Relatório Final
MAC 5701 - Tópicos em Ciência da Computação *

José de Ribamar Braga Pinheiro Júnior (N.o USP: 4879050)
Orientador - Prof. Fábio Kon
IME - Instituto de Matemática e Estatística USP - Universidade de São Paulo

22 de junho de 2004

Sumário

1 Introdução **2**

2 Motivações **2**

3 InteGrade **2**

4 Objetivos do trabalho. **5**

5 O projeto **5**
5.1 A nova arquitetura 6
5.2 A implementação 7

6 Conclusões **8**

*Código fonte disponível em: http://www.ime.usp.br/ jrbraga/integrade-sec/.

1 Introdução

Este documento descreve as atividades de pesquisas realizadas para a disciplina MAC 5701 - Tópicos em Ciência em Computação. Apresenta-se aqui as etapas executadas durante a vigência do curso, suas motivações e seus objetivos.

2 Motivações

Com a popularização da informática no mundo, duas ocorrências contraditórias nos parecem bem claras: a definição crescente de problemas que demandam muito processamento e o aumento do parque computacional. Atualmente, computadores encontrados comumente nos ambientes freqüentados por pessoas comuns, possuem a capacidade de processamento dos supercomputadores de uma década atrás. A possibilidade de permitir que estas máquinas, ociosas na maior parte do tempo, disponibilizem seus recursos computacionais para solucionar os problemas de outros usuários em uma rede de computadores, fez surgir a idéia da criação de uma infra-estrutura que os interconecte através de uma rede, denominada Computação em Grade (*Computacional Grid*).

As redes de computadores, em especial a internet, no entanto, possuem um legado fatal: foram criadas sem a preocupação com a segurança dos dados por elas transmitidos [GS96]. Naquela época as relações de segurança entre as partes envolvidas, que eram confiáveis e conhecidas, não obtiveram a preocupação necessária dos seus projetistas. A partir do momento que a arquitetura da internet (TCP/IP), mais recentemente , firmou-se como o padrão de fato das redes de computadores, estas herdaram o problema e técnicas adicionais tiveram que ser usadas para manter a segurança das informações.

Os computadores que participam das grades, também estão vulneráveis a problemas com a insegurança. Os seus sistemas operacionais podem não ser seguros o bastante, permitindo o acesso a recursos indevidos, ou códigos maliciosos podem ser executados de forma a se beneficiar de alguma falha na máquina para obter estes recursos [1]. Daí um problema grave nos sistemas de computação em grade: como garantir que as aplicações submetidas não causem prejuízos aos seus participantes. A solução seria implementar técnicas de segurança da informação para minimizar os possíveis problemas.

A segurança da informação possui quatro objetivos fundamentais: confidenciabilidade, integridade, disponibilidade e uso legítimo [For94]. A confidenciabilidade garante que as informações serão protegidas de entidades não autorizadas. A seguinte, a integridade, objetiva a consistência dos dados impedindo sua modificação de forma intencional ou não. A disponibilidade, garante que usuários legítimos terão o acesso aos recursos a que tem direito. E, finalmente, o uso legítimo determina que os recursos somente serão disponíveis a pessoas autorizadas.

3 InteGrade

InteGrade [Kon04] é um projeto de computação em grade sendo desenvolvido pelo IME-USP (Instituto de Matemática e Estatística da Universidade de São Paulo). Implementado utilizando a tecnologia CORBA [Gro03], um padrão da industria para sistemas

[1] Técnicas bastante comuns, como o estouro de buffer *buffer overflow* [GS96], podem ser utilizadas para obter acesso irrestrito aos recursos das máquinas.

de objetos distribuídos, o InteGrade permite que aplicações paralelas sejam executadas em ambiente distribuído. Sua diferenciação dentre os demais sistemas de computação em grade, resume-se ao fato de ter sua arquitetura projetada de forma a diminuir os impactos ao usuário que disponibilizam seus recursos a grade. Seja porque é definido sobre uma arquitetura de objetos distribuídos padrão aberta, reutiliza a base computacional já instalada ou ainda teve como premissa de projeto da sua arquitetura, a prioridade aos processos do usuário nativo, minimizando os impactos a quem se disponibiliza a usá-la.

