

# Recuperação de Arquitetura de software

## Servidores de aplicação J2EE

Michelet del Carpio

# Introdução

- A seguir serão apresentadas as arquiteturas dos servidores de aplicação
  - GlassFish
    - <https://svn.dev.java.net/svn/glassfish-svn/trunk/v3>
  - JOnAS:
    - [svn://svn.forge.objectweb.org/svnroot/jonas/jonas/tags/JONAS\\_5\\_1\\_1](svn://svn.forge.objectweb.org/svnroot/jonas/jonas/tags/JONAS_5_1_1)
  - Jboss
    - [http://anonsvn.jboss.org/repos/jbossas/tags/JBoss\\_5\\_1\\_0\\_GA](http://anonsvn.jboss.org/repos/jbossas/tags/JBoss_5_1_0_GA)

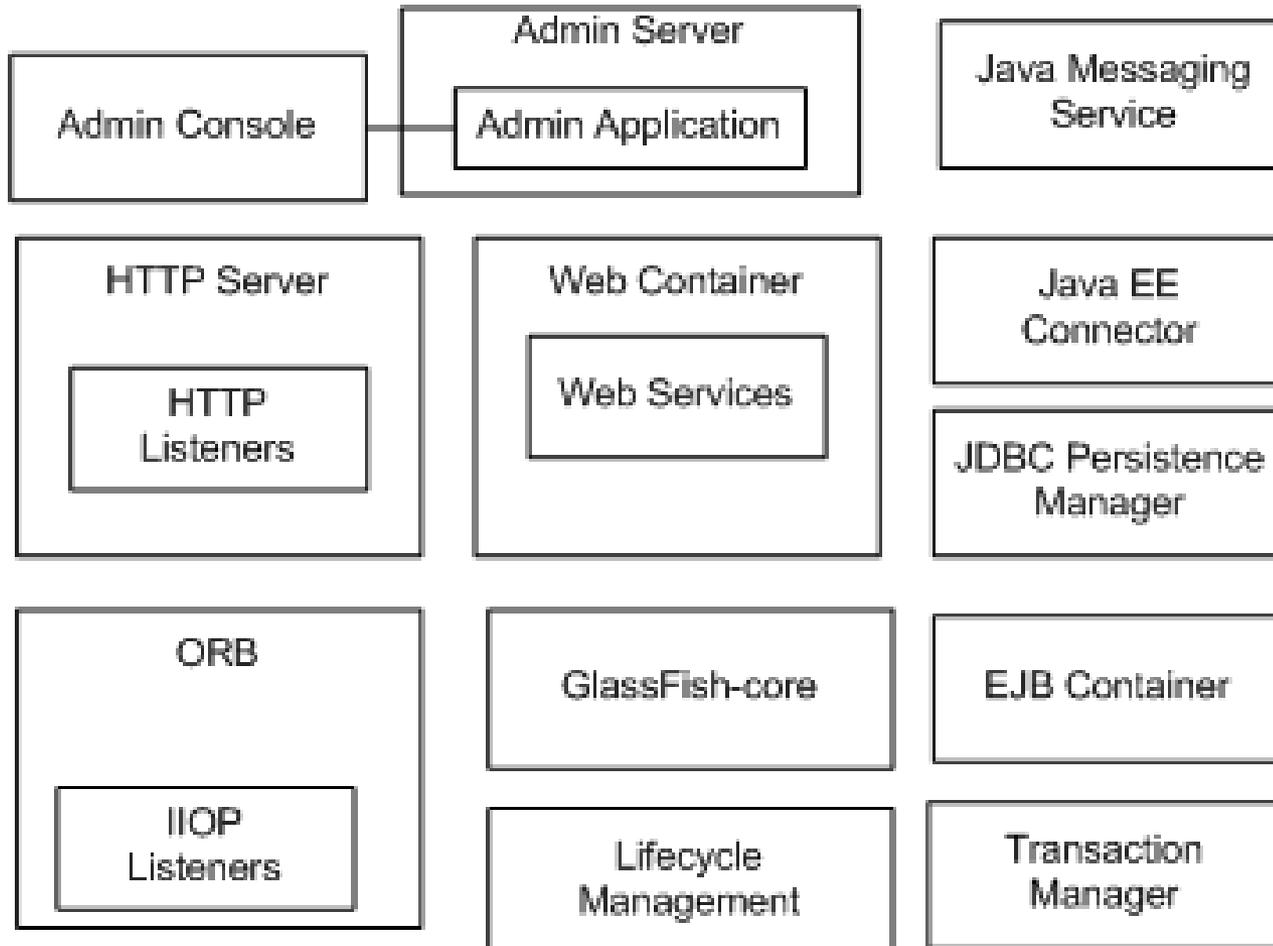
# Sobre o trabalho realizado

- Foi realizada uma análise dos modelos extraídos a partir do código fonte dos projetos avaliados.
- Os modelos extraídos estão disponíveis para acesso público no seguinte repositório:
- <https://modelos-sl.googlecode.com/svn/trunk/>
- Esta apresentação encontra-se também no seguinte *wiki*  
<http://ccsl.ime.usp.br/wiki/index.php/EntregaFinal>

# GlassFish v3

- Servidor de aplicação Open Source desenvolvido sob coordenação da SUN.
- Implementação completa da especificação J2EE 6.0
- Diferenciais:
  - Cada módulo é um sub-projeto
    - Gerenciamento dos módulos / sub-projetos (desenvolvimento )
  - Totalmente compatível com o Apache Felix, interoperável com o Eclipse Equinox
    - Implementação do HK2
  - Eclipse Link
    - Implementação JPA

