TDC2012

the developer's conference

Trilha – Java
Modularização em Java com OSGI

Filipe Portes

Mestre de Obra de Software

Globalcode

Mim Tarzan...

TDC2012
the developer's conference

- Quase graduado em Ciência da Computação
- Um dos Coordenadores do Grupo de Usuários Java de Goiás - @gojava
- Mais de 5 anos de experiência com desenvolvimento e arquitetura Java
- Atualmente trabalha no Centro de Gestão e Estudos Estratégicos em Brasília – DF
- Sósia oficial do Brad Pitt

Globalcode

Modularização

TDC2012 the developer's conference

- Definindo um módulo:

  "A Software Module is a **deployable, manageable, natively reusable, composable, stateless** unit of software that **provides a concise interface** to consumers"

  - Instalável
  - Gerenciável
  - Reutilizável
  - Combinável
  - Não guarda estado
  - Oferece uma Interface clara

Fonte: Java Application Architecture: Modularity Patterns with Examples Using OSGi

Globalcode

TDC2012
the developer's conference

Globalcode

TDC2012
the developer's conference

Globalcode

2 Faces da Modularização

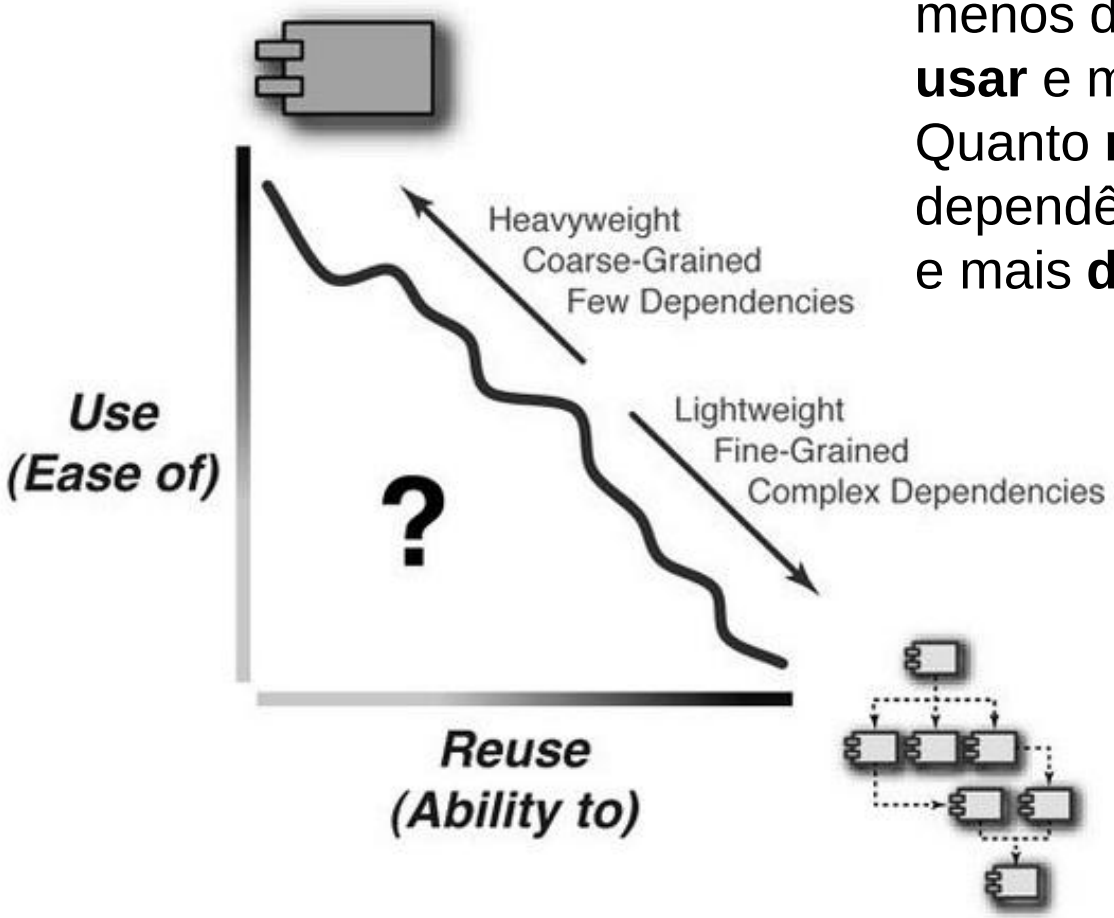
- Modelo de Desenvolvimento
  - Formas de construir arquiteturas modulares, e tratar os problemas comuns nesse cenário.
  - Design Patterns
- Modelo de Execução
  - Foca em como gerenciador sistemas modulares em tempo de execução, ou seja, Plataformas que suportem um Eco-sistema Modular oferecendo recursos que facilitem e potencializem a modularização
  - OSGI, jigsaw, etc...

Design Lógico e Físico

- **Design Lógico:** podemos dizer que são as relações entre as classes, métodos e pacotes. Como serão as camadas. Padrão MVC,TDD, DDD, TDC, ABC, XYZ, XPO...
- **Design Físico:** como serão as unidades de deploy desse software, apenas 1 projeto, 1 projeto principal e outros menores, como se dará a comunicação entre eles?