A arquitetura definida para o InteGrade [Kon04] pode ser vista na figura 1. O Nó Dedicado representa um recurso computacional dedicado a grade. O Nó Compartilhado é utilizado pelo usuário e provê conjuntamente seus recursos computacionais à grade. O Nó Usuário submete aplicações para execução, assim como pode disponibilizar também seus recursos à grade. E, finalmente, o Gerenciador do Aglomerado é responsável pelo monitoramento dos demais nós, escalonamento de aplicações, análise de uso dos nós no aglomerado e coleta de informações.

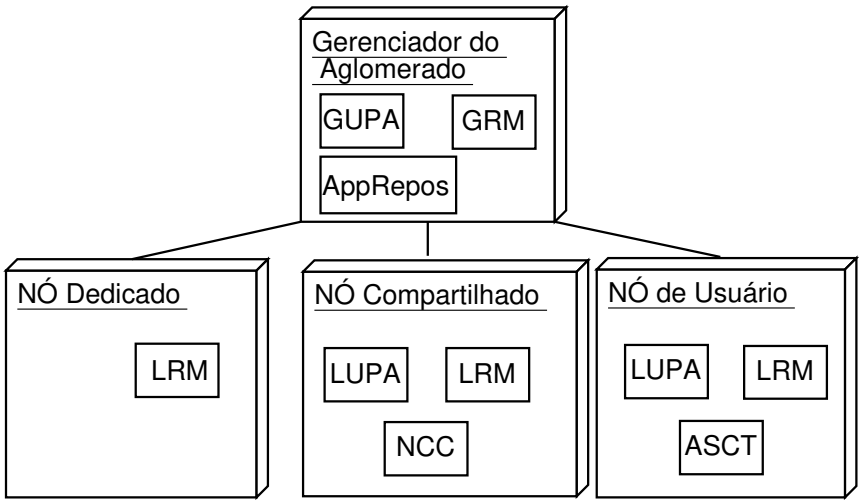

Figura 1: Arquitetura do Intra-Aglomerado do InteGrade.

Os módulos GRM (Global Resource Manager) e LRM (Local Resource Manager colaboram para o gerenciamento dos recursos do aglomerado. O LRM é executado em cada nó, coletando e atualizando periodicamente informações pertinentes (uso de CPU, memória, etc) entregando-as ao GRM. De posse destas informações, o GRM pode escalonar aplicações a serem executadas em um determinado nó, de acordo com suas necessidades e restrições.

O ASCT permite aos usuários submeterem as suas aplicações a grade. Através dele os usuários podem especificar pré-requisitos definidos para a aplicação (plataforma, memória necessária, etc) e controlar a execução dos processos.

O módulo AppRepos é o repositório de aplicações. Ele armazena em memória não volátil as aplicações submetidas à grade e as disponibiliza aos nós que irão executá-las, assim que um escalonamento ocorrer. A seqüencia que isto ocorre é a seguinte: através do ASCT o usuário registra uma aplicação no AppRepos, em seguida um escalonamento

é feito pelo GRM indicando ao nó escolhido (através do LRM) que ele deve executar uma aplicação, o LRM recupera o binário da aplicação e a executa.

Os módulos LUPA (Local Usage Pattern Analyzer) e GUPA (Global Usage Pattern Analyzer), possuem a função de análise e coleta de padrões de uso dos nós. O LUPA é executado em cada nó do aglomerado e compila as informações do padrão de uso da máquina entregando-as ao GUPA. O GUPA disponibiliza as informações ao GRM que pode usá-las para tomar decisões sobre o escalonamento das aplicações a ele submetido.

Finalmente, o módulo NCC permite aos donos dos recursos ajustar as condições para o compartilhamento dos recursos. O período que os recursos serão disponibilizados, a quantidade de memória e CPU disponível, são exemplos de parâmetros que podem ser definidos.