# Principais Sub-projetos

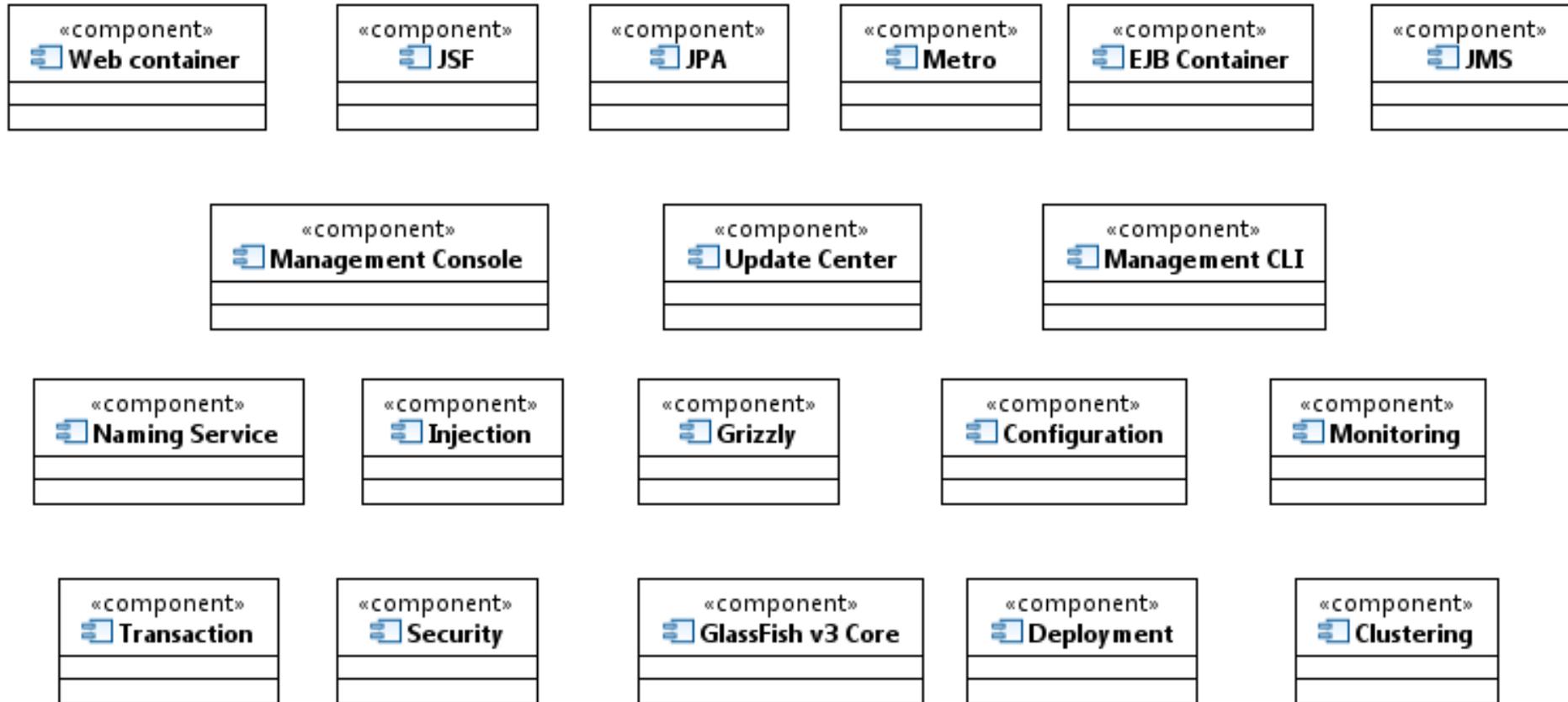


Principais Sub-Projetos do GlassFish

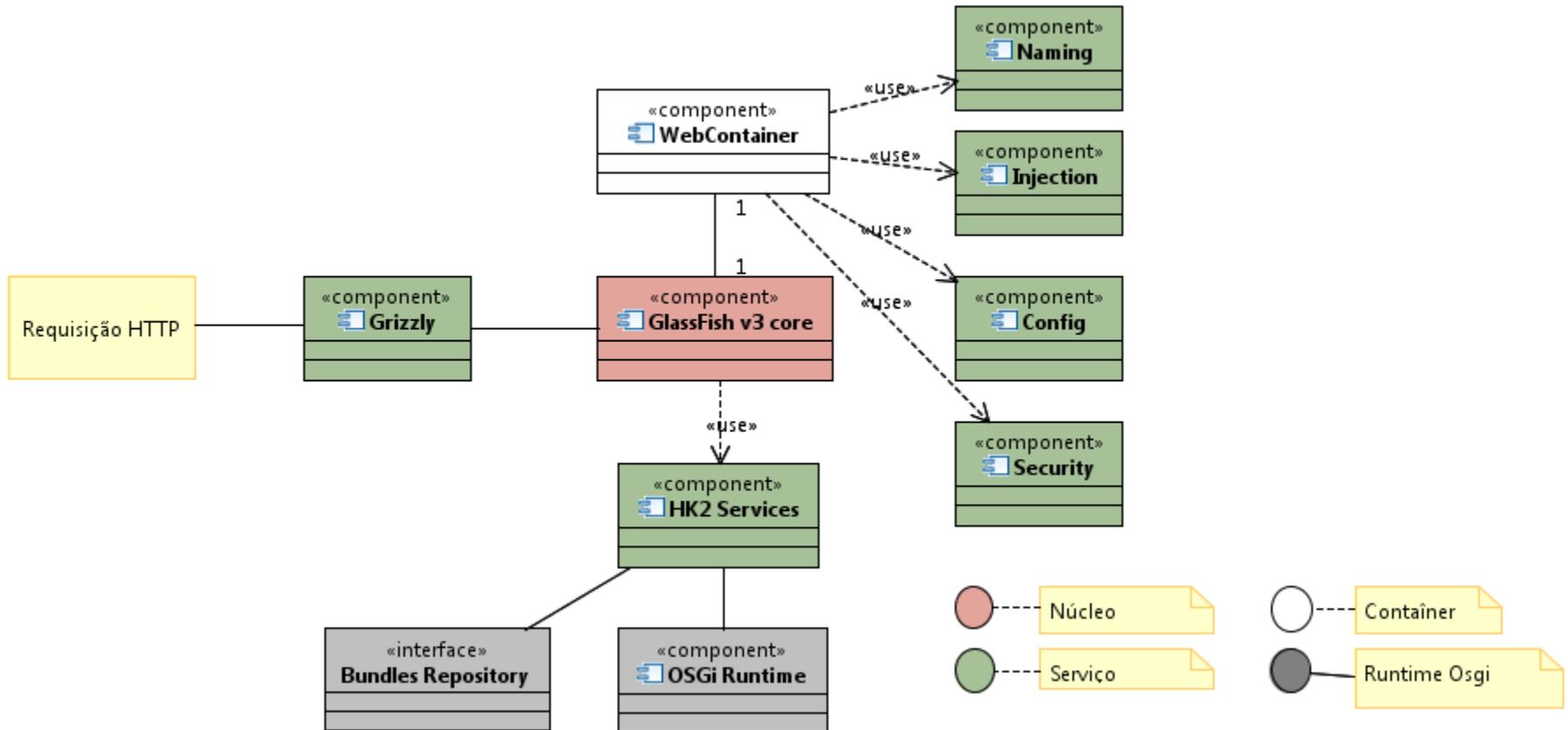
# GlassFish v3

- Elementos arquiteturais mais importantes
  - Núcleo
    - Inicialização/Desligamento
    - Leitura de parâmetros de configuração
    - Implantação de aplicações
  - Serviços / *Bundles*
    - Funcionalidades comuns entre os *containers* (p.ex: segurança, serviços de nomes)
    - Painel de administração
  - *Containers*
    - Responsáveis pelas aplicações dos usuários
    - Independentes entre si

