

# OSGi™ – Um Sistema Dinâmico de Módulos para Java

Raoni Kulesza  
raoni@larc.usp.br

MAC-5863

Sistemas de *Middleware* Avançados 2006/2

Prof. Dr. Francisco Reverbel

# Agenda

- Introdução
- Arquitetura Base do *Framework OSGi*
  - Ambiente de Execução
  - Módulos
  - Ciclo de Vida
  - Serviços
  - Segurança
- Aplicações e Ferramentas *OSGi*
- Considerações Finais
- Referências

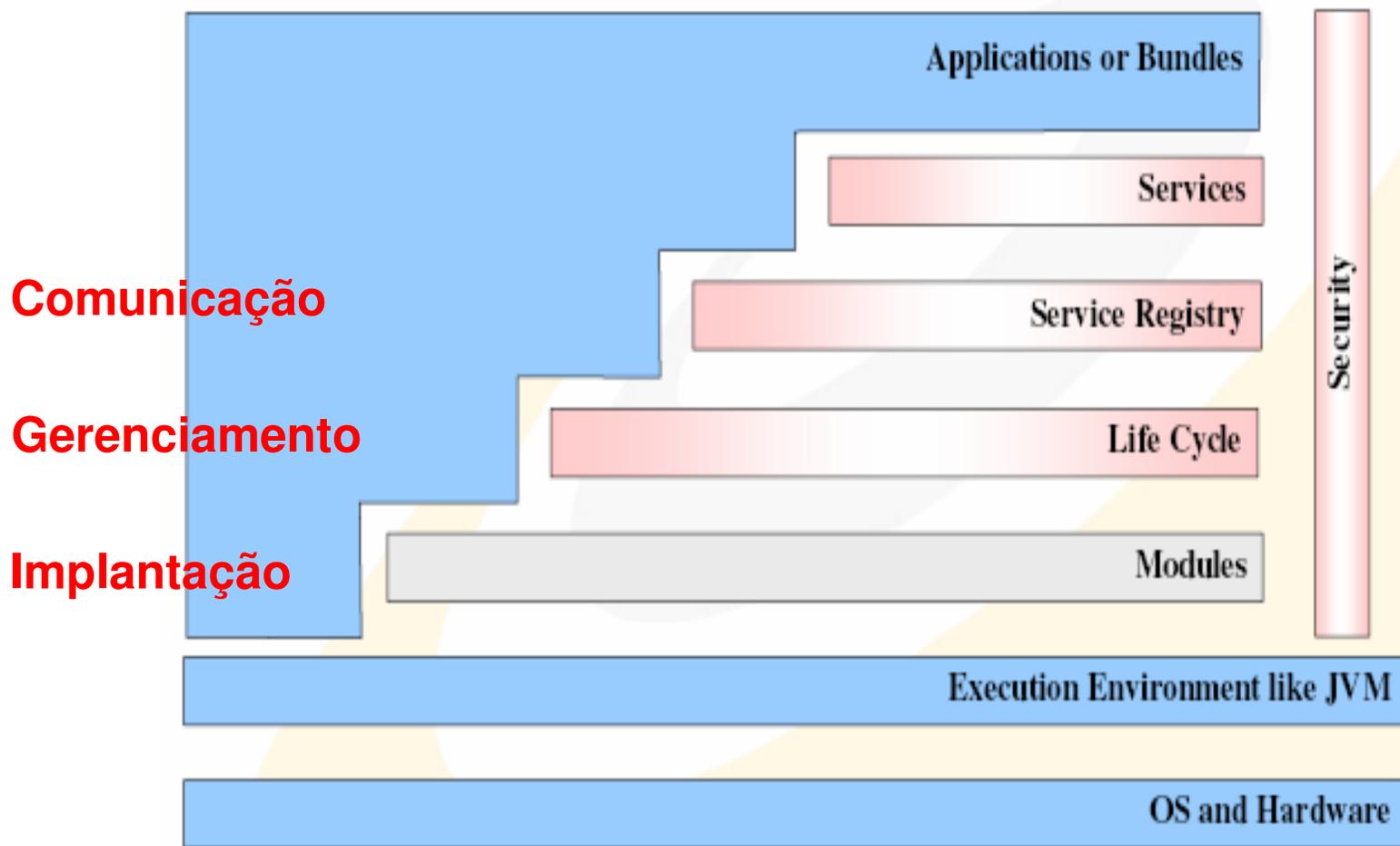
# Introdução

- *Open Services Gateway Initiative – OSGi Alliance*
  - Fundada em março de 1999
  - Organização independente, sem fins lucrativos
  - Define e promove especificações abertas para distribuição de serviços gerenciáveis através de ambientes em rede
- Especificações da *OSGi Service Platform*
  - Aberta e Padronizada
  - Orientado a componentes
  - Serviços em rede