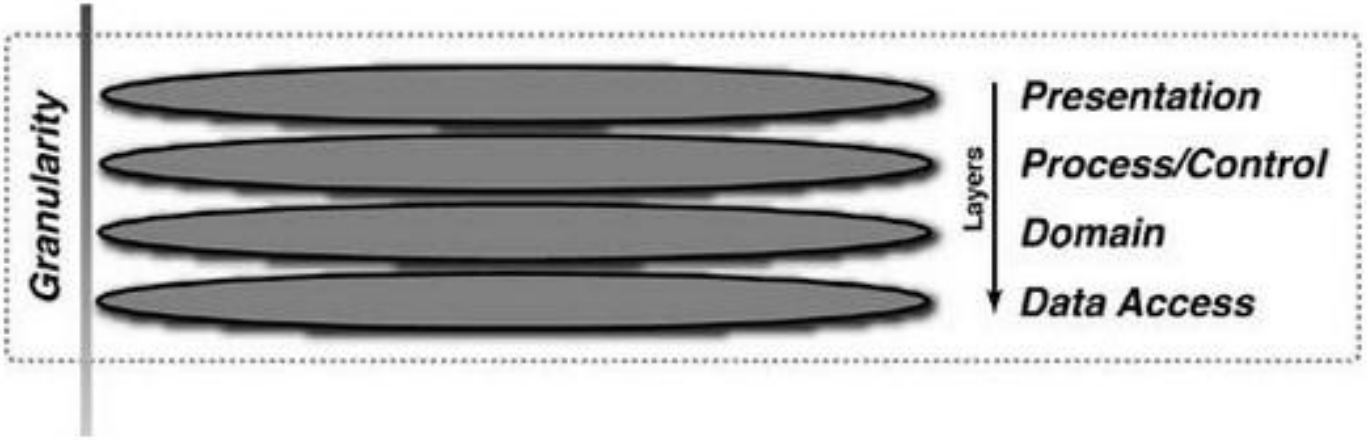
Globalcode

Uso vs Reuso



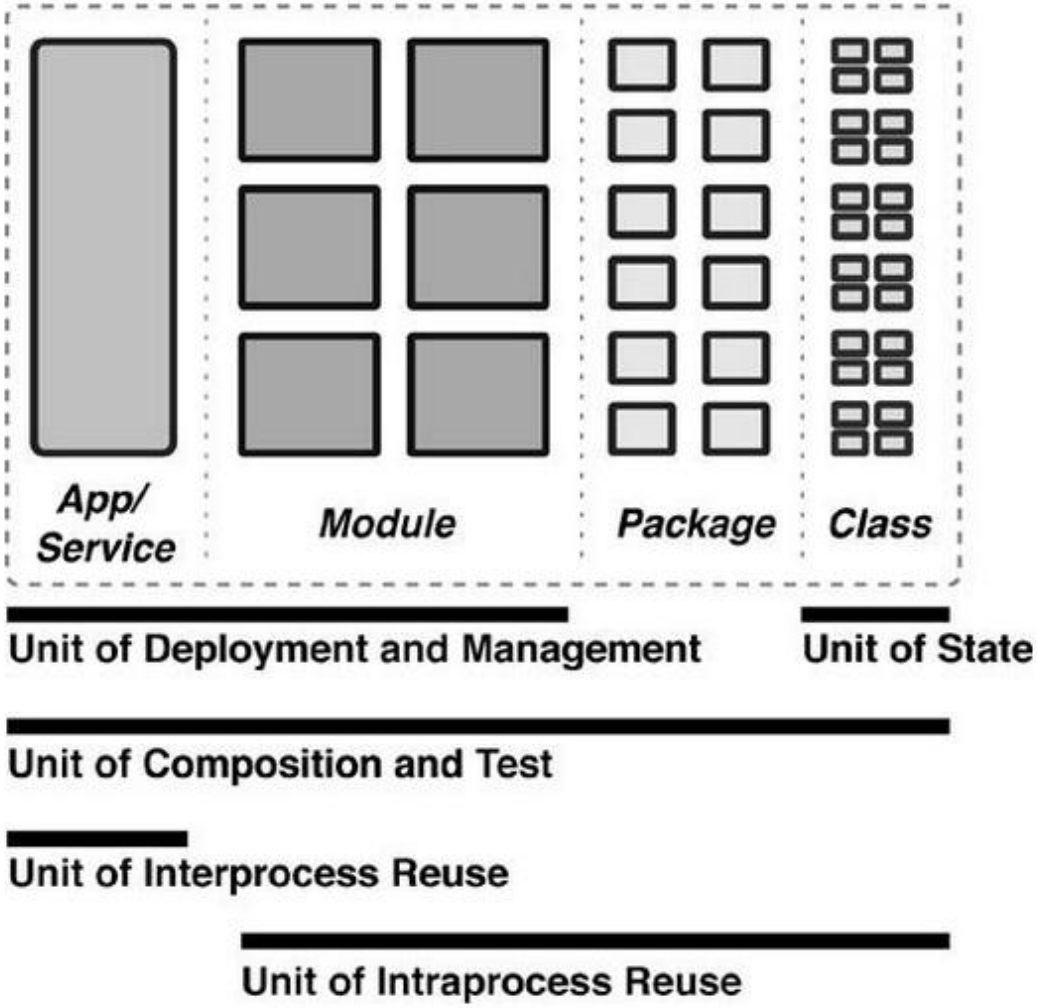
Quanto **maior a granularidade**, menos dependências, mais **fácil de usar** e mais **difícil de reutilizar**.
Quanto **menor a granularidade**, mais dependências, mais **fácil de reutilizar** e mais **difícil de usar**.