# Componentes OSGi - GlassFish



# Tratamento de requisições



# JOnAS 5.1

- Servidor de aplicação Open Source desenvolvido pelo Consorcio OW2
- Diferenciais:
  - Eclipse Equinox
  - Kerneos, console de administração feito em Flex, fornecido pelo JAsMINE
  - Projeto CMI (*Clustering for RMI*)
  - Ferramentas de Implantação
    - DeployMe (implantação de topologias j2ee em servidores ‘físicos’)
    - JaDORt (ferramenta de implantação de aplicações j2ee e pacotes osgi )

# Jonas AS 5.1

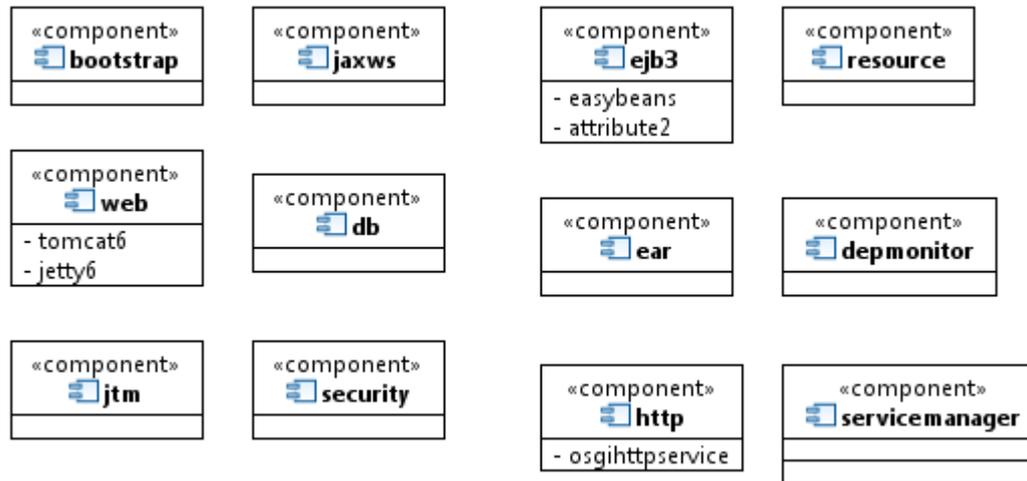
- Implementação da especificação J2EE 5.0
- Aderente parcialmente à especificação J2EE 6.0
  - Perfis de servidor (web profile, full profile)
  - EJB 3.1 via EasyBeans
  - Servlet 3.0

# Elementos arquiteturais do Jonas 5.1

- Em si, todos os elementos são serviços/componentes OSGi. Contudo, podem-se separar os serviços que fornecem as funcionalidades da especificação J2EE dos serviços que fornecem o suporte do ciclo de vida do próprio servidor de aplicações.
  - Serviços J2EE
    - ejb3, ejb2, httpservice (grizzly), etc.
  - Serviços que cuidam do ciclo de vida do servidor
    - jasmine,jadort, deployMe
    - servicemanager
-

