Autorização e controle de acesso para o prontuário eletrônico do paciente em ambientes abertos e distribuídos: uma proposta de modelo e arquitetura

Gustavo H. M. B. Motta[1, 2, 3], Sérgio S. Furuie[1], Fabiane B. Nardon[1,2], Marco A. Gutierrez[1] e Marcos Yamaguti[1]

[1] Instituto do Coração – Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo
[2] Escola Politécnica da Universidade de São Paulo
[3] Departamento de Informática – Universidade Federal da Paraíba
e-mail: gustavo.motta@incor.usp.br

Resumo

A perspectiva do aumento do uso de Sistemas de Informações Hospitalares em aplicações distribuídas de telemedicina a curto e médio prazos, com o Prontuário Eletrônico do Paciente (PEP) disponível a um grande número de instituições e pessoas, suscita preocupações acerca do acesso às informações clínicas. O problema é que o controle de acesso ao PEP, em nenhuma circunstância, deve prejudicar o atendimento ao paciente por negar acesso legítimo às informações e aos serviços requisitados pelo pessoal médico. Outra questão refere-se a como administrar uma política de autorização e impor o controle de acesso ao PEP, visto que este é composto por segmentos que estão distribuídos em bases de dados distintas, acessadas por aplicações diversas, em plataformas heterogêneas. Este trabalho apresenta um modelo de autorização adequado para as exigências de controle de acesso ao prontuário eletrônico, capaz de assegurar a privacidade do paciente e a segurança de acesso aos seus dados, mas flexível o suficiente para conceder o acesso em situações excepcionais. Propõem ainda a implementação deste modelo de autorização e controle de acesso numa arquitetura baseada em padrões abertos e distribuída, capaz de ser acessada pelos diversos segmentos em que o PEP se distribui, mas com uma administração unificada para política de autorização e controle de acesso.

Palavras-chave: controle de acesso, arquitetura aberta e distribuída, prontuário eletrônico do paciente

1. Introdução

A concepção de modelos para autorização e controle de acesso ao prontuário eletrônico do paciente (PEP) vem despertando o interesse de pesquisadores da área nos últimos anos [3, 2 e 9], particularmente com a perspectiva do aumento do uso das informações clínicas presentes no PEP em aplicações distribuídas de telemedicina, a curto e médio prazos. Entretanto, duas questões principais desafiam sua concepção.

A primeira é que o controle de acesso ao PEP, em nenhuma circunstância, deve prejudicar o atendimento ao paciente por negar acesso legítimo às informações e aos serviços requisitados pelo pessoal médico. No entanto, fora deste contexto, as informações do prontuário devem permanecer sigilosas, exceto quando em atendimento à vontade do paciente ou a determinações legais. O problema é que não existe um modelo claro sobre a política de autorização e controle de acesso a ser adotada para o PEP, isto é, como determinar quem tem direito a acessar certas classes de informações, com quais privilégios e em quais condições. É indesejável impor um controle de acesso tão restrito que impeça um médico, em uma sala de emergência, acessar o prontuário de um paciente gravemente doente. Neste caso, a circunstância da emergência deve ser considerada uma exceção, sobrepondo-se a restrições de acesso estabelecidas [5]. Um modelo para autorização e controle de acesso ao PEP deve ser flexível o suficiente para suportar exceções, estabelecidas estática ou dinamicamente, levando em conta informações contextuais ou circunstanciais [4].

A segunda questão refere-se a como administrar uma política de autorização e impor o controle de acesso ao PEP, visto que este é composto por segmentos que estão distribuídos em bases de dados distintas, acessadas por aplicações diversas, em plataformas heterogêneas. É necessária a adoção de uma arquitetura aberta e distribuída capaz de suportar a administração da política de autorização e o controle de acesso de modo unificado e consistente, a partir de diferentes sistemas, em plataformas e linguagens de programação distintas, mas de forma padronizada.

Este trabalho apresenta um modelo de autorização e controle de acesso ao prontuário eletrônico do paciente para uso no Instituto do Coração (InCor) do Hospital das Clínicas da Faculdade de Medicina da USP e propõe sua implementação numa arquitetura aberta e distribuída, visando responder às questões postas anteriormente. O trabalho está organizado da seguinte forma: a seção 2 apresenta os modelos de autorização e controle de acesso para o PEP; a seção 3 descreve a arquitetura adotada na implementação dos modelos e, finalmente, a seção 4 traz as conclusões do trabalho.