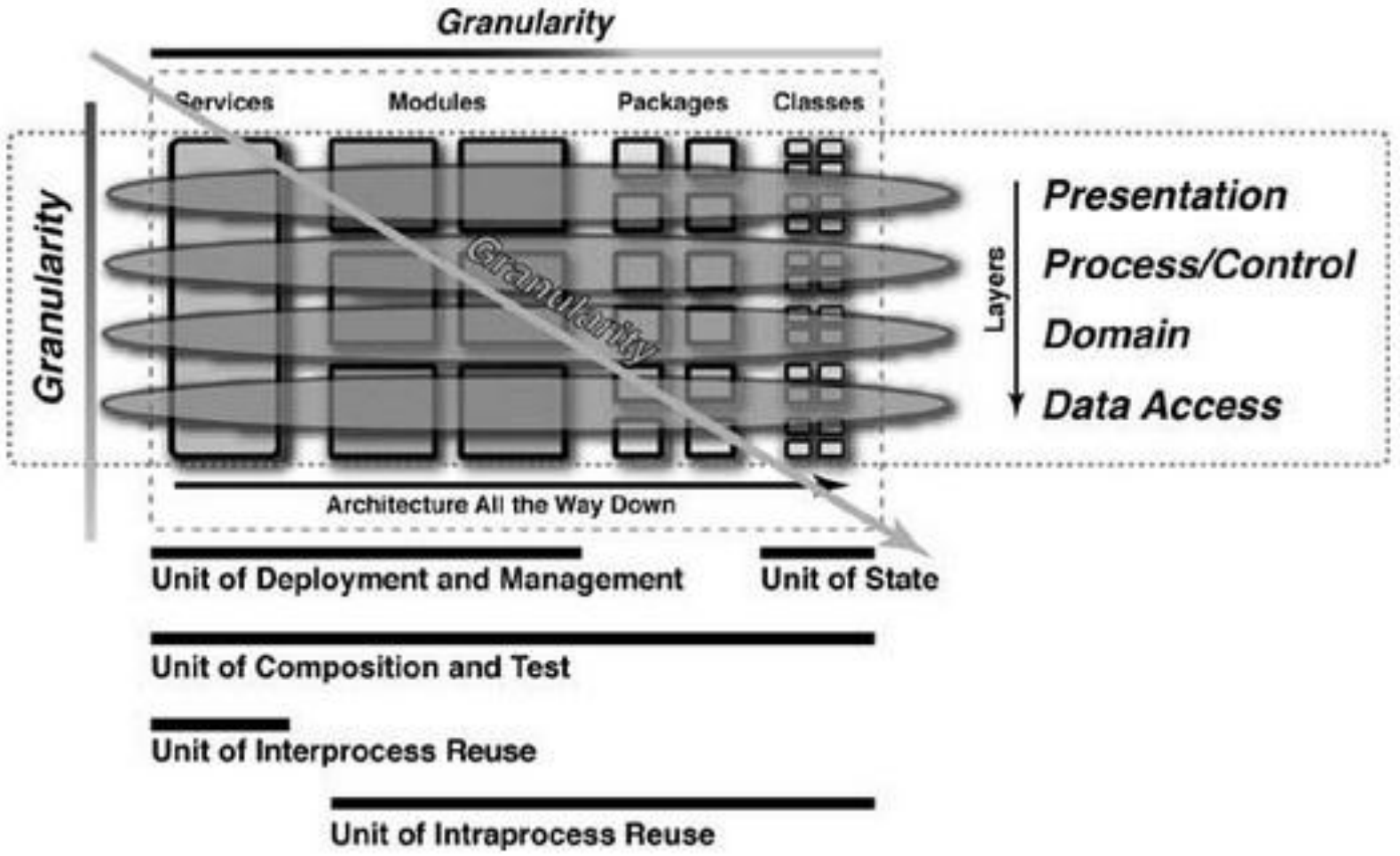
Design em camadas



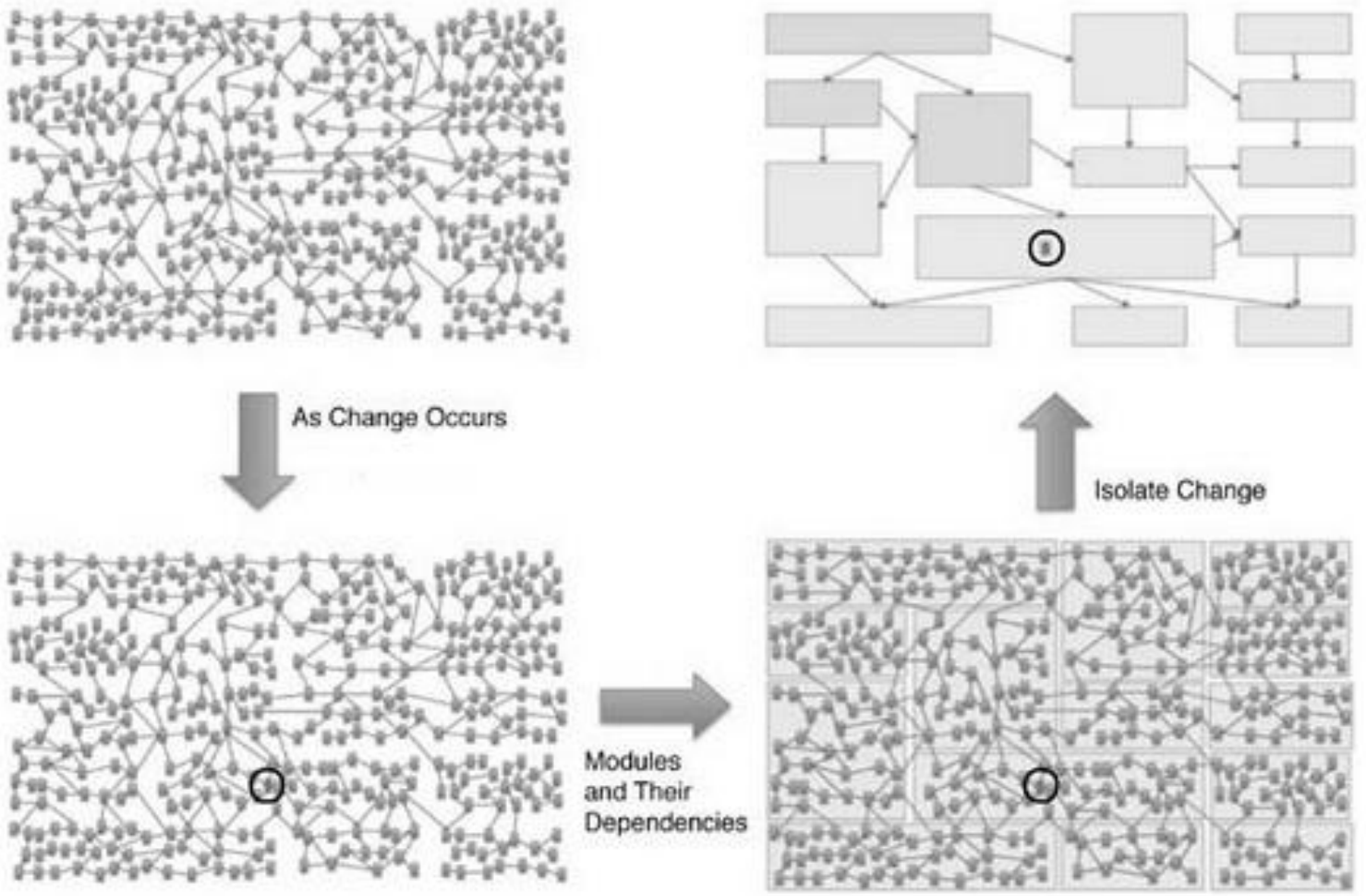
Design comum encontrado em diversos sistemas, nas camadas superiores observa-se uma **granularidade maior**, ou seja, entidades mais **fácil de se usar**, a medida que se desce para as camadas inferiores a **granularidade diminui**, ou seja, entidades menores e mais **fáceis de reutilizar**.