# Objetivos da *OSGi Service Platform*

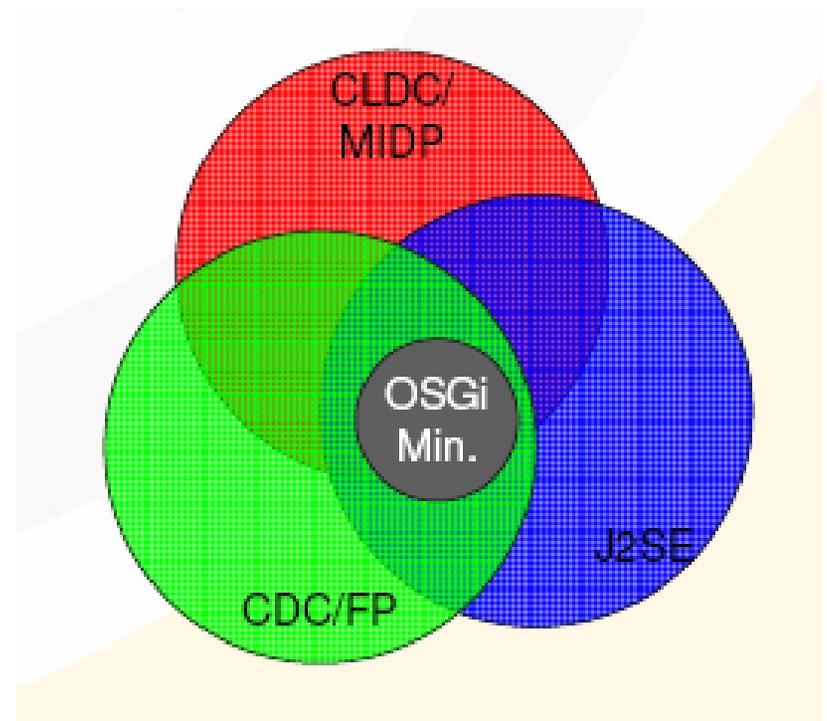
- *Framework* Java™ para desenvolver software baseado em componentes que possibilite implantação remota e:
  - Confiável
  - Distribuição e Gerenciamento em Larga Escala
  - Vários tipos de dispositivos
  - Colaborativo
- Criado através de colaboração de várias indústrias
- Atualmente na especificação 4.0
- Procura resolver problemas da área de desenvolvimento baseado em componentes

# Arquitetura Base



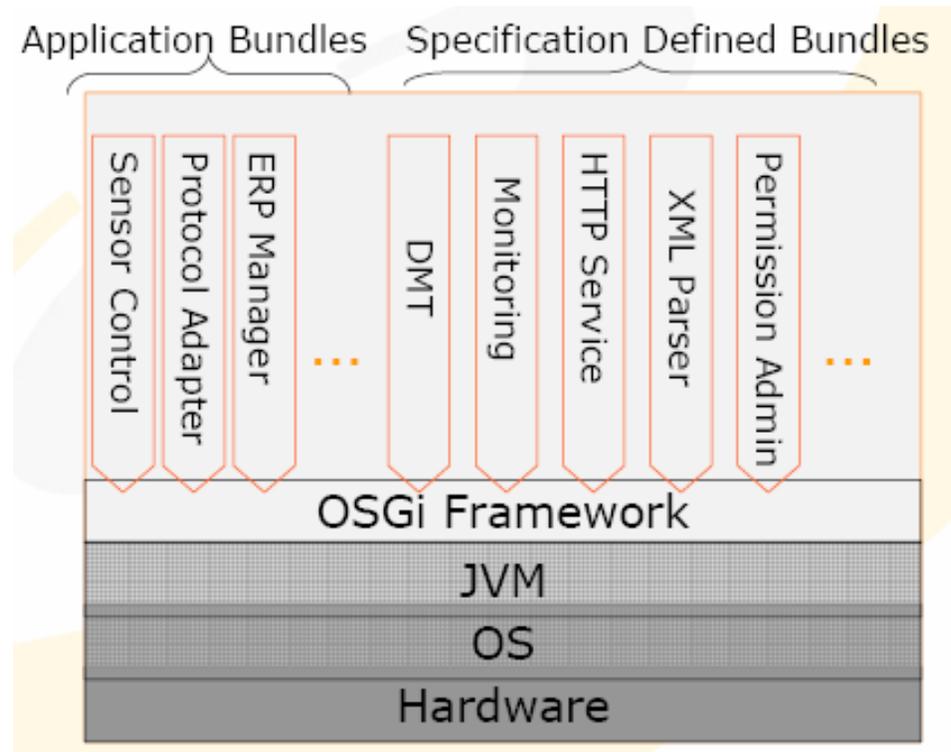
# Ambiente de Execução

- As APIs usam apenas um sub-conjunto de J2SE e J2ME CDC
  - AE OSGi Mínimo
- Implementações podem estender o AE OSGi Mínimo
- Segurança não é obrigatório
- Possibilidade de compatibilidade com CLDC