Atualmente o InteGrade encontra-se numa fase intermediária de desenvolvimento. O sistema já é executado em caráter experimental nos laboratórios do IME [2], permitindo que usuários cadastrados possam submeter suas aplicações. Uma equipe de pesquisadores e desenvolvedores continua trabalhando no projeto, incluindo professores deste instituto e de outras universidades.

As definições das políticas de segurança e sua implementação foram definidas dentro do cronograma do projeto InteGrade para a presente fase. Como utiliza a infra-estrutura de comunicação da internet, o integrade recebe o legado de todas as ameaças presentes naquela arquitetura. Uma ameaça é qualquer evento ou circunstância que permita comprometer a segurança de dados de um sistema [TS02]. Existem quatro tipos de ameaças que foram consideradas, a saber:

- **Interceptação** - refere-se ao acesso não autorizado a um determinado recurso ou serviço. Durante este processo, os dados poderiam, por exemplo, ser copiados pela escuta de um meio de comunicação entre dois processos ou pelo acesso indevido a um sistema de arquivos. No InteGrade a interceptação ocorre durante a troca de mensagens entre os módulos e entre as aplicações executadas na grade.

- **Interrupção** - ocorre quando um dado ou um serviço é colocado indisponível. Ataques deste tipo são comumente denominados *Denial Of service* (negação de serviço). A interrupção ainda é um problema muito difícil de resolver [For94]. Se, por exemplo, um usuário (ou um processo) indevido, inundar os módulos de requisições, estes exaurirão os recursos dos nós, levando-os a falhar. Aplicações indevidas poderiam, da mesma forma, ser executadas nas máquinas com intuito explícito de tornar os nós indisponíveis.

- **Modificação** - consiste na alteração não autorizada dos dados. Esta vulnerabilidade poderia ser realizada pela interceptação, modificação e retransmissão dos dados em uma rede de computadores, por exemplo. Uma aplicação ao ser submetida através do ASCT é armazenada, temporariamente, no gerenciador do aglomerado. Falhas poderiam ocorrer durante a transferência dos dados que representam as aplicações (arquivo binário) ou até mesmo com um ataque direto ao sistema de arquivo do gerenciador.

- **Fabricação** - realizada através da criação de atividades e dados que não existam. Um intruso poderia no InteGrade, por exemplo, inserir entradas no GRM que representariam, erroneamente, uma aplicação registrada ou até mesmo um nó de usuário.

[2] Laboratórios do LCPD e Eclipse do IME-USP.

Em função destas ameaças, considerou-se preliminarmente as seguintes políticas de segurança:

- A comunicação entre os processos dos nós participantes do aglomerado e o seu gerenciador, deverá permitir confidencialidade e integridade dos dados;
- Caberá somente ao usuário do sistema, garantir a segurança da aplicação submetida à grade;
- Um determinado usuário e/ou sua aplicação deverá ser identificado de maneira inequívoca e assinada na grade;
- Um determinado nó poderá definir o nível de acesso que será obtido pelos processos a ele submetidos ou aos recursos nas máquinas hospedeiras (nó ou gerenciador);
- Os eventos gerados pelos sistemas de segurança serão divididos em níveis e poderão, se assim desejado, ser registrados.

4 Objetivos do trabalho.

Diante do contexto delineado na seção anterior , notamos que o InteGrade necessita de uma implementação de segurança que proteja os seus dados das ameaças listadas, segundo as políticas definidas. Sabemos, no entanto, que implementar todas essas políticas carece de tempo e uma árdua pesquisa. Durante este trabalho consideramos somente a implementação do terceiro ítem das políticas listadas.

Objetivou-se neste trabalho, uma implementação de segurança para o Integrade de forma a permitir que usuários e aplicações sejam identificados de maneira inquestionável. Tal definição impedirá, ou tornará muito difícil, que usuários mal intencionados desconhecidos possam submeter aplicações indevidas a grade, causando prejuízos aos seus usuários.