Design Modular

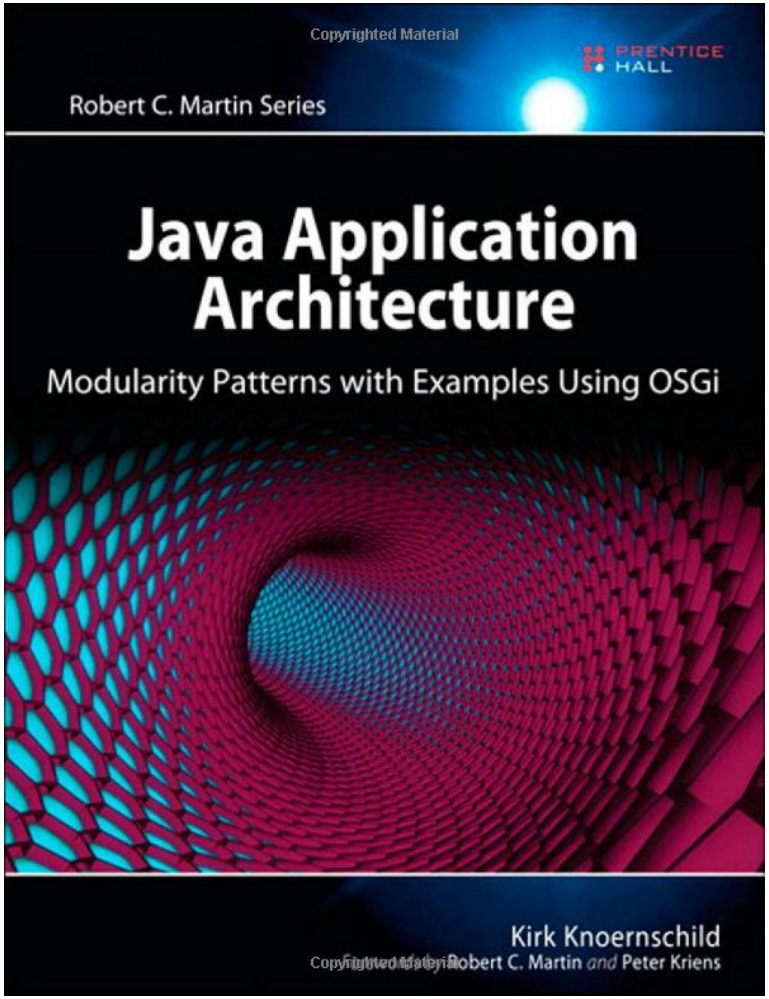



Granularity
Services
Modules
Packages
Classes
Granularity
Granularity
Layers
Presentation
Process/Control
Domain
Data Access
Architecture All the Way Down
Unit of Deployment and Management
Unit of State
Unit of Composition and Test
Unit of Interprocess Reuse
Unit of Intraprocess Reuse

Mudanças Isoladas



Bibliografia



Excelente Livro sobre Modularização de Software e Design Patterns com exemplos práticos em Java e OSGI

"Esse livro non Ecxsisssteee" – Padre Quevedo sobre esse livro

"Meu Precioooossooo" – Gollum sobre esse Livro

"Linnndo, Luuuxo, Maara, D++, Liiindoo" – Narcisa sobre esse livro

"Aaaahôôôoo trem que pula" – típico goiâno sobre esse livro

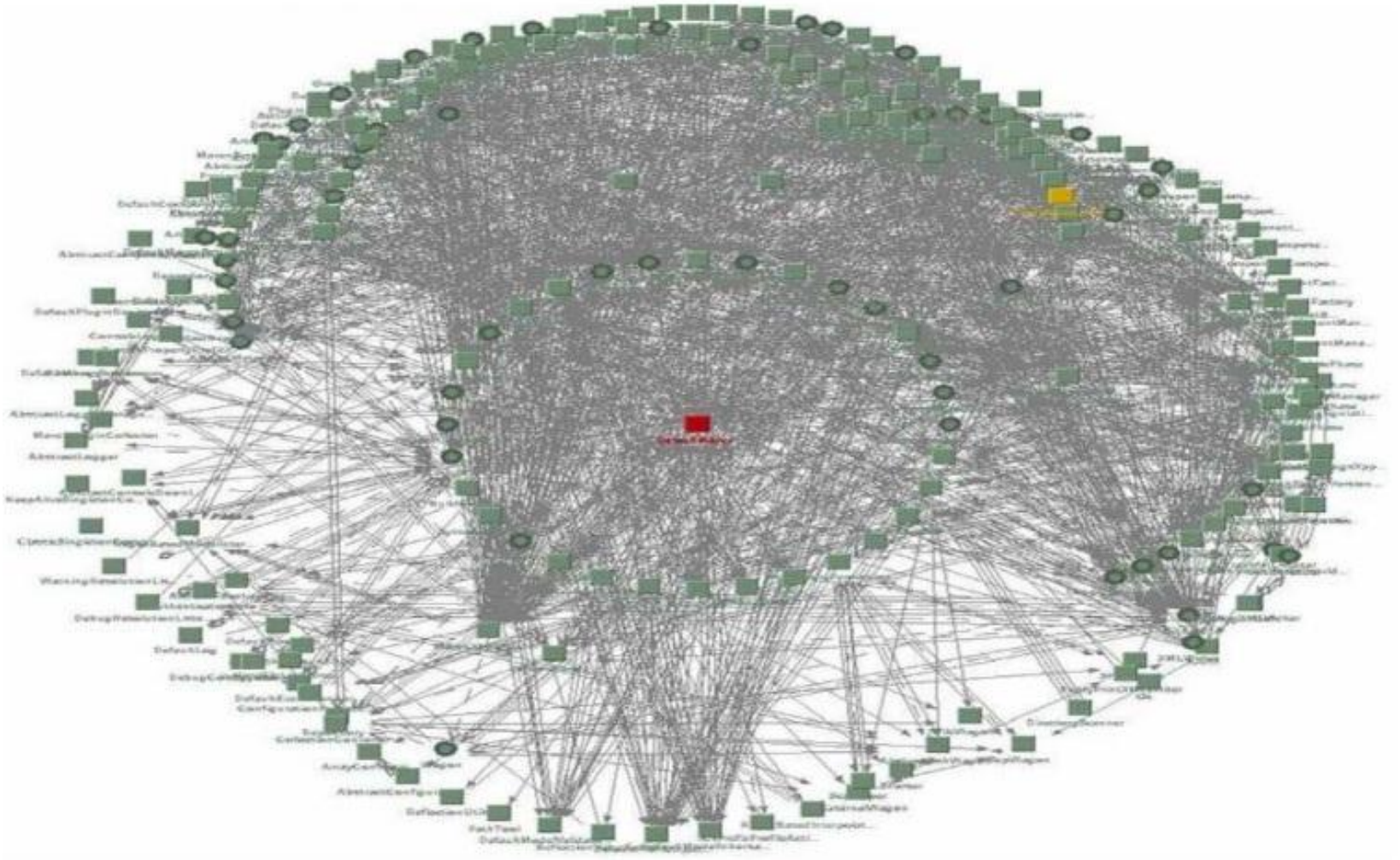
*As imagens utilizadas nos slides anteriores foram retiradas do livro, Por favor não me processem!!
http://www.amazon.com/Java-Application-Architecture-Modularity-Patterns/dp/0321247132