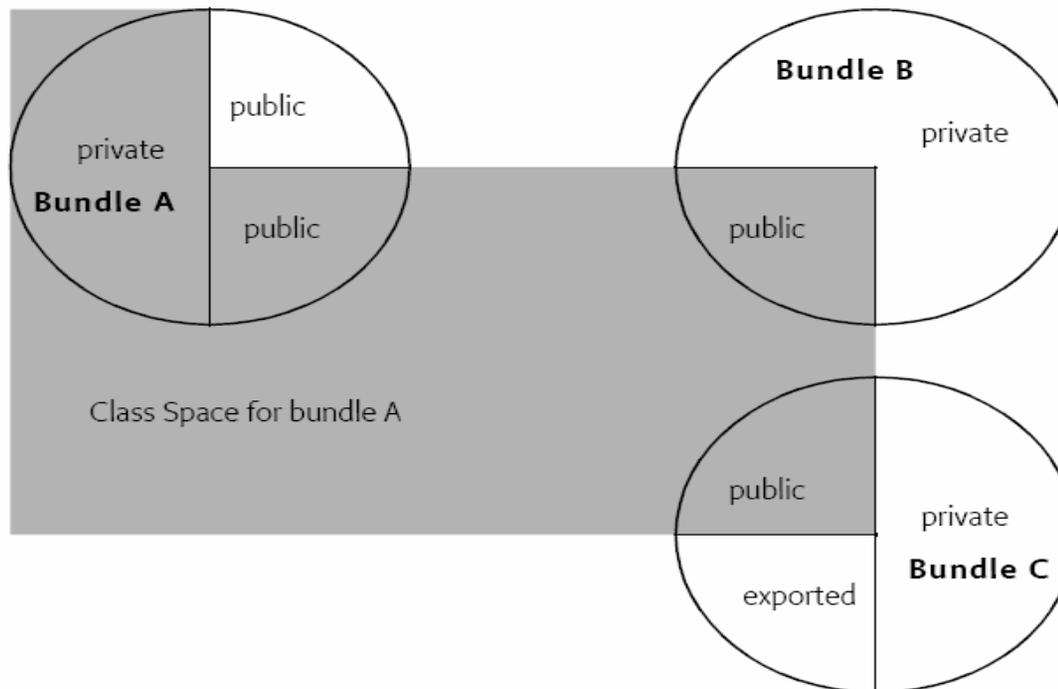
# Módulos - *Bundles*

- *Bundles* – Unidade de modularização (JAR)
  - Código, *Manifest*, Recursos e [OSGI-OPT]
- Pode-se importar e exportar serviços
- Vários *bundles* podem ser executados numa única JVM



# Módulos – Carga de Classes

- Estende o modelo de implantação da plataforma Java
  - Modularização, Proteção e Controle de Versão
  - Framework OSGi é uma rede de *classloaders*
- Boot -> *Framework* -> *Bundle space*

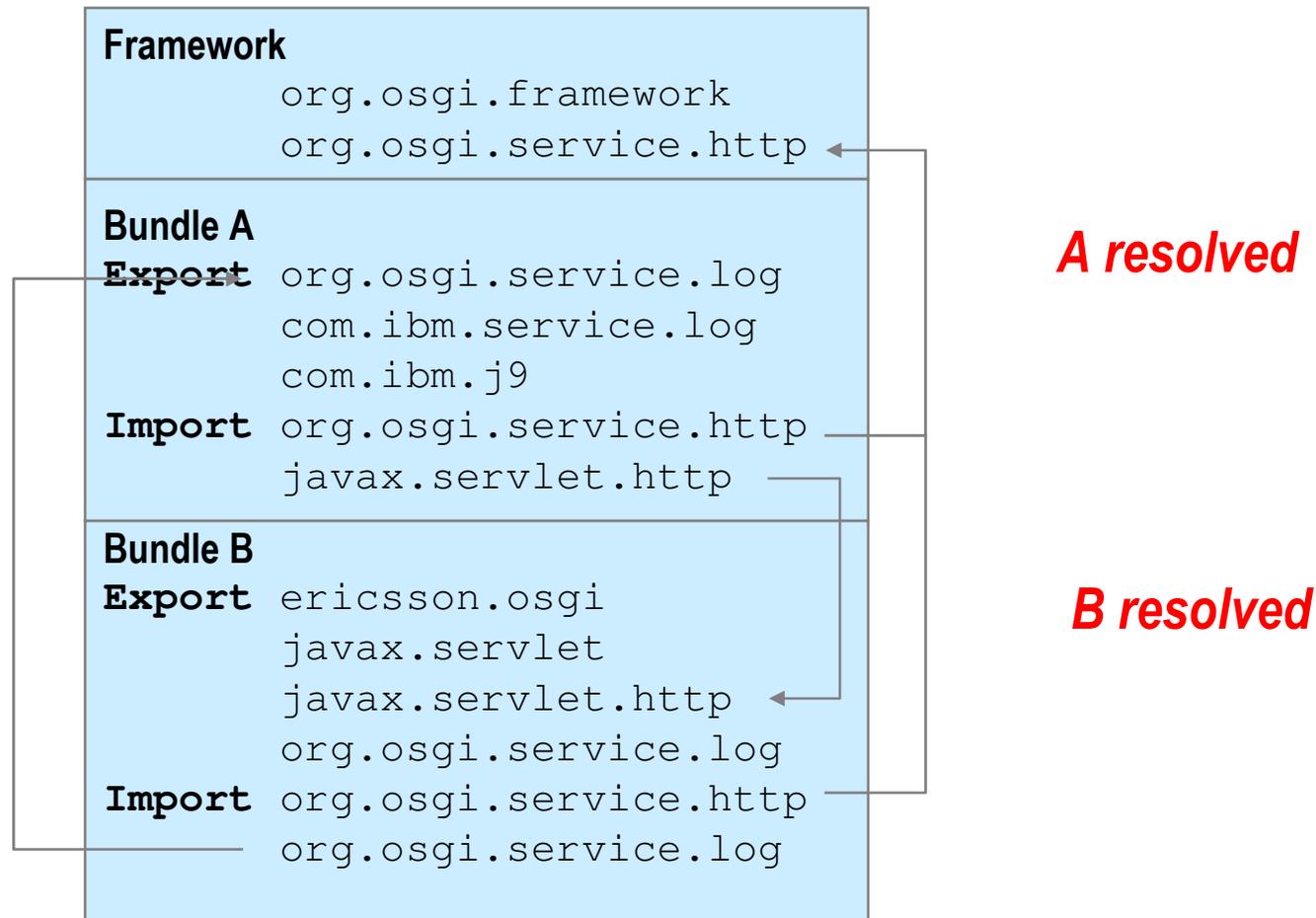


# Módulos – Dependências

- *Bundles* devem ser resolvidos pelo framework (*resolver*)
  - O processo de associar pacotes exportados e importados para um *bundle*
- Deve ser realizado antes de carregar e executar o *bundle*
- Descrito através de cabeçalhos específicos do arquivo de *MANIFEST*
  - *Import-Package Header*
  - *Export-Package Header*
  - *Required-Bundle*

Ex.: *Import-Package: br.usp.ime.foo; br.usp.ime.bar; version="[1.0,1.24]"; resolution:=mandatory*