2. Modelo de autorização e controle de acesso

Uma autorização estabelece as permissões (direitos ou privilégios) de acesso que um sujeito* pode exercer em um determinado recurso computacional. O controle de acesso vai limitar as ações ou operações que um usuário legítimo de um sistema de computação pode realizar [8], com base nas autorizações aplicáveis ao mesmo no momento do acesso. Em geral, o controle de acesso exige a autenticação prévia do usuário, visando estabelecer a sua identidade para o sistema de segurança, tipicamente através de identificação pessoal (nome para *login*) e

*Um sujeito pode ser um usuário humano ou um programa que atua em benefício deste.

senha ou com cartões de identificação, certificados digitais ou dados biométricos. As subseções seguintes descrevem o modelo de autorização e controle de acesso proposto para o PEP.

2.1. Autorizações

O modelo apresentado aproveita as idéias de autorizações positivas e negativas e o suporte a exceções estáticas através de autorizações fortes e fracas do modelo de autorização de acesso para sistemas gerenciadores de bancos de dados relacionais definido por Bertino et al. [1]. Entretanto, ao contrário deste, o modelo proposto neste trabalho baseia a política de autorização de acesso nos papéis que os usuários exercem na organização e não em grupos de usuários. Ademais, propõe a utilização de exceções dinâmicas influenciadas por fatores circunstanciais ou contextuais que acontecem no momento da solicitação da autorização de acesso. Por fim, o tipo de recurso a ser protegido não se restringe tabelas de banco de dados relacionais, podendo também ser objetos, métodos, programas, entre outros.

Uma autorização de acesso é formada por uma tupla `<p, r, tp, priv, ta>`, onde `p` é o papel para o qual o privilégio é estabelecido; `r` especifica o recurso para o qual o privilégio se aplica; `tp` especifica o tipo de privilégio, positivo (+) quando concedido e negativo (-) quando proibido; `priv` é o privilégio de acesso estabelecido e `ta` especifica se o tipo de autorização é forte ou fraca. Este modelo apresenta características que o torna adequado para o controle de acesso ao PEP, descritas nas subseções a seguir.

2.1.1. Hierarquia de papéis

Um papel é definido como o conjunto de ações e responsabilidades associados com uma determinada atividade de trabalho [8]. As autorizações de acesso e respectivos privilégios são atribuídos a cada papel dentro de uma instituição, de acordo com as ações e responsabilidades pertinentes. Estes papéis são organizados em hierarquias, de modo que privilégios comuns sejam agrupados em papéis mais gerais.

A Figura 1 (a) ilustra a hierarquia de papéis parcial definida para o InCor e a Figura 1 (c) mostra as autorizações atribuídas a cada papel. O papel *Residente* herda todas as autorizações de acesso definidas para os papéis *Médico* e *Usuário*, especificando apenas as suas próprias autorizações. Papéis podem ser vistos, portanto, como grupos de autorizações. Neste esquema, o acesso do usuário aos recursos (Figura 1 (b)) é concedido de acordo com os papéis que ele exerce na organização. Cada usuário tem atribuído para si um conjunto particular de papéis e, no momento da autenticação (*login*) ou durante a sessão em curso, ele escolhe apenas um deles para assumir, com os respectivos privilégios.

Diferente do enfoque baseado em grupos de usuários, adotado por Bertino et al. [1], Beznosov [2] recomenda o uso de papéis porque permitem que um mesmo usuário atue com diferentes conjuntos de privilégios em momentos distintos, além de facilitar a administração da política de autorização. Por restringir, em cada sessão, o potencial de acesso que um usuário pode exercer, reduz-se também a vulnerabilidade do acesso, pois a totalidade das autorizações nunca fica disponível simultaneamente, exceto quando o usuário só tem um papel associado. Esta é uma das vantagens de se adotar papéis em vez de grupos de usuários, pois neste caso, o usuário sempre possui, em qualquer sessão, a totalidade das autorizações definidas para os grupos a que pertence.

2.1.2. Acesso seletivo aos recursos

O prontuário eletrônico do paciente é segmentado e pode estar distribuído em sistemas situados em plataformas heterogêneas. Há casos em que o acesso a um segmento do PEP é feito diretamente numa tabela de banco de dados, ou via um procedimento (ou programa), ou através de páginas WEB. Logo, deve ser possível especificar diferentes autorizações de acesso para diferentes partes do prontuário.