Principais Patterns

- 18 patterns descritos no padrão GOF
- Base Patterns
  - Manage relationships
  - Module Reuse
  - Cohesive Modules
- Depency Patterns
  - Acyclic Relationships
  - Physical Layers
- Usability Patterns
  - Published Interface
  - External Configuration
- Extensibility Patterns
- Utility Patterns

Globalcode

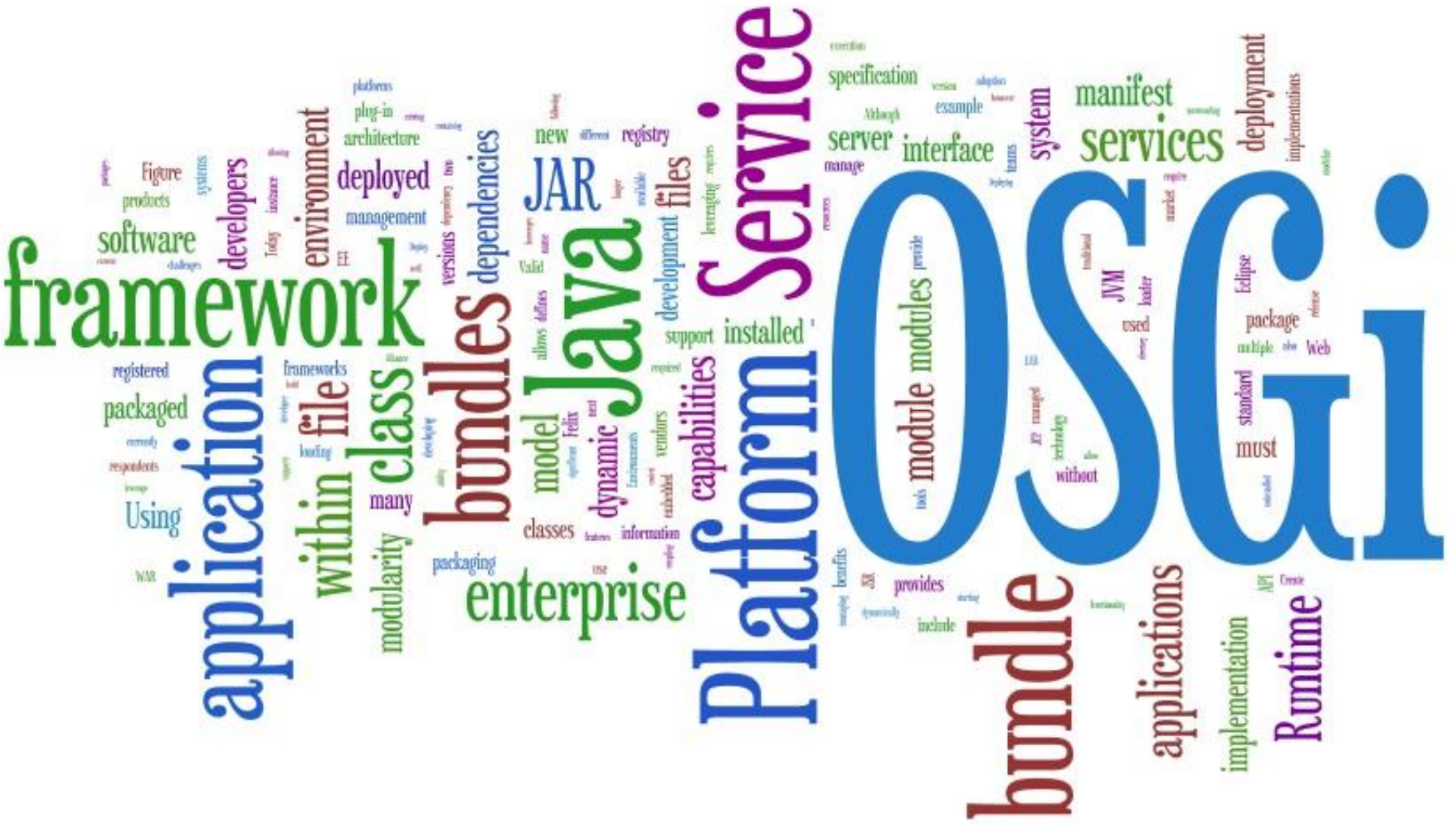
Jar Hell



TDC2012
the developer's conference

Globalcode

TDC2012
the developer's conference



Globalcode

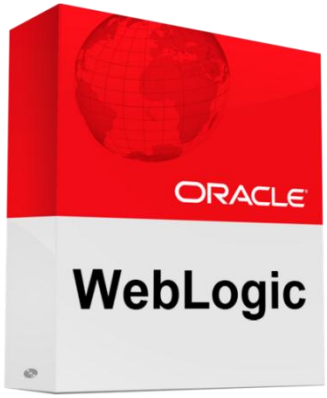
- "Just about **every** software developer is an **OSGi consumer** today because just about **every platform and every IDE use OSGi**. The major platform vendors, including IBM, Oracle, and Red Hat are all using OSGi to build up their platforms. What's interesting is that **OSGi hasn't penetrated the enterprise** developer space yet. At least, it hasn't gone mainstream yet. Some people might complain that **OSGi is too complex**. But what they're really saying is that **designing modular software is really really hard**. Because it is." – Kirk Knoernschild