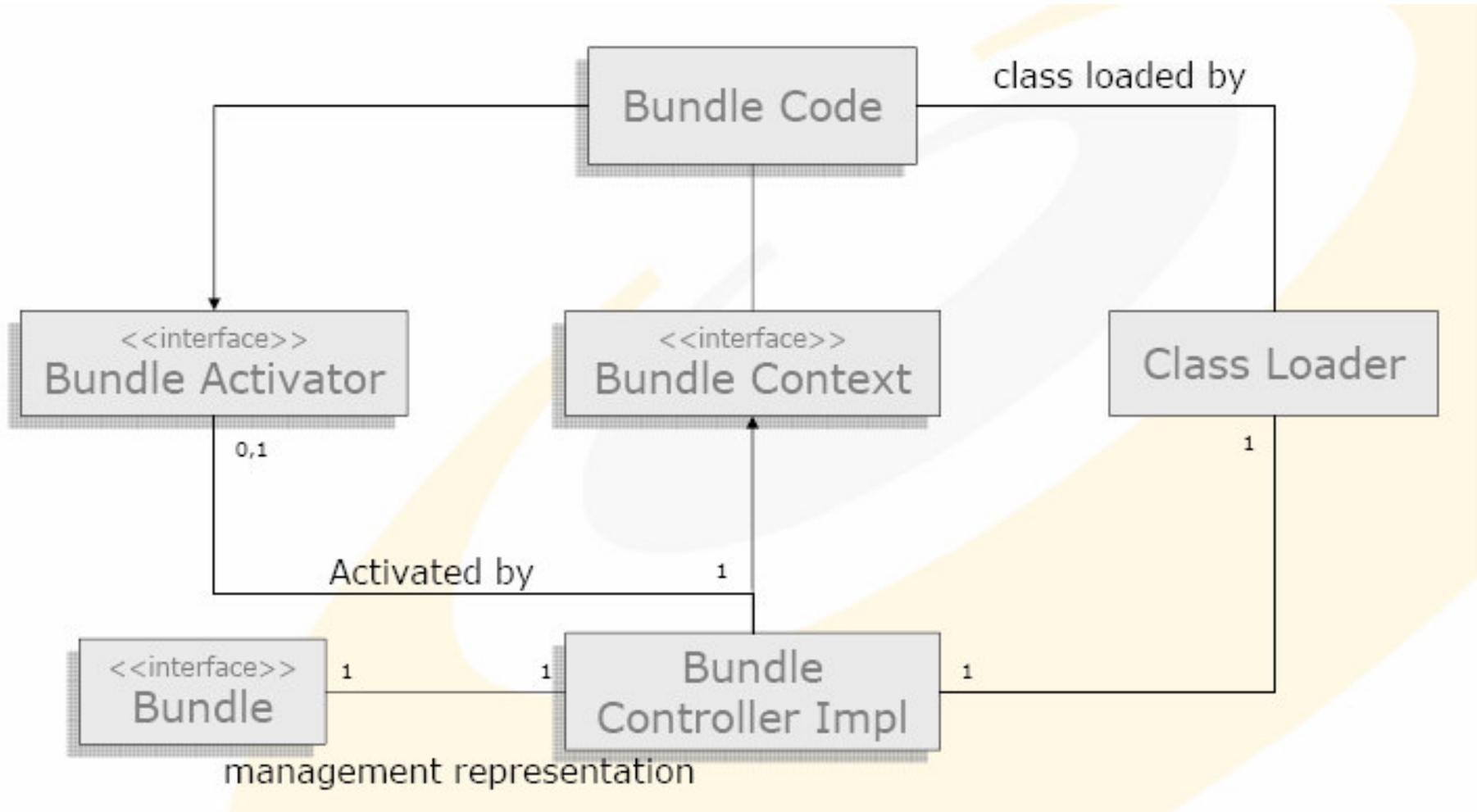
# Módulos – Resolução de Dependência



# Ciclo de Vida – Visão Geral

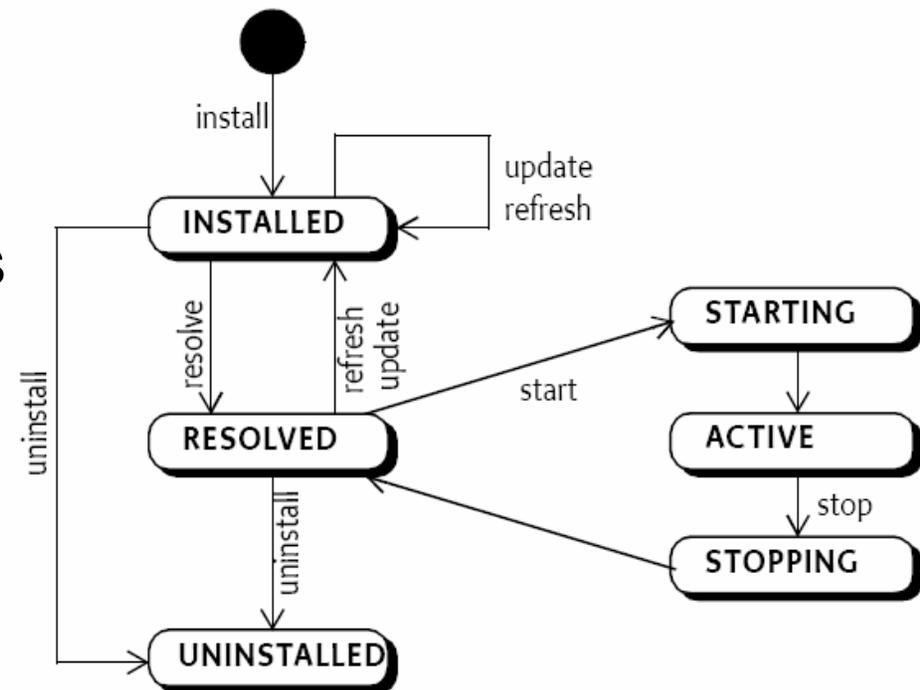
- Define uma API para controlar operações de segurança e ciclo de vida dos *bundles*
  - Instalação
  - Inicialização
  - Finalização
  - Atualização
  - Desinstalação
  - Monitoração
- Disponibiliza acesso ao *framework (system bundle)*
- Permite o gerenciamento remoto da plataforma