O modelo de autorização proposto representa hierarquicamente os recursos do PEP, conforme ilustrado na Figura 1 (b). Para cada parte dele, pode-se definir uma ou mais autorizações, permitindo a seletividade do acesso. Cada segmento representa um tipo de recurso específico, podendo ser um procedimento, uma página na WEB, uma tabela, um arquivo, um objeto, um método, etc. De acordo com o tipo de recurso, diferentes formas de acesso são estabelecidas. Por exemplo, a Figura 1 (b) descreve parcialmente a estrutura de páginas na WEB que dá acesso ao PEP no InCor. Os recursos PEP, IP, DM, Exm e AL são do tipo página na WEB e os privilégios possíveis de acesso são *consulta* e *autoria*. O recurso EL é do tipo procedimento, com privilégio de acesso *execução*. O usuário que assuma o papel de *Assistente* tem a autorização para executar este procedimento (Figura 1 (c)) e portanto, pode emitir laudos.

2.1.3. Autorizações positivas e negativas

Autorizações positivas definem os acessos que são permitidos, enquanto autorizações negativas estabelecem aqueles acessos que são negados [1]. No caso em que o acesso é proibido para a maioria dos usuários, usa-se uma autorização negativa; em situação contrária, define-se uma autorização positiva. Por exemplo, o acesso ao PEP é negado na autorização especificada para o papel *Usuário* na Figura 1 (c), visto que, usuários comuns não podem consultar o prontuário. Por outro lado, um usuário com o perfil *Médico* tem esta autorização positiva, uma vez que, em sua maioria, os médicos têm o direito de acessar o prontuário. Esta capacidade do modelo descrito facilita a administração da política autorização.

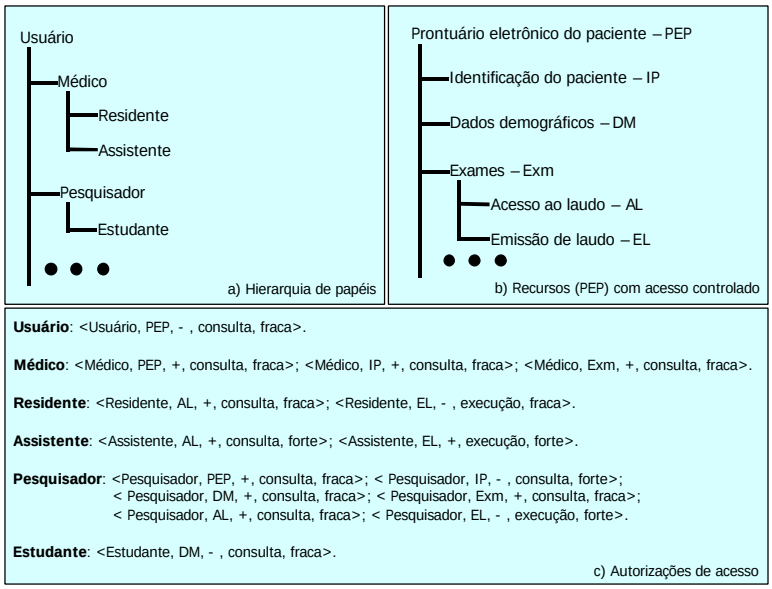

Figura 1 – Exemplo de uso do modelo de autorização de acesso ao PEP

2.1.4. Suporte controlado a exceções estáticas e dinâmicas

Uma exceção especifica que determinadas autorizações não são válidas em certas circunstâncias ou contextos[(1)]. Uma exceção é definida através de uma autorização criada de forma estática ou dinâmica.

Exceções estáticas

A hierarquia de papéis permite que o tipo de privilégio (+ ou -) de uma autorização definida para um papel seja modificada estaticamente num papel descendente deste. Por exemplo, o papel *Médico* excepcionalmente tem a autorização para acessar o PEP, pois modifica o tipo de privilégio da autorização que nega o acesso e que foi herdada do papel *Usuário*. Entretanto, qualquer outro papel definido abaixo de *Usuário* e que não especifique esta exceção, não terá acesso ao PEP.