http://java.dzone.com/articles/osgi-vs-jigsaw-kirk


TDC2012
the developer's conference


NetBeans


eclipse


ORACLE
WebLogic


OSGi™
Alliance


as7


GlassFish

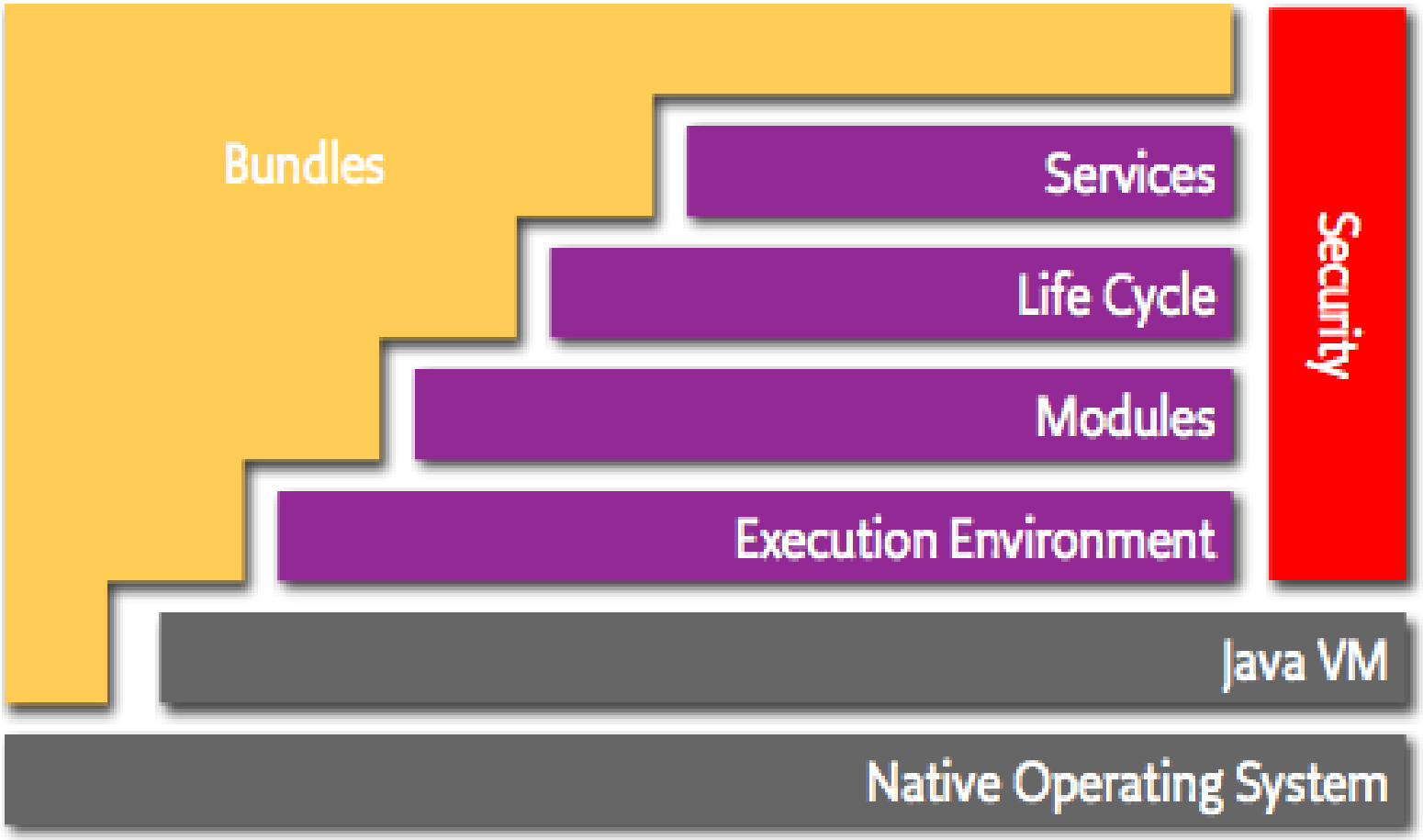

IBM
WebSphere


Globalcode

Principais Benefícios

- Encapsulamento
- Deploy Dinâmico
- Versionamento
- Gerenciamento de Dependências
- Outros: http://www.osgi.org/Technology/WhyOSGi