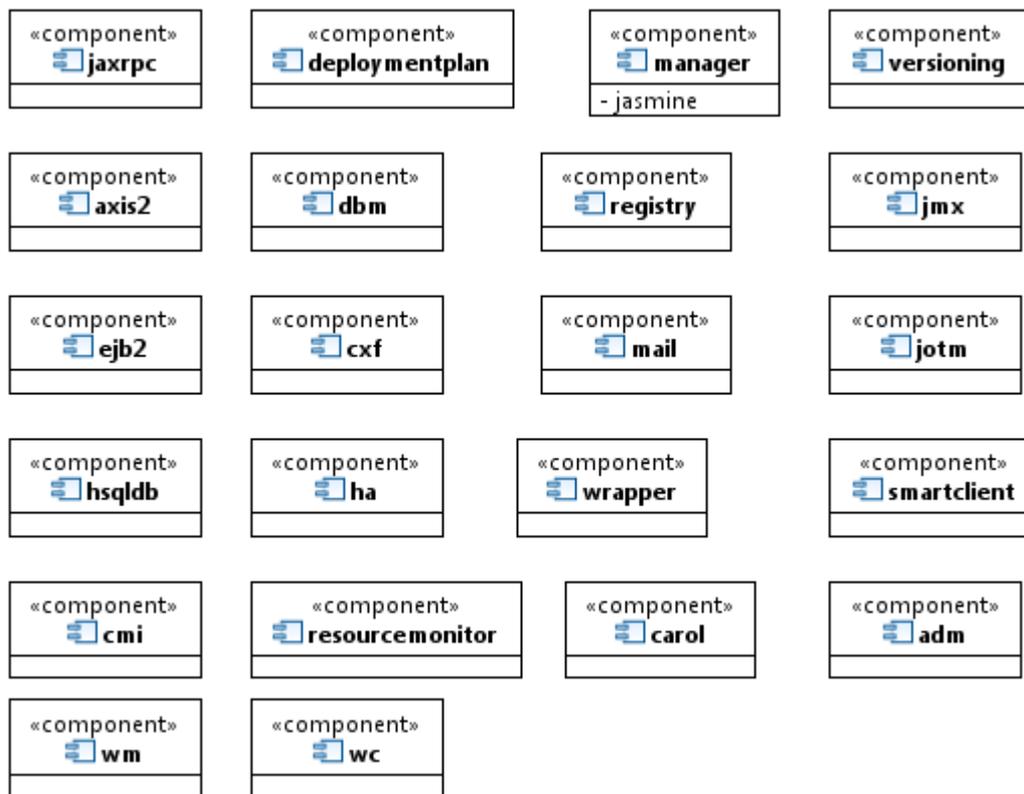
# Componentes OSGI - JOnAS

- Ainda, de acordo com a especificação J2EE 6.0 os servidores precisam fornecer perfis de execução. O JOnAS fornece um perfil mínimo ou essencial, composto pelos seguintes componentes:

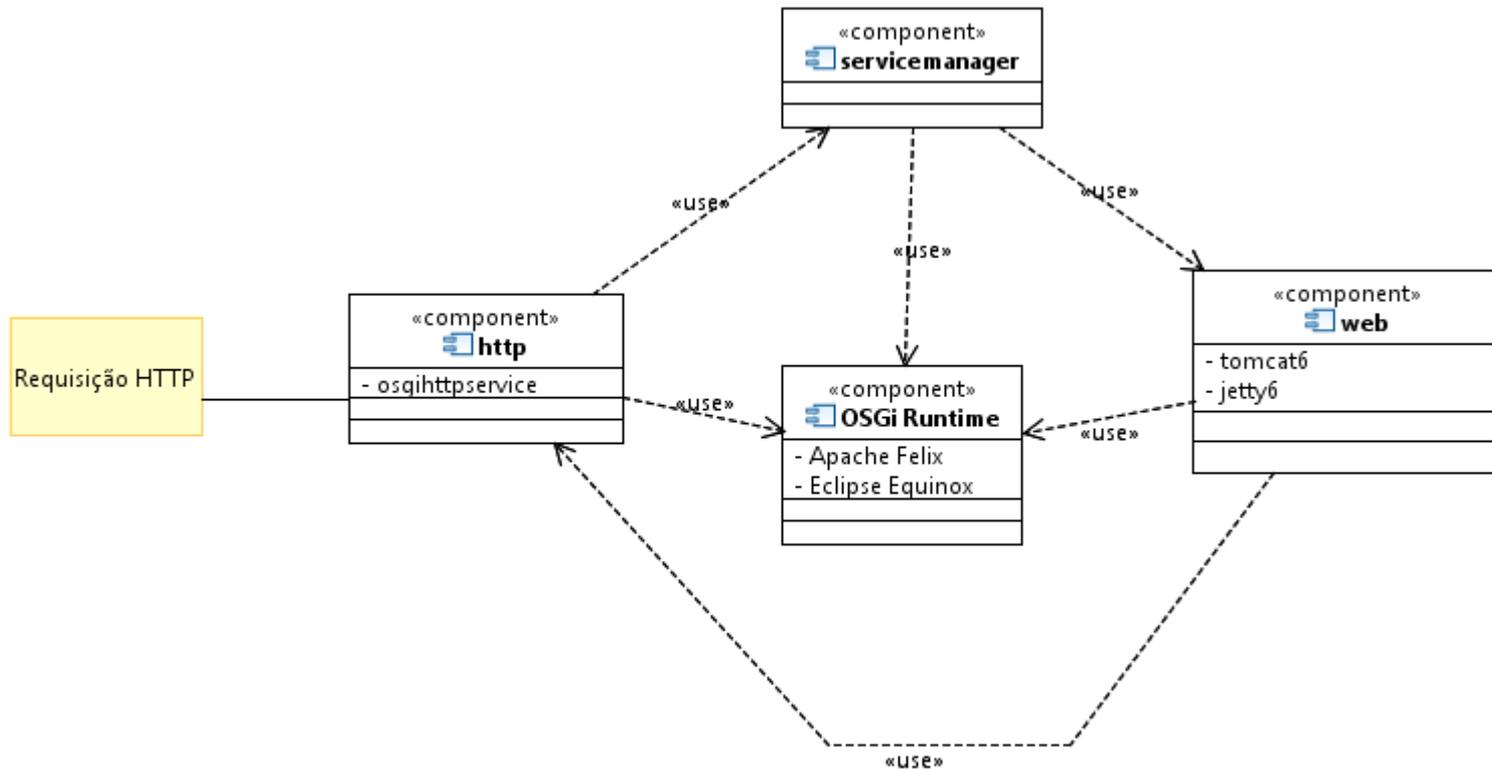


# Serviços/Componentes OSGi - JOnAS

Os seguintes serviços/componentes fazem parte do perfil On Demand e podem ser ativados através da configuração do servidor ou quando o serviço 'servicemanager' identificar dependências entre serviços