Exceções dinâmicas

Uma exceção dinâmica possibilita a modificação do tipo de privilégio de uma autorização, mediante uma circunstância ou contexto existente no momento em que o usuário exerce um determinado papel numa sessão. Uma autorização pode ser modificada através de uma outra autorização, equivalente, gerada dinamicamente, mas com tipo de privilégio oposto. Autorizações oriundas de exceções dinâmicas sempre são do tipo fraca. Os fatores que influenciam o surgimento de exceções dinâmicas são descritos a seguir:

- **A localização do usuário**: de acordo com o local onde o usuário acessa uma informação, autorizações positivas ou negativas podem ser relaxadas por meio de uma exceção gerada dinamicamente. Por exemplo, um médico no papel de *Residente* não pode emitir laudos (Figura 1 (c)), mas esta autorização negativa pode ser relaxada pela criação dinâmica de uma autorização equivalente positiva, sempre que o acesso for efetuado em estações da sala de emergência ou do ambulatório;

- **Relacionamento usuário/paciente**: certos tipos de relacionamentos entre o usuário e o paciente não podem ser estabelecidos estaticamente, quando da definição dos papéis e seus privilégios. Estes relacionamentos só podem ser determinados dinamicamente, no momento em que uma autorização de acesso é solicitada. Por exemplo, não é possível saber *a priori* quais são os clientes de um plano de saúde, a fim de liberar, para o médico auditor, o acesso apenas aos PEPs dos pacientes do plano. Por variar com o tempo, este relacionamento deve ser verificado no momento do acesso, para que uma autorização apropriada seja criada;

- **Data e hora do acesso**: o momento do acesso é um fator importante para negar ou conceder uma autorização. Para um médico exercendo o papel de *Residente*, o turno de trabalho pode determinar janelas de tempo em que o acesso ao prontuário é permitido.

O modelo apresentado permite que exceções dinâmicas sejam criadas, mas não prescreve como serão especificadas e geradas, exceto que devem ser fracas. Esta prescrição será realizada em trabalhos futuros. Sem exceções, a execução de uma política de autorização para o PEP é de difícil implementação. Ou se teria uma definição de privilégios permissiva, comprometendo a privacidade do paciente, ou ela seria por demais rígida, comprometendo o acesso às informações essenciais para um atendimento adequado.

Controle às exceções

A fim de controlar o uso de exceções, para cada autorização concedida, positiva ou negativa, deve-se especificar se ela admite exceções ou não. As autorizações fracas admitem exceções, ao passo que, as fortes, não as admitem. Esta capacidade permite que se estabeleçam, desde as políticas mais severas, até as mais permissivas, com a variação entre os dois extremos controlada pela especificação de autorizações fortes e fracas.

Por exemplo, a autorização negativa para acesso ao PEP do papel *Usuário* admite exceções, pois é do tipo *fraca*. Assim, papéis que herdam de usuário podem redefinir o tipo de privilégio, como nos casos das autorizações dos papéis *Médico* e *Usuário* ilustrados na Figura 1 (c). Por outro lado, o papel *Pesquisador* tem uma autorização negativa do tipo *forte* para emissão de laudos. Neste caso, esta autorização, por ser forte, não pode ter seu tipo de privilégio modificado, seja em algum papel descendente de *Pesquisador*, seja através de uma autorização gerada dinamicamente.

Resolução de conflitos

Duas autorizações $<p_1, r_1, tp_1, priv_1, ta_1>$ e $<p_2, r_2, tp_2, priv_2, ta_2>$ conflitam se e somente se p_1 é ancestral de p_2 ou p_1 é descendente de p_2 ou $p_1 = p_2$ e $r_1 = r_2$ e $tp_1 \neq tp_2$ e $priv_1 = priv_2$ e $ta_1 = ta_2$. Por exemplo, se as autorizações <Médico, EL, - , execução, forte> ou <Usuário, EL, - , execução, forte> fossem definidas, elas conflitariam com a autorização <Assistente, EL, +, execução, forte>. Entretanto, quando definida para o papel *Pesquisador*, ela não conflita, pois não há nenhuma autorização definida em *Pesquisador* ou em qualquer ascendente ou descendente seu que seja forte, com o mesmo privilégio e especificada para o mesmo recurso computacional e que tenha tipo de privilégio oposto, isto é, positivo. Logo, as autorizações <Assistente, EL, +, execução, forte> e <Pesquisador, EL, - , execução, forte> não conflitam. Neste modelo, não se admitem autorizações conflitantes do tipo forte, como no exemplo anterior.

