

## *EJB simples e descomplicado, na prática*

Kleber Xavier – Arquiteto Senior / Globalcode

[kleber@globalcode.com.br](mailto:kleber@globalcode.com.br)

Vinicius Senger – Arquiteto Senior / Globalcode

[vinicius@globalcode.com.br](mailto:vinicius@globalcode.com.br)

@vsenger

Autores Livro Gratuito: “33 patterns com Java”

<http://www.slideshare.net/vsenger/33-design-patterns-com-java>

# Agenda

---

- Introdução
- Por que EJB
- Tipos de EJB

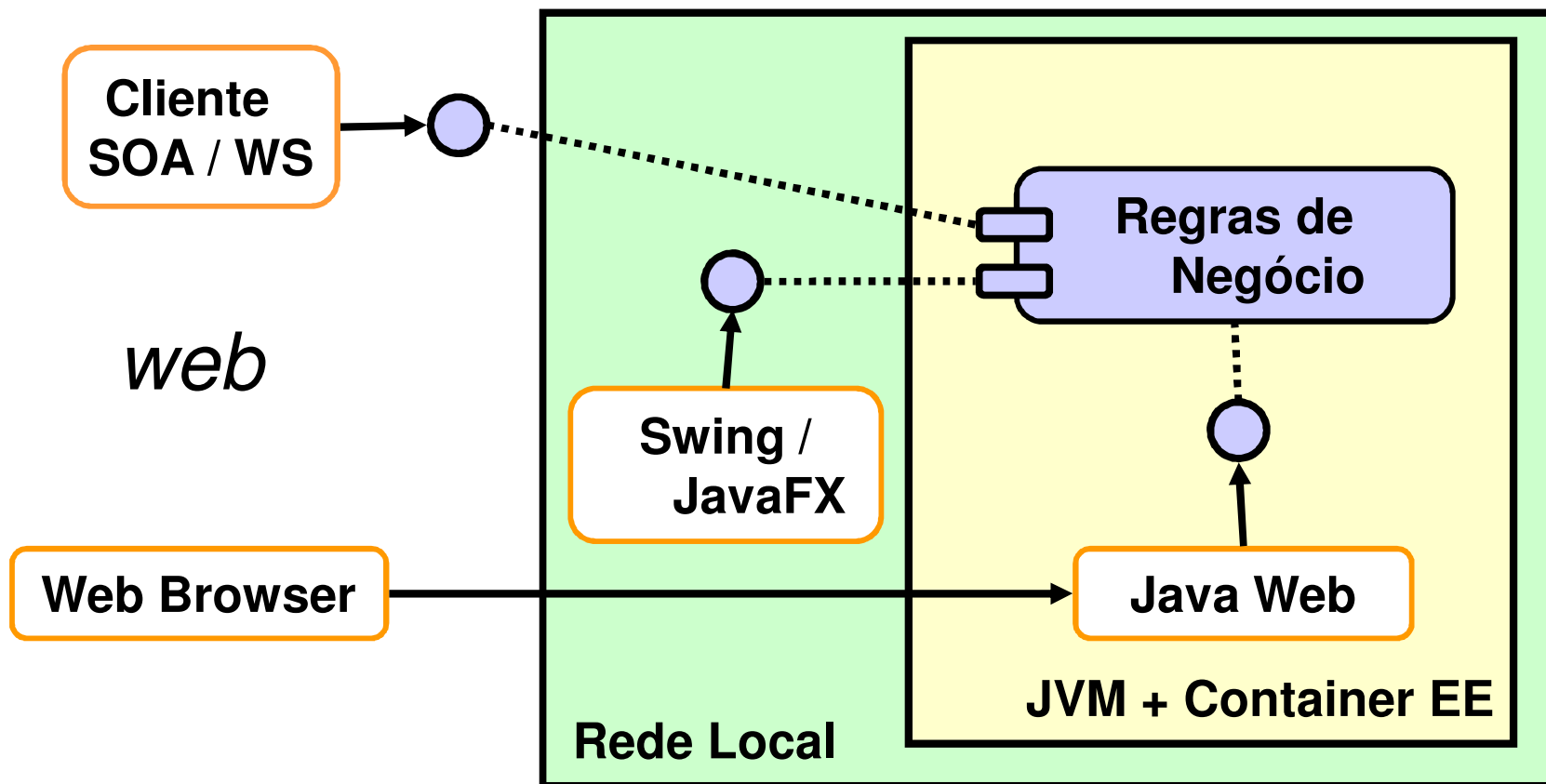
- EJB é um modelo de componentes com foco em arquiteturas multi-camadas (baseado em princípios básicos de rede)
  - RMI (Remote Method Invocation)
  - JNDI (Java Naming and Directory Interface)
- Responsável pela regra de negócio das aplicações Web e não Web
- EJB é um modelo de alto nível para objetos distribuídos / computação distribuída