# Tratamento de Requisições



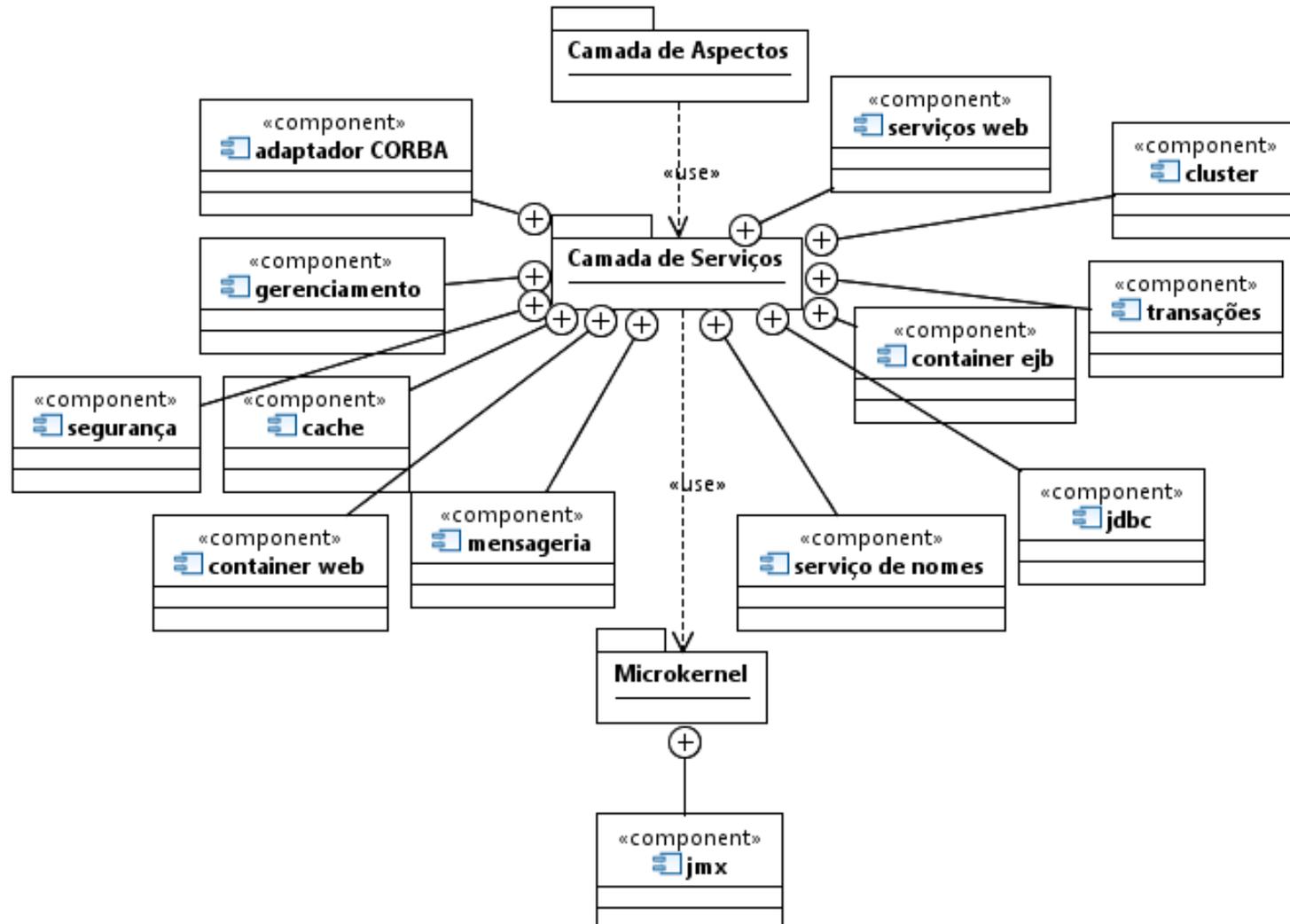
# JBoss 5.1

- Servidor de aplicação Open Source desenvolvido pela Red Hat
- Esta versão suporta a especificação J2EE 5
- Diferenciais:
  - Não utiliza OSGi na sua execução
  - Fornece suporte à orientação a Aspectos