# Ciclo de Vida – Arquitetura



# Ciclo de Vida – Funcionamento

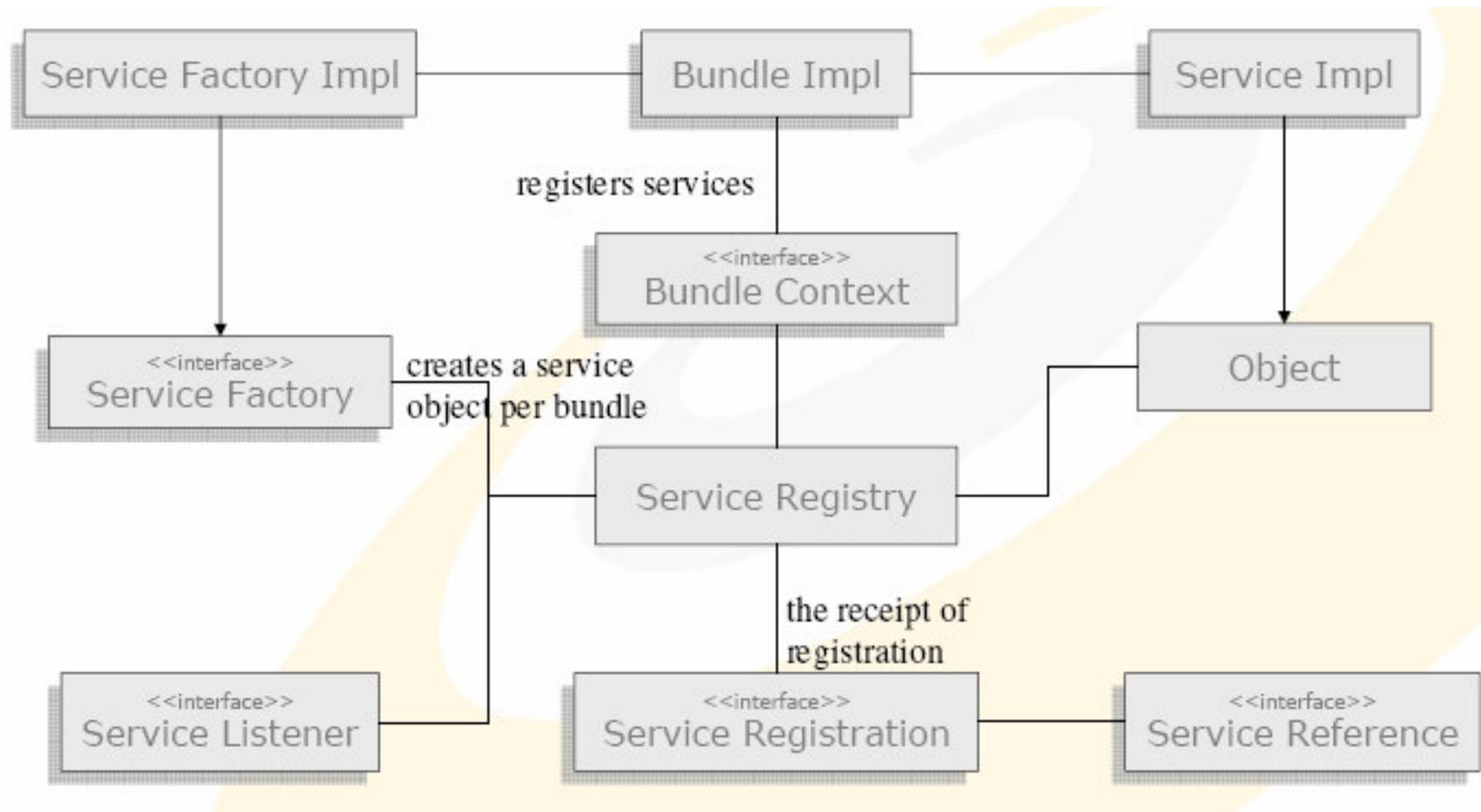
- O *Bundle* é iniciado pela classe *Bundle Activator* (definido no *Manifest*)
- 2 métodos
  - Start: Inicializa o mais rápido possível
  - Stop: desaloca recursos
- *Bundle Context* provê acesso ao *framework*
- Framework provê serviço com níveis e grupos de inicialização



# Serviços – Visão Geral

- Serviços são objetos Java normais que são registrados através de uma ou mais interfaces Java no registro de serviços (*Service Registry*)
- Publicação, Pesquisa e Acoplamento dinâmico
- Políticas de acesso e controle de versão para serviços
- Rastreamento de serviços durante a reinicialização do *framework*
- Provê arquitetura orientada a serviços “*in-VM*”

