R

RBAC Role Based Access Control.

RSL Resource Specification Language.

S

SAML Security Assertion Markup Language.

V

VOMS Virtual Organization Membership Service.

W

WS Web Service.

X

XML Extended Markup Language.