Globalcode

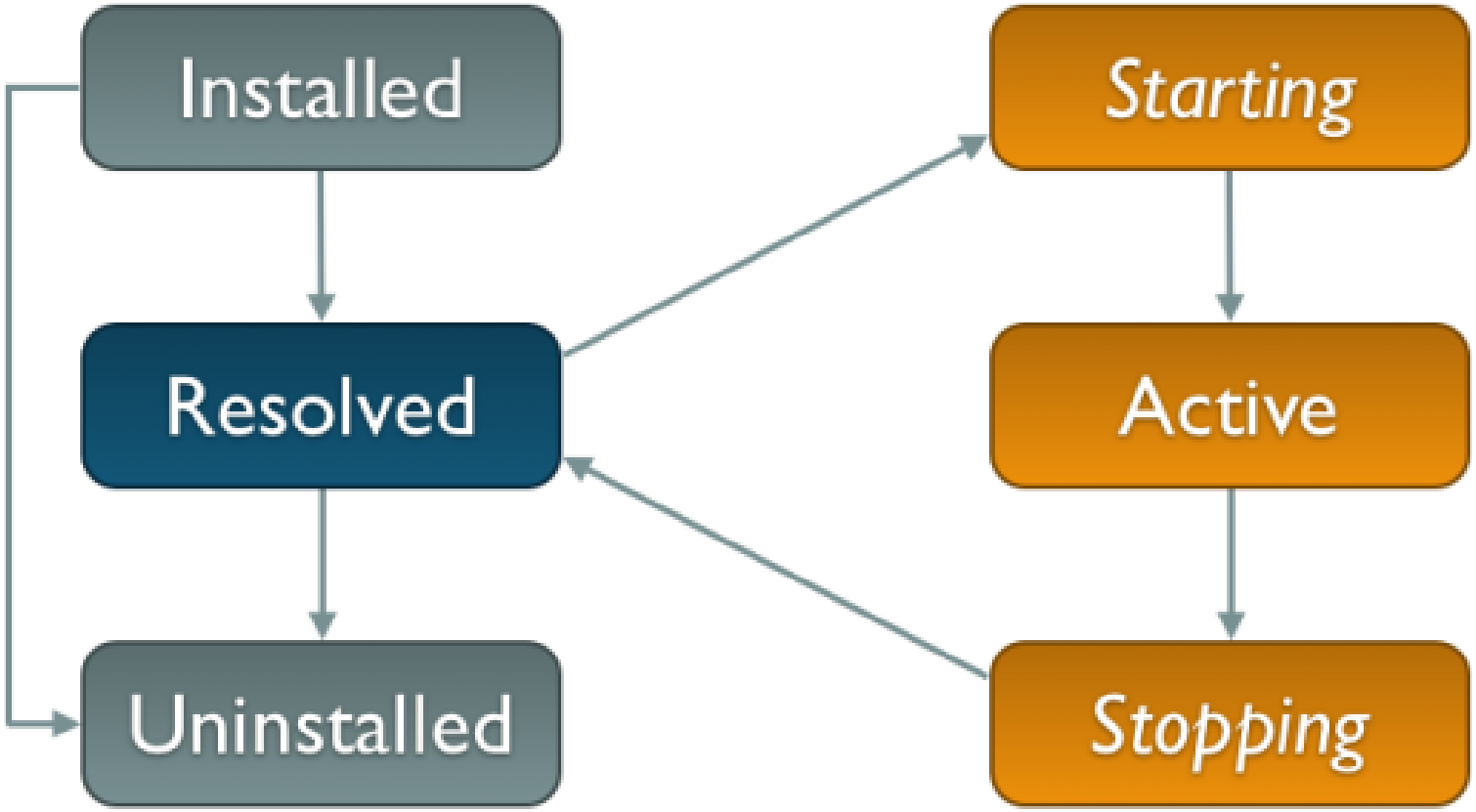

Bundles
Services
Life Cycle
Modules
Execution Environment
Security
Java VM
Native Operating System

Módulos = Bundles

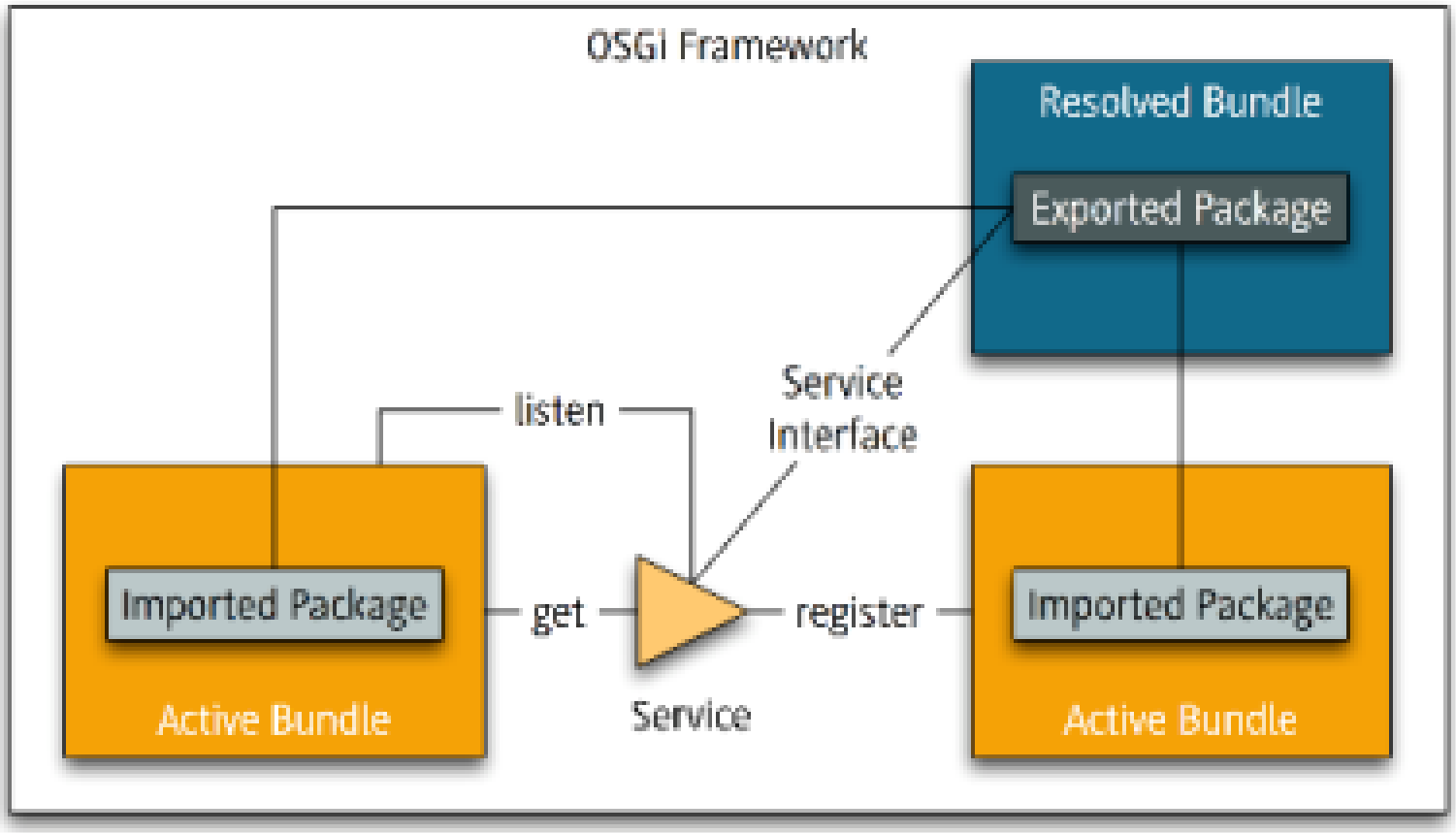
- Bundle é um módulo ou parte dele
- Equivalente a um Jar, porém com diversas informações adicionais que são interpretadas pela plataforma.
  - Identificador
  - Versão
  - Dependências (bundles e versões)
  - Exposições/publicações