# Serviços - Arquitetura



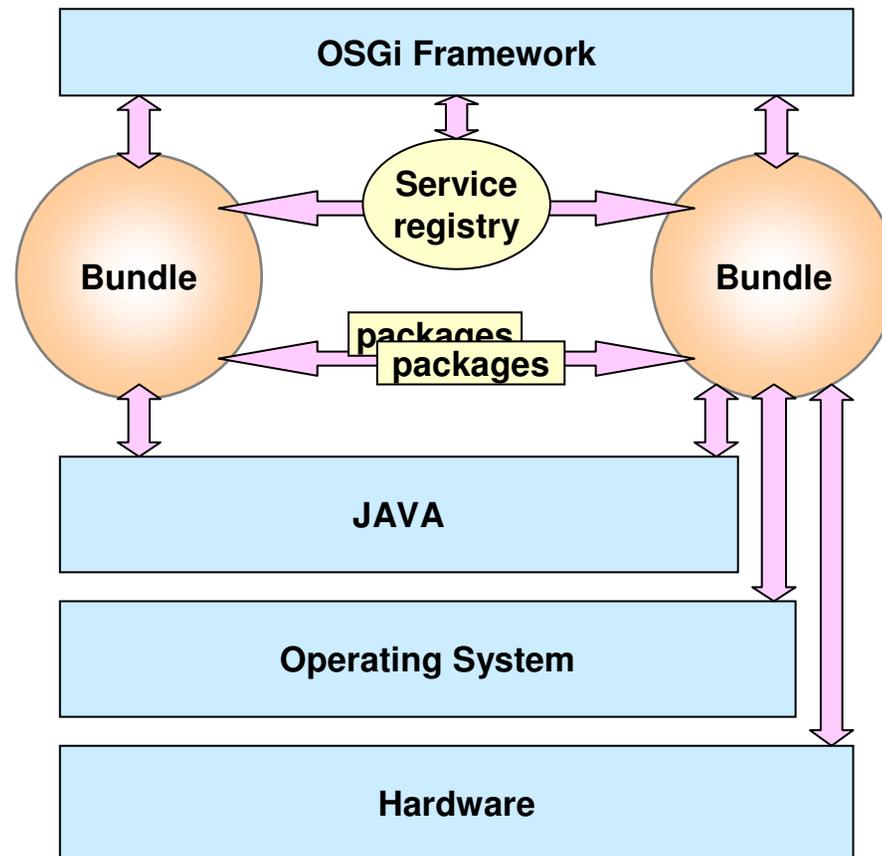
# Serviços - Funcionamento

- Registrando Serviços
  - *ServiceRegistration* e *ServiceReference*
  - *BundleContext.registerService()*
- Propriedades de Serviço
  - Par de chave-valor
  - Informações que descrevem serviços para seu “público”
- Identificador de Persistência (*Persistent Identifier* – PID)
  - Auxilia no controle de reinicialização do framework
- Localizando Serviços
  - Obtém *ServiceReference* através de *BundleContext.getServiceReference()*

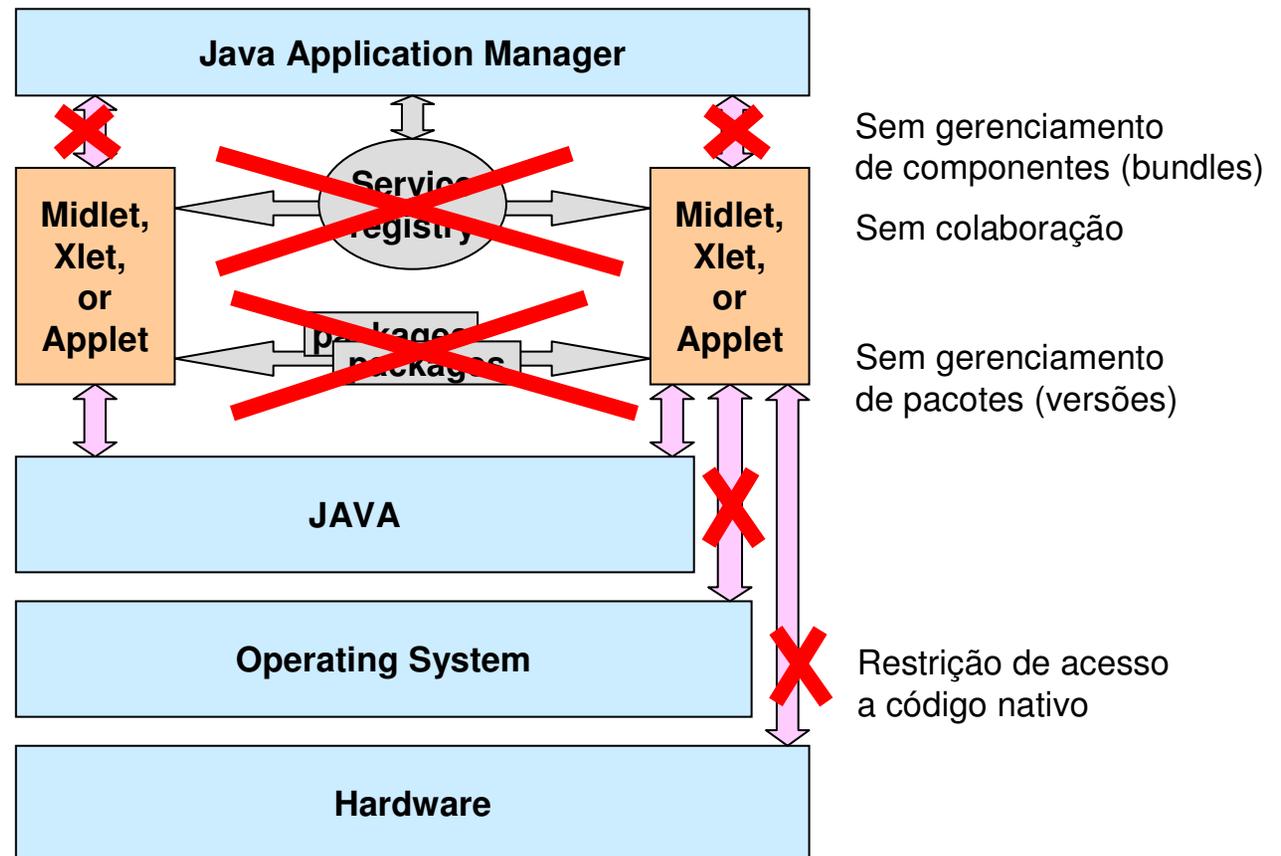
# Modelo de Colaboração

- *O framework OSGi é mais do que um ambiente de execução de Applets, MIDlets, Xlets*
- *Bundles* podem colaborar através de:
  - Objetos de serviços
  - Compartilhamento de pacotes
- O registro dinâmico permite que os *bundles* procurem e rastreiem objetos de serviços
- *O framework gerencia essa colaboração*
  - Dependências, segurança

