# Segurança de Código Móvel no Ambiente $\mu$CODE

**Leonardo Souza Silva**[1] **, Márcio Eduardo Delamaro**[2] **, Rodrigo Fraxino de Araújo**[2]

[1]Universidade Católica Dom Bosco - UCDB
Avenida Tamandaré, 6000 - Jd. Seminário - Caixa Postal 100
79117-900 Campo Grande, MS

[2]Centro Universitário Eurípides de Marília - UNIVEM
Avenida Hygino Muzzi Filho, 529 - Caixa Postal 2041
17525-901 Marília, SP

leonardo@ucdb.br, delamaro@fundanet.br, rfaraujo@gmail.com

***Abstract.*** *Code mobility is an alternative approach to the implementation of distributed systems. It presents several advantages over conventional server/client architecture as greater flexibility and reduced bandwidth consumption. On the other hand, having a host executing an unknown piece of code, coming from a possible unknown place is a threat to the system security. In this paper we describe the implementation of security features using code signature and permissions on top of a pre-existing API that supports code mobility for Java programs. The new API, built on top of $\mu$Code is called safe-$\mu$Code.*
*Key-words: Mobile Code, Security.*

***Resumo.*** *Mobilidade de código é uma abordagem alternativa a implementação de sistemas distribuídos. Ela apresenta diversas vantagens em relação as tradicionais arquiteturas cliente/servidor trazendo maior flexibilidade e reduzindo o consumo de banda. Por outro lado, executar em uma máquina um código desconhecido, oriundo de uma localização possivelmente desconhecida também é uma ameaça à segurança do sistema. Neste trabalho é descrita a implementação de características de segurança através de assinatura de código e permissões em uma API já existente e que suporta mobilidade de código para programas Java. A nova API, construída sobre o ambiente $\mu$Code é chamada safe-$\mu$Code.*
*Palavras-chave: Códigos Móveis, Segurança.*

## 1. Introdução

A necessidade por aplicações mais flexíveis, associada ao surgimento de novos domínios de aplicação, motivaram a procura por abordagens alternativas as técnicas tradicionalmente utilizadas no desenvolvimento de aplicações distribuídas. Assim, através da introdução do chamado grau de mobilidade no desenvolvimento dessas aplicações, foi possível inovar a forma como tais aplicações são concebidas e construídas, uma vez que permitem desvincular o fragmento de código do local onde ele será executado [FUGGETTA et al., 1998, TANENBAUM and STEEN, 2002].

Além de **Mobilidade Física**, o tipo de mobilidade de uma aplicação também pode ser classificado como **Mobilidade Lógica** ou **Mobilidade de Código**, que pode ser entendida como sendo um software que viaja através de uma rede heterogênea, atravessando domínios de segurança e sendo executado automaticamente no seu destino, o que favorece o desenvolvimento de aplicações e serviços mais flexíveis,

dinâmicos e customizados [WANGHAN and FRAGA, 2001][FUGGETTA et al., 1998, TANENBAUM and STEEN, 2002].

Dentre os reais desafios a serem enfrentados no contexto da efetiva utilização da mobilidade de código, um dos principais tópicos é a questão da segurança e suas implicações. No cerne dessa discussão, está a necessidade em evitar que informações e recursos sejam acessados ou manipulados ilegitimamente, ou mesmo, garantir a correta execução das aplicações [SCHODER and EYMANN, 2000, WANGHAN and FRAGA, 2001].

Este trabalho aborda a implementação de um mecanismo explícito de segurança para o ambiente $\mu$Code, através do uso dos recursos de assinatura de código e permissões objetiva-se iniciar a construção de uma versão segura para o ambiente mencionado, permitindo, em um primeiro momento, a proteção das máquinas pertecentes a um domínio de aplicação contra os chamados códigos maliciosos.