5 O projeto

A solução proposta para a política de segurança que permite que usuários e aplicações sejam identificados de forma inequívoca na grade, foi implementada utilizando-se o GSS [Lin00] . O GSS define um conjunto de APIs que permite ao programador transparência sobre os mecanismos de segurança que os implementa possibilitando uma maior portabilidade dos sistemas de segurança. O GSS é um framework que provê serviços que suporta diversos mecanismos e tecnologias, como o kerberos ou chaves privadas/públicas [Ter00].

O GSS disponibiliza, basicamente, duas coisas: contexto e serviços de segurança. O contexto de segurança pode ser tratado como um estado de confiança entre aplicações. Os serviços de segurança disponibilizados pelo GSS são a integridade e a confidencialidade. Enquanto o primeiro garante que os dados não sejam modificados, o último certifica que eles sejam interceptados.

5.1 A nova arquitetura

Definimos dois novos módulos para serem acrescidos a arquitetura do Integrade: LSM (Local Security Manager) e GSM (Global Security Manager) visto na figura 2. O LSM e o GSM utilizam o GSS para implementar um contexto de segurança entre os seus módulos. O GSS, por sua vez, implementa seus serviços através do Kerberos.

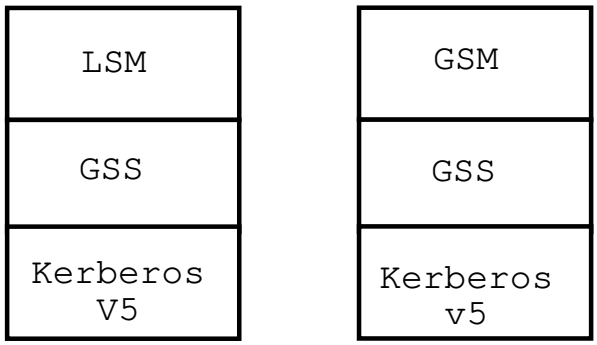

Figura 2: Novos módulos do InteGrade.

A figura 3 nos dá a visão da localização dos novos módulos dentro da arquitetura. O GSM, localizado no Gerenciador do Aglomerado, é responsável pelo gerenciamento dos contextos de segurança entres os diversos módulos. O LSM, presente nos módulos AppRepos, AsctGui [3] e LRM, autentica cada módulo e cria contextos de segurança com o GSM. A relação de confiança entre os LSMs e o GSM é utilizado para garantir a assinatura na troca de mensagens entre os módulos.

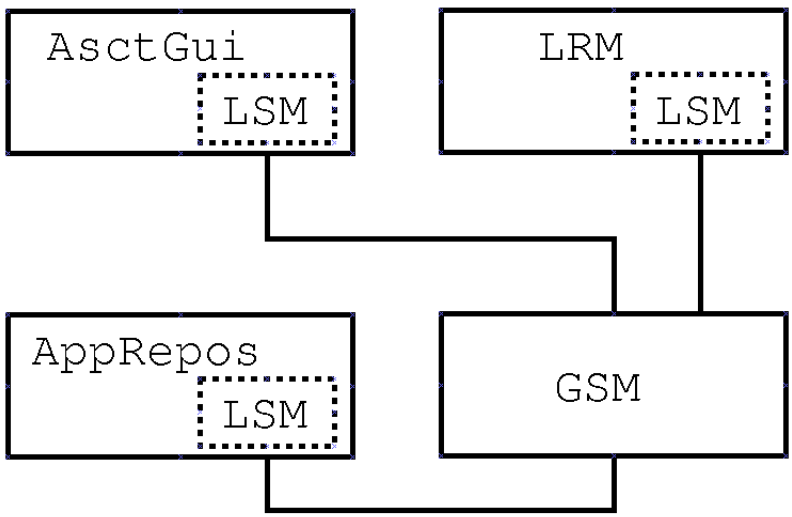

Figura 3: Localização dos novos módulos.