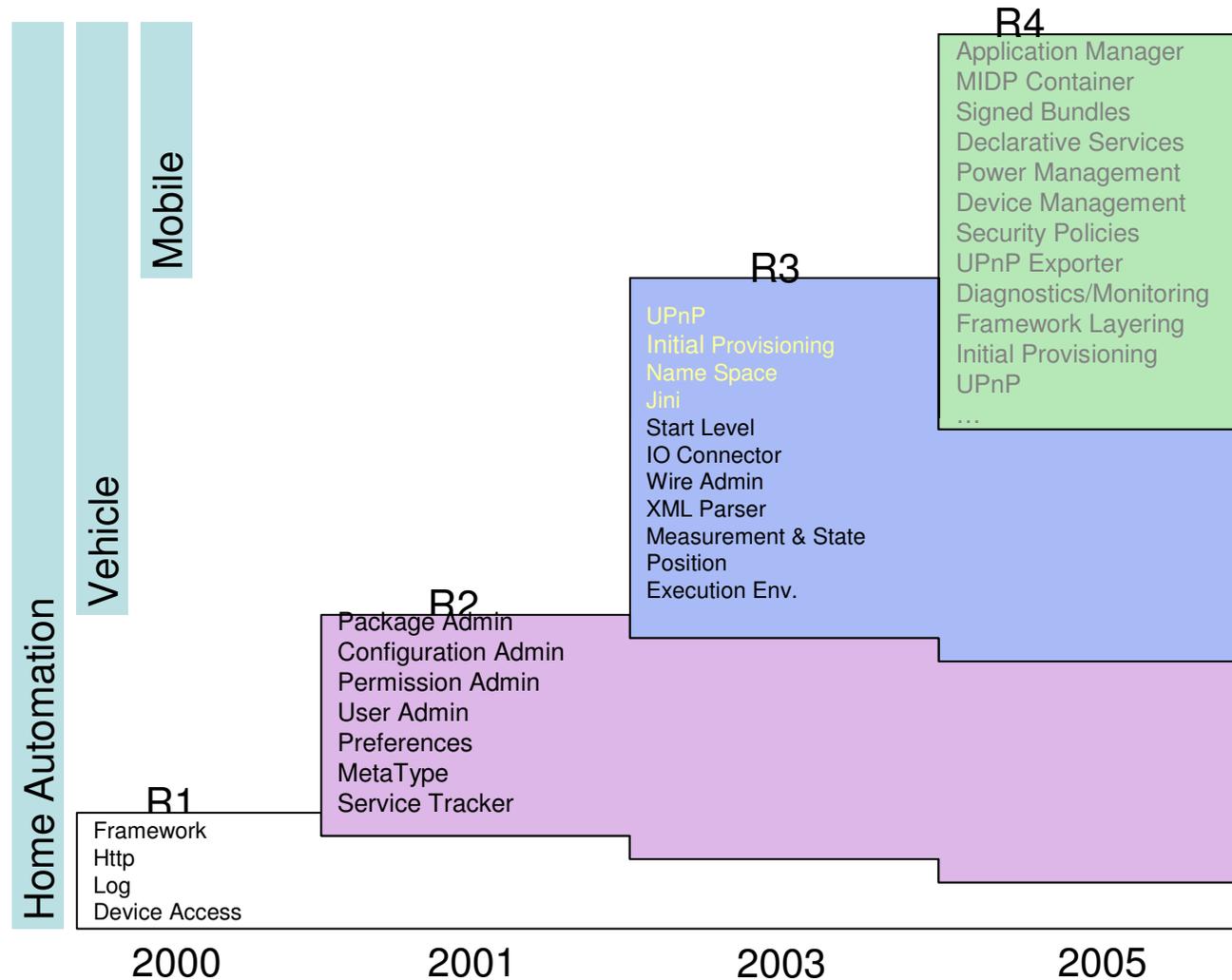
# Modelo de Colaboração



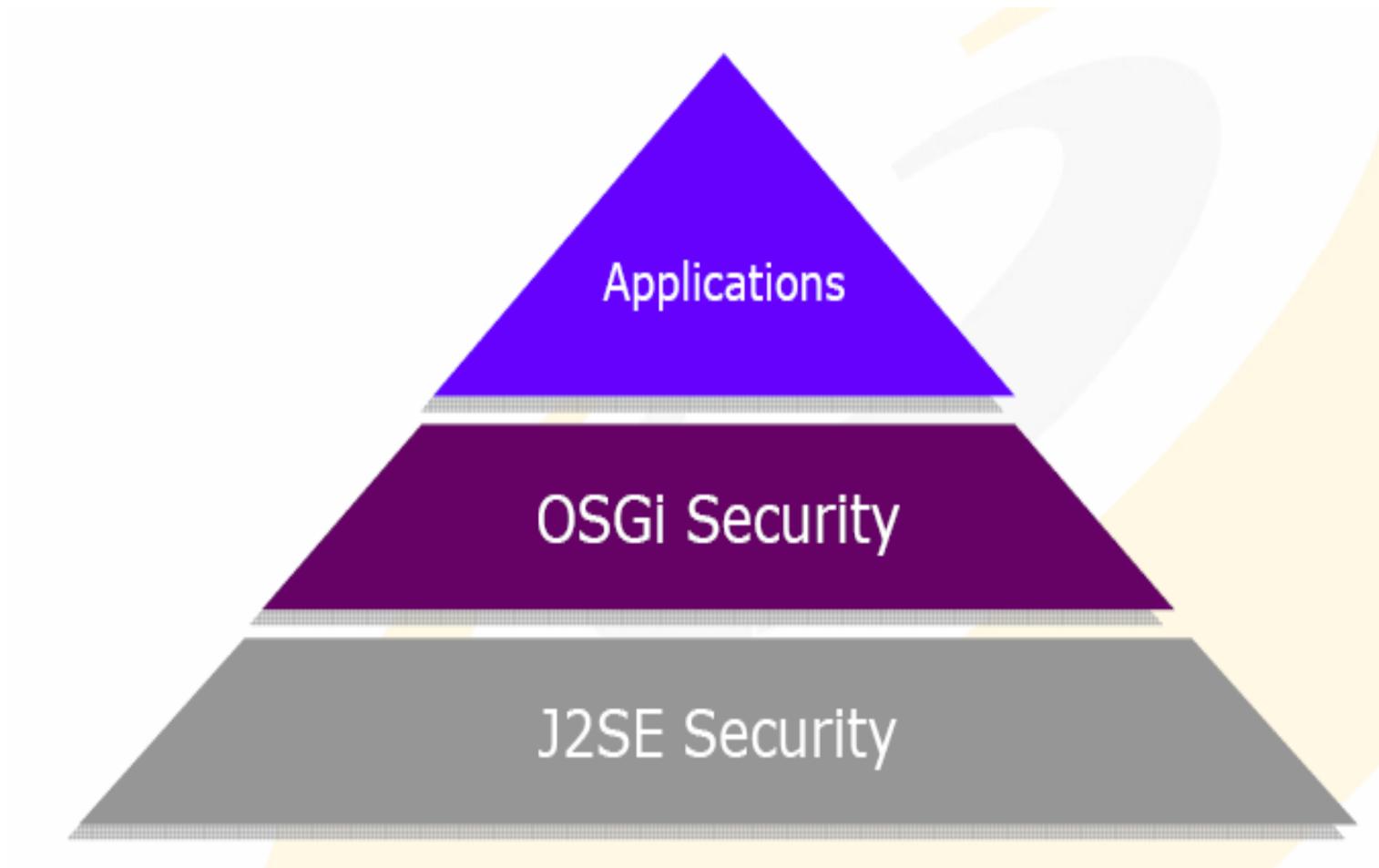
# Modelo de Colaboração

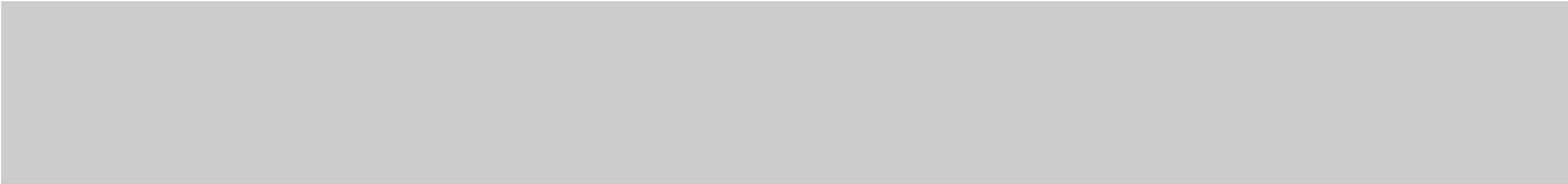


# Serviços Padrão OSGi



# Segurança





# Aplicações e Ferramentas OSGi

# Aplicações

- Arquiteturas de Gerenciamento para Dispositivos Móveis (JSR – 232)
- Eclipse - Equinox
- ObjectWeb – Jonas J2EE Server
- WebSphere Everyplace Device 6.0
- Siemens VDO Top Level Architecture
- Global System for Telematics
  
- Muitas outras aplicações...
  - BMW, Nokia, Siemens, Philips, MHP-GEM...

## Ferramentas *Open-source*

- Eclipse Equinox: implementação OSGi R4
- Oscar: implementação R4 da *ObjectWeb*
- Apache Felix: projeto baseado no Oscar (R3) e incubado na *ASF*
- Knopflerfish: implementação R4 da *Gatespace Telematics*

# Considerações Finais

- Especificação *OSGi* é bastante abrangente e esta apresentação foi apenas uma introdução
- A plataforma de serviços *OSGi* (*Service Platform*) provê um modelo de componentes baseado em:
  - Serviços e compartilhamento de pacotes
  - Modelo de programação dinâmico
- Bundles são unidades de modularização com:
  - Implantação independente, fácil de desenvolver e permitem reutilização
- Existem várias aplicações e produtos (comerciais e *open-source*) *OSGi*
- A abrangência da plataforma (celulares, PDA, sistemas embarcados, PC, servidores) é o ponto forte do padrão (*client middleware*, computação ubíqua e autônoma)

# Referências

- *OSGi Service Platform, Core Specification, Release 4, v.4.0.1, Julho 2006*
  - [http://www.osgi.org/resources/spec\\_download.asp](http://www.osgi.org/resources/spec_download.asp)
- *About the OSGi Service Platform, Technical Whitepaper, Revision 4.1, Novembro 2005*
  - [http://www.osgi.org/resources/spec\\_download.asp](http://www.osgi.org/resources/spec_download.asp)
- OSGi Service Platform
  - André Assad – Recife, Sun Tech Days 2006
- OSGi™ Component Programming
  - Thomas Watson e Peter Kriens - EclipseCon2006
- <http://www.osgi.org>
- <http://bundles.osgi.org>
- <http://www.eclipse.org/osgi>
- <http://www.aqute.biz>