## 2. Segurança em Código Móvel

O ambiente computacional de uma linguagem de código móvel está ligado a uma plataforma distribuída, onde aplicações baseadas em códigos móveis pertencentes a diferentes usuários, podem ser concorrentemente executadas. Dessa forma, uma máquina pode receber unidades de execução que pertençam a diferentes usuários e que tenham diferentes permissões de acesso aos recursos da máquina [CUGOLA et al., 1997].

As questões relacionadas à segurança, em linhas gerais, envolve a proteção dos elementos que compõem o sistema da chamada computação *maliciosa*, as ameaças relacionadas ao uso de códigos móveis, podem ser categorizadas como: (a) Agentes *vs.* Plataforma de Agentes; (b) Plataforma de Agentes *vs.* Agentes; (c) Agentes *vs.* Agentes e (d) Elementos Externos *vs.* Sistema (Plataforma de Agentes + Agentes) [WANGHAN and FRAGA, 2001].

## 3. *Safe* $\mu$CODE

Desde sua concepção, o $\mu$CODE [1] objetivou oferecer um ambiente com a flexibilidade e a extensibilidade necessárias ao desenvolvimento de aplicações distribuídas voltadas à ambientes de larga escala, como a Internet. Um dos principais diferenciais deste ambiente está no fato de que seu enfoque principal não são apenas os agentes móveis, mas sim contemplar também os demais paradigmas de mobilidade de código existentes, como por exemplo: código sob demanda e invocação remota [PICCO, 1998].

A versão original do ambiente $\mu$CODE é desprovida de qualquer mecanismo especialmente projetado com o intuito de prover segurança, sendo que compõem o núcleo deste ambiente os conceitos de ***Group***, ***Group Handler***, $\mu$***Server*** e ***Class Space***. Assim, a implementação do mecanismo explicíto de segurança para o ambiente $\mu$CODE abordará inicialmente somente a proteção dos servidores $\mu$*Server* dos chamados códigos maliosos, através do uso de assinatura de código e permissões.

A versão segura do ambiente $\mu$CODE está baseada na construção e utilização de uma unidade de migração digitalmente assinada, o *Signed Group*, e na atribuição de permissões que possibilitem disciplinar a execução das aplicações com base no grau de confiança do emissor. O *Group* desempenha o papel da unidade de migração no ambiente, permitindo que os programadores tenham a mão um container passível de ser

[1] Ambiente desenvolvido pelo prof. Dr. Gian Pietro Picco (Dipartimento di Elettronica e Informazione/Politecnico di Milano) e disponível para download através da URL: http://mucode.sourceforge.net/

arbitrariamente preenchido com classes e objetos, incluindo objetos do tipo threads, e que posteriormente podem ser remetidos a servidores *μServer*

### 3.1. Funcionamento do *Safe* μCODE

A migração de um grupo digitalmente assinado pelos servidores *μServer* que compõem uma aplicação está representada na Figura 1 , sendo que a partir da máquina "Cliente", um *SignedGroup* possui condições de migrar entre os diversos servidores *μServer* relacionados a uma aplicação, as setas bidirecionais na Figura 1 reforçam a ausência de uma ordem pré-estabelecida para visita dos servidores. No momento da migração, a chave privada do emissor será utilizada para realizar a assinatura digital das classes inseridas no grupo sendo que, também compõe o *Signed Group* um rótulo que permitirá reconhecimento do emissor posteriormente.