# Por que EJB ?

- Componentes de Negócio Distribuídos
  - Vários tipos de aplicações clientes podem utilizar as mesmas regras de negócio;
- Infra-estrutura do Container de EJBs
  - Segurança
  - Controle Transacional
  - Gerenciamento de recursos
  - Agendamento de tarefas
- Padrão de Mercado para Componentes de Negócio

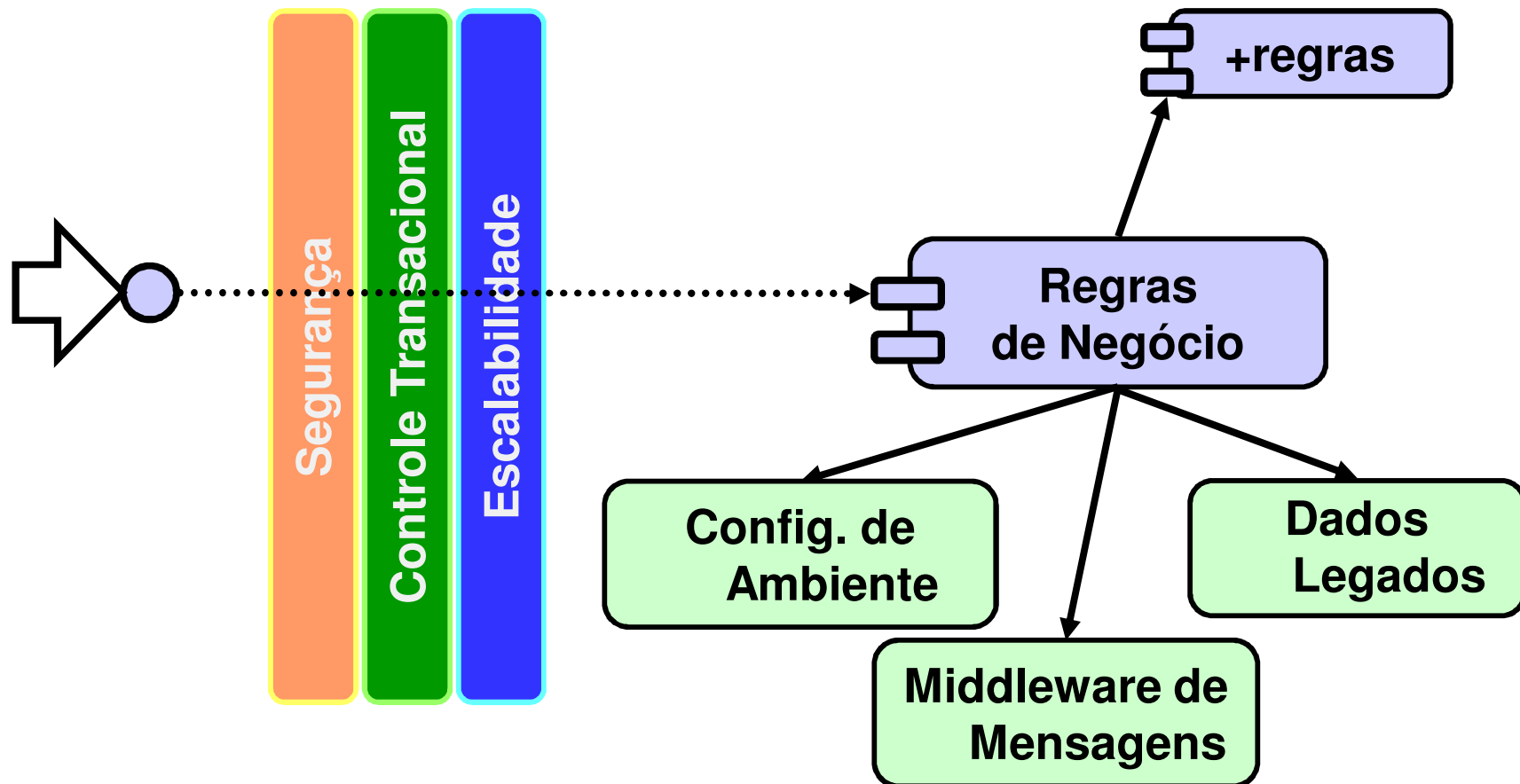
# Por que EJB ?