O conflito entre autorizações do tipo fraca é admitido e a política de resolução dá-se da seguinte maneira e ordem: (a) uma autorização fraca negativa, especificada num papel, prevalece sobre uma autorização fraca positiva conflitante, especificada para o mesmo papel; (b) uma autorização fraca, negativa ou positiva, especificada num papel, prevalece sobre qualquer autorização fraca conflitante especificada em papéis ascendentes ou descendentes deste. Por exemplo, as autorizações <Médico, PEP, +, consulta, fraca> e <Usuário, PEP, - , consulta, fraca> conflitam, porém para um usuário assumindo o papel de médico, a primeira autorização tem primazia sobre a segunda, definida para o papel usuário. De outro modo, quando duas autorizações fracas conflitam em papéis diferentes, numa mesma linha hierárquica, tem a prioridade aquela autorização do papel que o usuário está assumindo correntemente.

2.2. Controle de acesso

É responsabilidade do controle de acesso implementar um mecanismo que possibilite o acesso aos recursos computacionais estritamente de acordo com a política de autorização especificada. São pré-condições para o seu adequado funcionamento que o usuário que solicita acesso tenha sido corretamente autenticado e que tenha assumido apenas um dos papéis, dentre aqueles disponíveis para ele. Uma solicitação de acesso para um recurso computacional rc com privilégio de acesso priv é concedido ou negado de acordo com o seguinte algoritmo:

1. Para toda autorização especificada para o papel assumido, verifica-se se existe uma autorização forte, positiva ou negativa, com privilégio de acesso priv e recurso computacional rc . Caso exista uma com tipo de privilégio positivo, o acesso é concedido e o algoritmo termina. Caso exista uma com tipo de privilégio negativo, o acesso é negado e o algoritmo termina. Não havendo nenhuma das duas situações anteriores, repete-se recursivamente o processo para as autorizações do papel imediatamente superior na hierarquia, até o papel raiz. Autorizações fortes conflitantes não são admitidas;
2. Caso o algoritmo não termine, então não há uma autorização forte, positiva ou negativa, capaz de conceder ou negar o acesso ao recurso rc com privilégio de acesso priv. Neste caso, o algoritmo verifica primeiro se há alguma autorização gerada dinamicamente, aplicável para o papel, que negue o acesso. Caso exista, o acesso é negado e o algoritmo termina. Caso contrário, o algoritmo verifica em seguida se há alguma autorização gerada dinamicamente, aplicável para o papel, que autorize o acesso. Caso exista, o acesso é concedido e o algoritmo termina;
3. Caso não termine no passo anterior, o algoritmo então verifica se há alguma autorização estática fraca, aplicável para o papel, que autorize ou negue o acesso de acordo com o seguinte procedimento: (a) para toda autorização especificada para o papel assumido, verifica-se se existe uma autorização fraca, negativa, com privilégio de acesso priv e recurso computacional rc . Em caso afirmativo, o acesso é negado e o algoritmo termina; (b) caso o algoritmo não termine, para toda autorização especificada para o papel assumido, verifica-se se existe uma autorização fraca, positiva, com privilégio de acesso priv e recurso computacional rc. Em caso afirmativo, o acesso é concedido e o algoritmo termina. Não ocorrendo nenhuma das duas situações anteriores, repete-se recursivamente o processo para as autorizações estáticas fracas do papel imediatamente superior na hierarquia, até o papel raiz;

4. Caso o algoritmo não termine, então não há autorização aplicável para o papel que negue ou conceda o acesso ao recurso computacional rc com privilégio de acesso priv. Neste caso o acesso é negado e o algoritmo pára.

De acordo com o algoritmo especificado, uma autorização forte, com tipo de privilégio positivo ou negativo, tem prioridade sobre qualquer autorização fraca, inclusive as dinâmicas. Observa-se que não há necessidade de analisar todas as autorizações, com o algoritmo terminando na primeira autorização que se aplique a solicitação de acesso. A resolução de conflitos para autorizações fracas é naturalmente obedecida no algoritmo. Ademais, sempre que há conflito num mesmo papel, prevalece a autorização que nega o acesso.

3. Descrição da arquitetura

A arquitetura proposta para suportar o modelo de autorização e controle de acesso ao PEP, visto na seção 2, baseia-se num modelo cliente-servidor com três camadas (Figura 2). Compõe-se de um servidor de dados, responsável pelo armazenamento das autorizações, papéis, representações dos recursos protegidos e usuários; de um servidor de controle de acesso e autenticação, com a incumbência de implementar o mecanismo de controle de acesso e atender às solicitações de autenticação de usuários; e de aplicações clientes no terceiro nível, que solicitam autorizações de acesso, invocam funções de autenticação e de administração de autorizações, através de uma API (*Application Programming Interface*) padronizada.