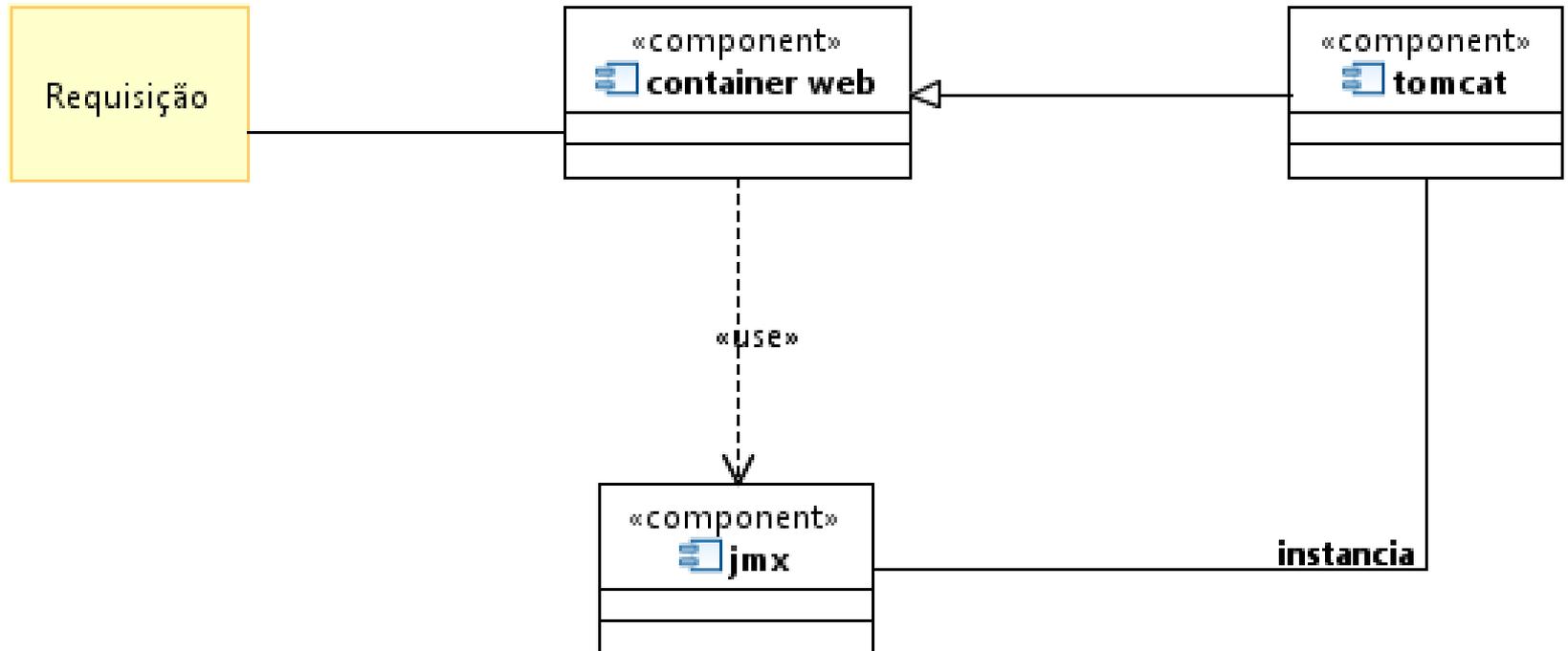
# Elementos arquiteturais do JBoss 5.1

- Os elementos do servidor de aplicação são componentes organizados em três camadas lógicas.
  - Camada de microkernel
  - Camada de serviços
  - Camada de suporte a aspectos

# Componentes JBoss v5.1



# Tratamento de requisições



# Comparação entre as arquiteturas

- Os servidores estudados primam por serem modulares e extensíveis. O objetivo deles é ter a capacidade de agregar outros componentes (ou serviços) e atender à especificação J2EE.
- Os servidores Glassfish v3 e JOnAS 5.1 são executados em cima de um framework OSGi, a diferença do JBoss 5.1 que consiste em uma aplicação Java tradicional que precisa um máquina virtual para sua execução.
- No caso do GlassFish v3, ele precisa de dois módulos necessários para seu funcionamento. O primeiro módulo é o HK2 que se comunica diretamente com o framework OSGi e tem a finalidade de carregar as interfaces de comunicação com os serviços que serão utilizados pelo servidor. O módulo que cuida da seguinte etapa, ou seja, do levantamento desses serviços é o glassfish-core.
- No caso do JOnAS 5.1 não é necessário este módulo de comunicação direta. Os serviços podem alocados de forma horizontal em cima do framework OSGi, sempre e quando atendam um determinado perfil de execução.
- No caso do Jboss 5.1, o estilo de arquitetura de execução é o micro-kernel e sua organização apresenta uma arquitetura em camadas. Ao utilizar o JMX como micro-núcleo, os serviços são incluídos como 'extensões' e não afetam o funcionamento independente de componentes.

# Conclusões

- No domínio de aplicação dos servidores de aplicação, as arquiteturas apresentaram características bem parecidas no ponto de vista conceitual. Já no ponto de vista tecnológico ou concreto as diferenças surgiram
- A recuperação de informações arquiteturais sobre o funcionamento do sistema foi uma das atividades mais difíceis de realizar.
- A documentação, em forma de diagramas de seqüência, por exemplo, é quase inexistente.