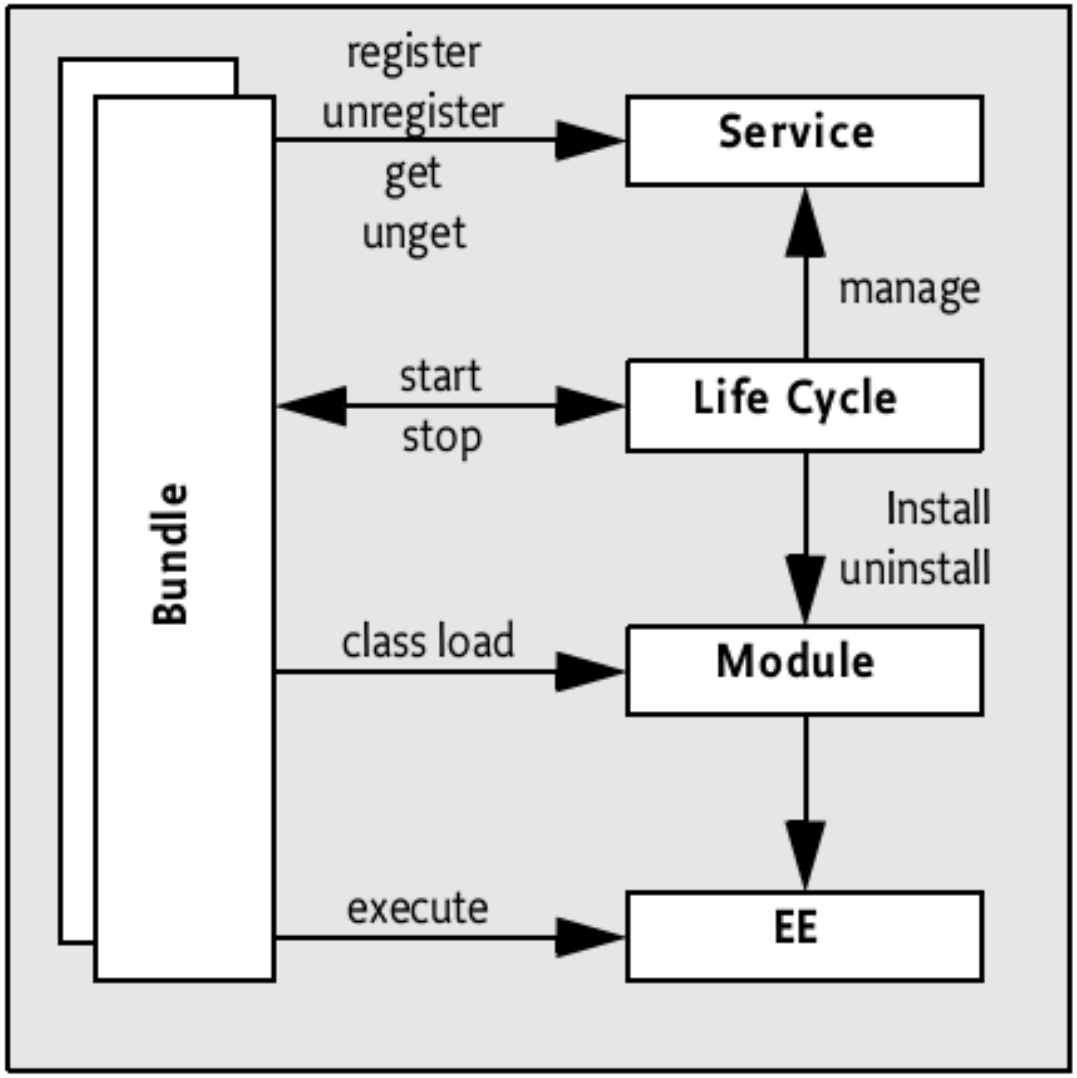
Globalcode

Life Cycle de um Bundle



Services


OSGi Framework
Resolved Bundle
Exported Package
Service
Interface
listen
Imported Package
get
register
Imported Package
Active Bundle
Service
Active Bundle


register
unregister
get
unget
Service
manage
start
stop
Life Cycle
Install
uninstall
Bundle
class load
Module
execute
EE

OSGI Services

- "This is similar to the **service-oriented architecture** made popular with **web services**. The key difference between web services and **OSGi services** is that web services always require some **transport layer**, which makes it **thousands times slower than OSGi** services that use direct method invocations."

*Retirado do site www.osgi.org

Bibliografia

- *OSGI core especification versão 5:* http://www.osgi.org/Download/Release5

- Modularidade com Java Module System & OSGi - vinicius senger http://www.thedevelopersconference.com.br/arquivos/TDC2008_OSGI.pdf

Demo

TDC2012
the developer's conference

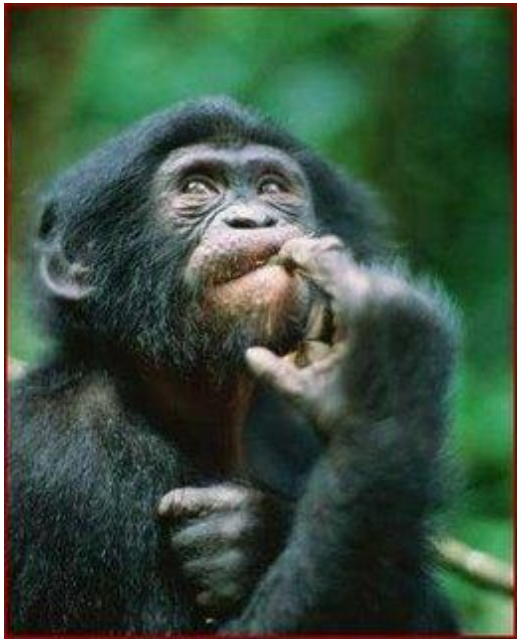
- OSGI + Maven + JavaFX = <3 <3
- Implementação Apache Felix
- https://github.com/filipeportes/ModuleFX

Globalcode

Dúvidas??

TDC2012
the developer's conference

- **Contato**
  - **@filipeportes**
  - **omeuefilipe@gmail.com**
  - https://github.com/filipeportes



Globalcode