Como o PEP compõe-se de aplicações heterogêneas, é preciso que as autorizações de acesso e de autenticação de usuários sejam solicitadas independentes de plataforma e linguagem de programação. A solução foi adotar o serviço de decisão para acesso a recursos (*Resource Access Decision Facility: RAD – Facility*) [3, 7], do CORBA *horizontal facilities* para suportar a implementação do mecanismo de controle de acesso descrito na subseção 2.2. O *RAD – Facility* oferece interfaces padronizadas que permitem o controle de acesso detalhado, ao nível da aplicação, mas de uma forma em que a lógica do controle de acesso é separada da lógica da aplicação, com transparência em relação ao mecanismo de decisão efetivamente implementado. Este *framework* prevê o tratamento dos fatores dinâmicos que influenciam a lógica de autorização e possibilita a combinação de diferentes políticas de controle de acesso. Oferece ainda interfaces padronizadas para a administração destas políticas de controle de acesso. A autenticação do usuário é realizada através da interface padrão para autenticação especificada no CORBA *Security Service* [6]. O controle de sessão dos usuários conectados é também responsabilidade deste servidor.

As autorizações, papéis, representação dos recursos protegidos e usuários são armazenados em um serviço de diretórios hierarquizado, cujo acesso e esquemas de descrição de dados são padronizados através do protocolo LDAP (*Lightweight Directory Access Protocol*) [10], definido pelo IETF (*Internet Engineering Task Force*). Por ser hierárquico e flexível, o LDAP é capaz de representar naturalmente as hierarquias de papéis e de recursos do modelo de autorização proposto. Esquemas de dados padronizados já existentes para o LDAP são usados no armazenamento de informações sobre usuários (*login*, nome, senha, e-mail, etc.), papéis, (nome, descrição, membros, etc.) e recursos (nome, descrição, localização, etc.). Embora conte com atributos predefinidos, o serviço de diretórios LDAP permite a definição de novos atributos, conforme o modelo que se deseje adotar. Como os atributos preexistentes não eram suficientes para representar o modelo de autorização proposto, novos esquemas foram criados com todos os atributos necessários para representá-lo.

Para assegurar a funcionalidade do serviço de diretórios de maneira ininterrupta, um mecanismo de réplica automática foi implantado, mas totalmente transparente para o servidor de controle de acesso e autenticação. Deste modo, as alterações nas informações de usuários, papéis, recursos e autorizações sempre são efetuadas nos dois servidores LDAP (Figura 2).

Um protótipo do servidor de controle de acesso e autenticação foi implementado em Java a fim de suportar as interfaces do CORBA *Security Service* e do *RAD – Facility* para autenticação e autorização de acesso, respectivamente. Estas interfaces estão disponíveis através do servidor CORBA/Visibroker. Este protótipo permite autenticação simples de usuário, baseada numa identificação única e senha e o controle de acesso é efetuado de acordo com o algoritmo descrito na subseção 2.2, mas sem autorizações dinâmicas. O *Netscape Directory Server*, que implementa o protocolo LDAP, está sendo usado para manter o cadastro de usuários, papéis recursos e autorizações de acesso. O acesso ao servidor LDAP, a partir do servidor de controle de acesso, é feito através da API JNDI (*Java Naming Directory Interfaces*), padrão para acesso a serviços de diretório em Java. A JNDI é usada para solicitar a autenticação do usuário no servidor LDAP e para efetuar as consultas necessárias para implementação do algoritmo de controle de acesso. Também foi desenvolvido em Java um protótipo para administração da política de autorização, que permite o cadastramento de usuários, papéis, recursos e autorizações.

No momento, esta solução está sendo usada em caráter experimental para autenticar usuários e controlar o acesso para o prontuário eletrônico do paciente disponível na intranet do InCor. O protótipo de administração também usa esta solução para o seu próprio controle de acesso, sendo seu uso permitido para os usuários privilegiados que possuem o papel de *Administrador*.