[3] O AsctGui é a versão gráfica do ASCT.

5.2 A implementação

Como todo o Integrade, o projeto proposto neste trabalho foi implementado utilizando-se o paradigma de programação orientado a objetos. O sistema foi desenvolvido utilizando as linguagens orientadas a objetos Java [Mic04] e C++ [Str00]. Para descrever um caso de uso do sistema utilizar-se-á a linguagem UML [BRJ00] (Unified Modeling Language). A linguagem UML é uma linguagem-padrão para a elaboração de projetos de software. Consideramos, a partir daqui, que estas linguagens já são conhecidas.

A figura 4 mostra o diagrama de classes que implementa os módulos LSM e GSM. AS classes GsmImpl e ContextManager implementam o módulo GSM e o LsmImpl o módulo LSM. A classe Security é uma classe utilitária que é compartilhada ente as demais.

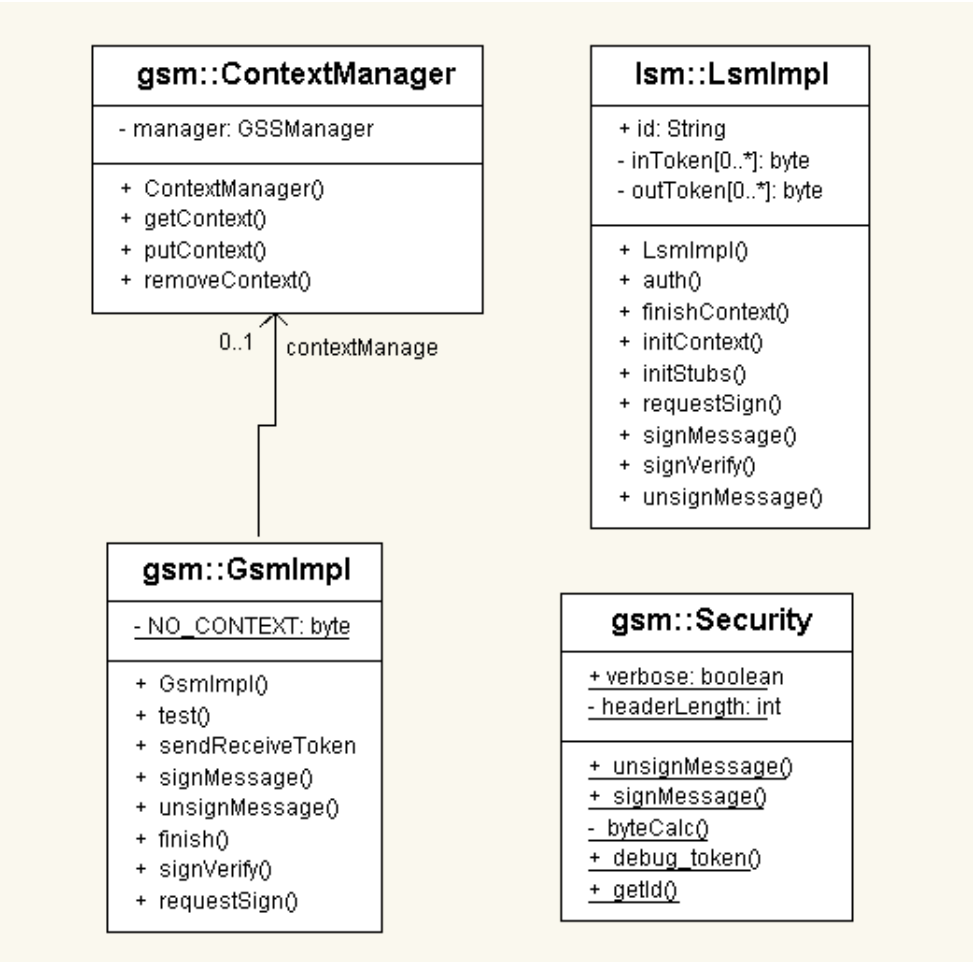

Figura 4: Diagrama de classes.