- Componentes de Negócio Distribuídos



# Por que EJB ?

- Infra-estrutura do Container de EJBs



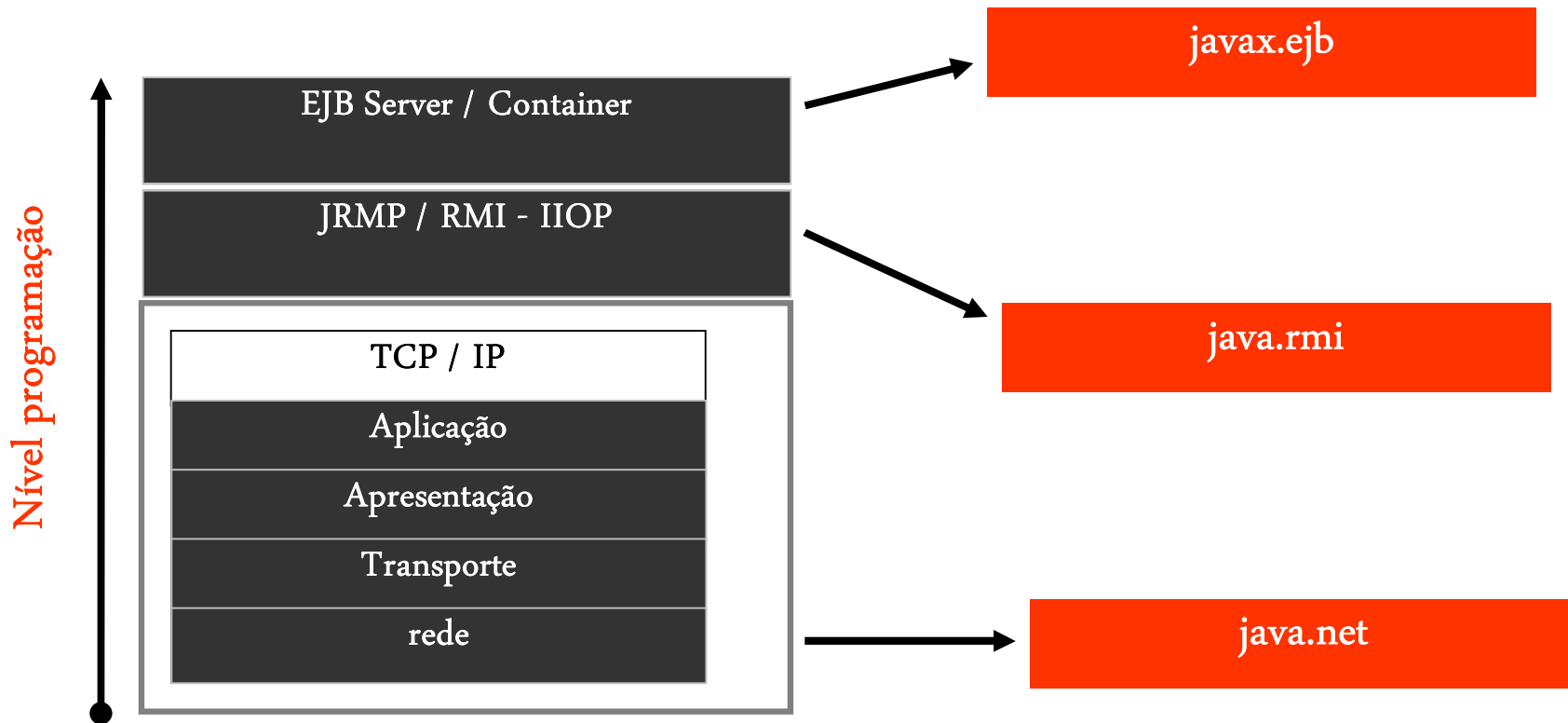
# Por que EJB ?

---

- Padrão de Mercado para Componentes de Negócio:
  - Oracle ⇒ Glassfish e WebLogic;
  - Red Hat ⇒ JBoss AS;
  - IBM ⇒ WebSphere;
  - Apache ⇒ Geronimo;
  - etc ...