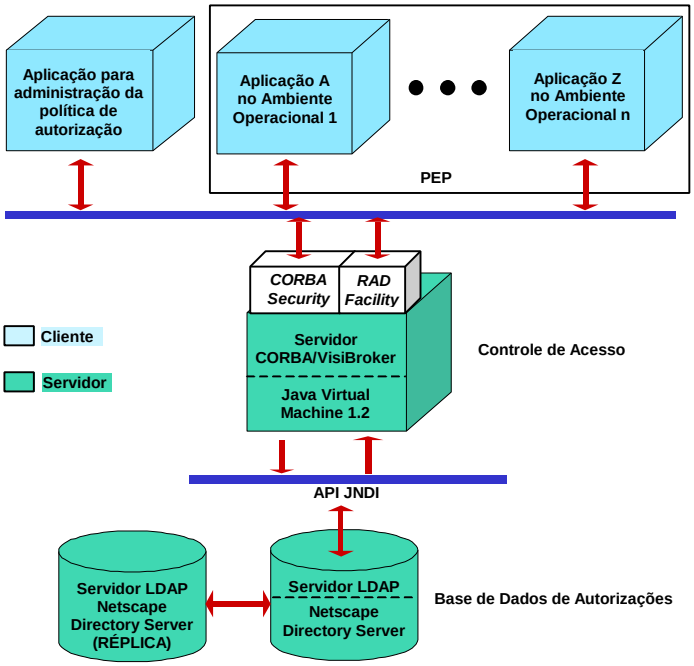

Figura 2 – Modelo da arquitetura para controle de acesso e autenticação de usuário

4. Conclusão

Este trabalho apresentou um modelo de autorização adequado para as exigências de controle de acesso ao prontuário eletrônico, buscando assegurar a privacidade do paciente e a segurança de acesso aos seus dados, mas flexível o suficiente para conceder o acesso em situações excepcionais. Propôs ainda a implementação deste modelo de autorização e controle de acesso numa arquitetura baseada em padrões abertos e distribuída, capaz de ser acessada pelos diversos segmentos em que o PEP se distribui, mas com uma administração unificada para política de autorização e controle de acesso. Está em andamento uma definição mais abrangente dos papéis de usuários do InCor e a definição das autorizações para acesso a outros sistemas que compõem o PEP. A especificação e implementação das exceções dinâmicas e de ferramentas para sua definição e administração serão realizadas em trabalhos futuros.

5. Referências

1. Bertino, E.; Jajordia, S. e Samarati, P. "A Flexible Authorization Mechanism for Relational Data Management Systems", *ACM Transactions on Information Systems* 17, 2 (Abril 1999), 101-140.
2. Beznosov, K. "Requirements for Access Control: US Health-care Domain", *Proceedings of the 3rd ACM Workshop on Role-based Access*, Fairfax, VA, USA, (1998), 43.
3. Beznosov, K.; Deng, Y.; Blakley, B.; Burt, C. e Barkley, J. "A Resource Access Decision Service for CORBA-based Distributed Systems", *Proceedings. of the 15th Annual Computer Security Applications Conference (ACSAC '99)*., (1999), 310-319.
4. Motta, G. H. M. B.; Furuie, S. S.; Nardon, F. B. e Gutierrez, M. A. "Considerações Sobre o Controle de Acesso ao Prontuário Eletrônico do Paciente", *Anais do CBIS´2000 – VII Congresso Brasileiro de Informática em Saúde*, São Paulo, SP, outubro de 2000.
5. National Academy of Sciences. *For the Record: Protecting Electronic Health Information*, National Academy Press, Washington, DC, USA, 1997.
6. Object Management Group. *CORBA Security Service Specification*. In: http://www.omg.org/cgi-bin/doc?formal/98-12-17.
7. Object Management Group. *Resource Access Decision Facility*. In: http://www.omg.org/cgi-bin/doc?dtc/00-08-06.
8. Sandhu, R. S. e Samarati, P. "Access Control: Principles and Practice", *IEEE Communications Magazine*, (Setembro 1994), 40-48.
9. Wiederhold, G.; Bilello, M.; Sarathy, V. e Qian, X "A Security Mediator for Health Care Information", *Proceedings of the 1996 AMIM Conference*, Washington, DC, USA, (1996), 120-124.
10. Yeong, W.; Howes, T. e Kille, S. *Lightweight Directory Access Protocol*. Internet Engineering Task Force – IETF, (Março 1995), In: http://www.ietf.org/rfc/rfc1777.txt?number=1777.