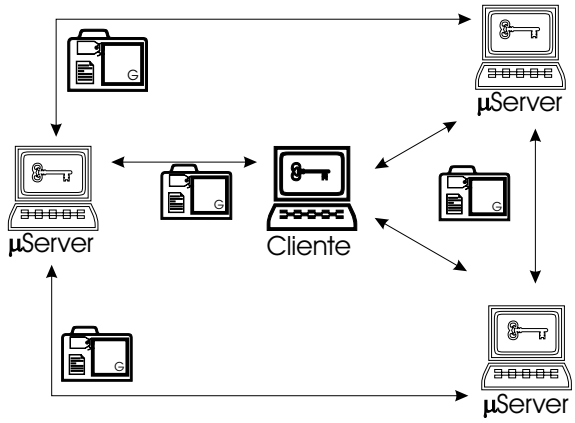


**Figura 1: Ambiente μCODE Seguro - Migração**

No destino, baseado no rótulo que acompanha o grupo assinado, o servidor *μServer* realiza uma consulta à chave pública do usuário-emissor visando verificar e validar a assinatura digital das classes, dando início ao processo de extração do conteúdo que forma o grupo recebido. Havendo divergência entre a assinatura do grupo e a chave pública do usuário, uma exceção será gerada e propagada pelo sistema, do contrário, os itens que compõe o grupo serão alocados em um *ClassSpace*, ficando a cargo do *μClassLoader* controlar o carregamento e execução das classes de acordo com a política de segurança estabelecida.

Introduzidas no mecanismo de segurança da linguagem Java, a partir do Java 2, as permissões desempenham um importante papel no ambiente *Safe* μCODE, uma vez que caberá a elas disciplinar a forma como as aplicações são executadas. As políticas de segurança estarão previamente alocadas em cada servidor *μServer*, expressas na forma de um arquivo texto no formato definido pela linguagem Java [GONG, 1999, McGRAW and FELTEN, 1998].

Todos os mecanismos de realocação de códigos originalmente suportados pelo ambiente μCODE, são mantidos na versão segura preservando assim um dos principais diferenciais do ambiente em relação aos seus pares, a flexibilidade na escolha do paradigma a ser utilizado no momento da implementação de suas aplicações.

## 4. Conclusões e Trabalhos Futuros

Este artigo apresentou a criação do *safe* μCODE, uma versão com características de segurança baseadas em assinatura de código de código e permissões para o ambiente

$\mu$CODE, onde através da implementação de um mecanismo explícito de segurança, torna-se possível realizar a proteção dos servidores deste ambiente contra códigos móveis maliciosos, característica até então inexistente na versão original do ambiente.

Dentre os próximos encaminhamentos deste trabalho, está a evolução do mecanismo de segurança através da agregação de recursos para proteção de códigos e agentes móveis contra as chamadas plataformas maliciosas, no caso do ambiente em questão, servidores *$\mu$Server* mal-intencionados.

## Referências

CUGOLA, G., GHEZZI, C., PICCO, G. P., and VIGNA, G. (1997). Analyzing mobile code languages. In *Mobile Object Systems: Towards the Programmable Internet*, pages 93–110. Springer-Verlag: Heidelberg, Germany.

FUGGETTA, A., PICCO, G. P., and VIGNA, G. (1998). Understanding code mobility. *IEEE Transactions on Software Engineering*, 24(5).

GONG, L. (1999). *Inside Java 2 Plataform Security - Architecture, API Design and Implementation*. Addison Wesley Longman Inc, Palo Alto - CA - USA.

McGRAW, G. and FELTEN, E. W. (1998). Mobile code and security. *IEEE Internet Computing*.

PICCO, G. P. (1998). $\mu$CODE: A lightweight and flexible mobile code toolkit. In *Mobile Agents - Proceedings of the 2nd International Workshop on Mobile Agents*, volume 1477 of *ISBN 3-540-64959-X*, pages 160–171, Stuttgart (Germany). K. Rothermel and F. Holh.

SCHODER, D. and EYMANN, T. (2000). The real challenges of mobile agents. *Communications of the ACM*, 43(6).

TANENBAUM, A. S. and STEEN, M. v. (2002). *Distributed Systems - Principles and Paradigms*. Prentice Hall, Upper Saddle River, New Jersey 07458.

WANGHAN, M. S. and FRAGA, J. d. S. (2001). (mini-curso) agentes móveis x segurança. Universidade Federal de Santa Catarina - UFSC. Simpósio sobre Segurança em Informação - SSI 2001.