A figura 5 mostra um diagrama de sequência que representa o caso de uso de uma aplicação sendo registrada e logo após sendo executada. Inicialmente todos os módulos se autenticam e iniciam seus contextos (1 e 2). Antes de registrar uma aplicação, o ASCT assina o arquivo executável (3) que é verificado e reassinado pelo AppRepos assim que é recebido (4). O ASCT solicita ao GRM execução da aplicação passando o seu identificador e este indica ao LRM que foi escalonado para execução (5). O LRM solicita o binário, assinando o identificador da aplicação (6), que é conferido pelo AppRepos . O

AppRepos, por sua vez, assina o arquivo binário relativo a aplicação com o contexto do LRM e o entrega (7). O LRM finalmente executa a aplicação (8).

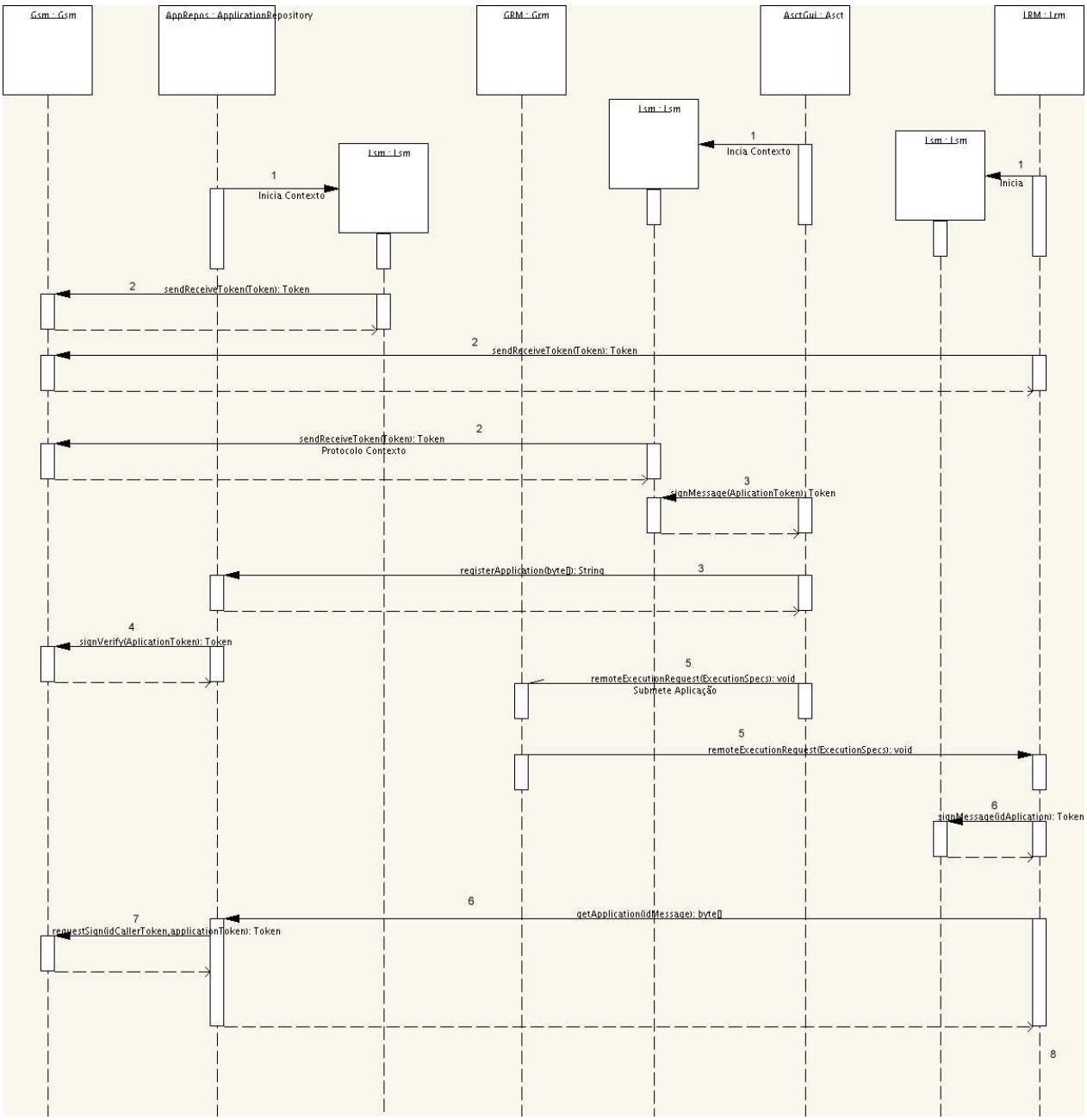

Figura 5: Diagrama de sequencia.

6 Conclusões

O sistema desenvolvido cumpre os requisitos iniciais definidos neste documento, ou seja permitir a assinatura das aplicações por usuários conhecidos. Para fazê-lo, a implementação proposta utilizou GSS, paradigma de orientação a objetos e padrões de projetos com a fi-

nalidade de desenvolver e apresentar os resultados obtidos , de uma forma expressiva, flexível e portável. A utilização de padrões de projetos, resolveu problemas comuns de computação com soluções já bem conhecidas e as apresenta como uma linguagem de alto nível. Ao escolher as linguagens Java e C++, o projeto tornou-se multiplataforma podendo ser compilado e executado nas diversas arquiteturas suportadas por estas linguagens. E ademais, o GSS permite a utilização de tecnologias de segurança de forma transparente. O resultado final foi uma implementação, que através de uma interface simples, permite a assinatura de aplicações em um sistema de computacional genérico.

O sistema no entanto possui algumas limitações. Uma das mais fortes deve-se a forma com que o GSS/Kerberos implementa as autenticações junto ao sistema. As autenticações do Kerberos são disponibilizadas através da criação de arquivos denominados *tickets* que são utilizados pelos processos para identificação de seus participantes. Se não for tomado o devido cuidado, ou se o sistema operacional não possuir políticas fortes de acesso ao sistema de arquivos, estes podem ser interceptados e utilizados para autenticação indevida. Além disto, por não ser objetivo deste trabalho, a assinatura de aplicações não resolve alguns problemas citados na seção 3, como por exemplo a interceptação dos dados.

Finalmente, sugere-se um refinamento maior na implementação de forma a abranger todas as políticas de segurança definidas. Sugere-se ainda, para trabalhos futuros, a implementação do sistema de segurança de uma forma hierarquizada de maneira a permitir que grades interligadas compartilhem os seus contextos de segurança, assim como proposto na arquitetura do InteGrade.

Referências

[BRJ00] Grady Booch, James Rumbaugh, and Ivar Jacobson. *UML Guia do Usuário*. Editora Campus, 2000.

[For94] W. Ford. *Computer Communications Security - Principles, Standard Protocols and Techniques*. Prentice-Hall, 1994.

[Gro03] Object Management Group. CORBA Security Service. http://www.omg.org/, 10 2003.

[GS96] Simson Garfinkel and Gene Spafford. *Practical UNIX & Internet Security*. OŔeilly & Associates, Inc., 1996.

[Kon04] Fabio Kon. Integrade. http://gsd.ime.usp.br/Integrade, Abril 2004.

[Lin00] J. Linn. Generic Security Service Application Program Interface. http://www.ietf.org/rfc/rfc2743.txt, 01 2000.

[Mic04] Sun MicroSystems. The Java Tutorial. http://java.sun.com/docs/tutorial, 2004.

[Str00] Bjarne Stroustrup. *A Linguagem de Programação C++*. Bookman Cia. Ed., 2000.

[Ter00] Routo Terada. *Segurança de Dados. Criptografia em Redes de Computadores*. Editora Edgard Blücher Ltda, 2000.

[TS02] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems. Principles and Paradigms*. Prentince Hall, 2002.