# Remote Method Invocation



- Java EE 1.4
  - Session Bean (Stateless/Stateful)
  - Entity Bean (BMP/CMP)
  - Message Driven Bean
- Java EE 5
  - Session Bean (Stateless/Stateful)
  - Message Driven Bean
- OBS.: JPA (API de persistência que substitui Entity Bean, mas não é um EJB)

- Tipos:
  - Session beans (Stateless e Stateful)
  - Message-driven beans
- Interface de negócio
  - Local
  - Remota
  - Service Endpoint (web services)
- Classe de implementação do componente
  - POJO somente com métodos do componente
- Uso simplificado

- Interface de negócio
  - Anotada com o tipo da interface:

@Local	( <i>javax.ejb</i> )
@Remote	( <i>javax.ejb</i> )
@WebService (JAX-WS)	( <i>javax.jws</i> )

```
import javax.ejb.Remote;
@Remote
public interface CarrinhoRemote {
    public List getProductos();
    public void addProduto(int codigoProduto, int quantidade);
}
```



- Classe de implementação:

```
import br.com.globalcode.model.Item;
import java.util.*;
import javax.ejb.Stateful;

@Stateful
public class CarrinhoBean implements CarrinhoRemote {
    List<Item> itens = new ArrayList<Item>();

    public void addProduto(int codigo, int qtd) {
        itens.add(new Item(codigo, qtd);
    }

    public List getProdutos() {
        return itens;
    }
}
```

- Empacotamento / Deployment:
  - É um jar com a classes dentro do seus respectivos pacotes;
  - Contém um diretório META-INF com os deployment descriptors;

- Uso:
  - Componentes são injetados dentro dos clientes
  - Anotações definem a dependência com o componente



- Usando o EJB na versão 3.x

Injeção de dependência

```
//...
import javax.ejb.EJB;

public class MostrarCarrinhoServlet extends HttpServlet {
    @EJB
    CarrinhoRemote carrinho;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        List l = carrinho.getProdutos();
    }
}
```

# Enterprise Java Beans 3.1

- JSR 318
  - Melhora ainda mais EJBs 3.0
  - Menor curva de aprendizado
  - Maior simplicidade para aplicações menos complexas
  - Novos recursos promissores
- Principais mudanças
  - Novo componente Singleton Bean
  - Melhoria no agendamento de tarefas
  - Alternativa simplificada a MDBs
  - Simplicidade no uso em aplicações Web / Desktop

- Subconjunto da EJB 3.1 API
- Não define API adicional
- Permite servidores mais leves para cenários simplificados
  - Nem toda aplicação utiliza 100% dos recursos EJB

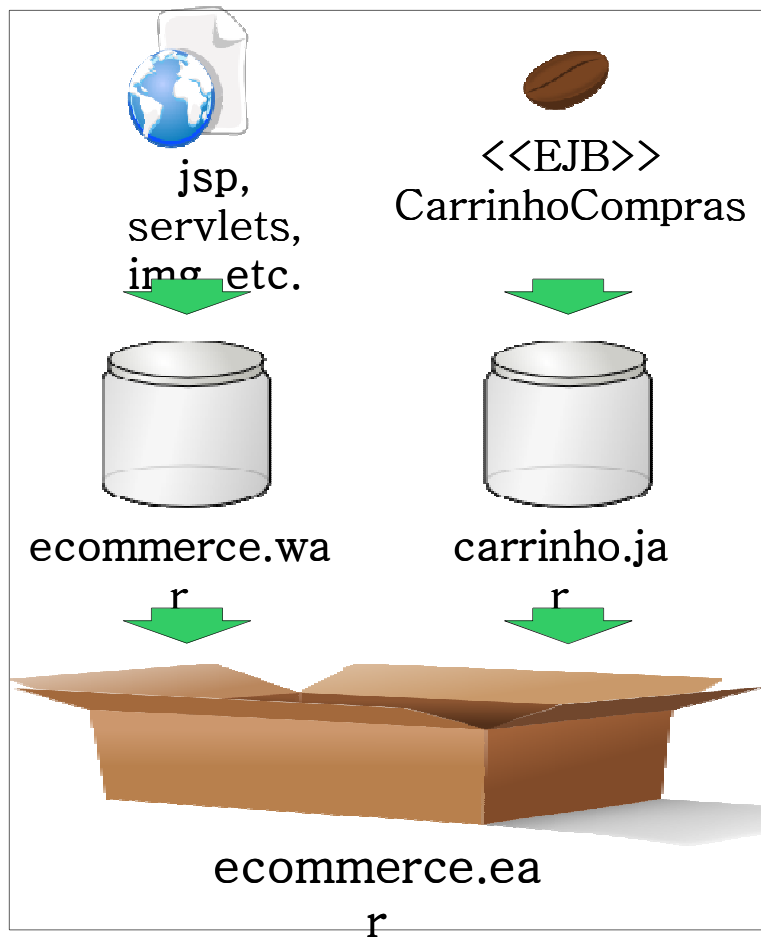
	<b>EJB LITE</b>	<b>EJB FULL</b>
Interface Local/NoInterface	SIM	SIM
Interface Remota		SIM
Stateless, Stateful, Singleton	SIM	SIM
JPA 2.0	SIM	SIM
Chamadas Assíncronas		SIM
MDBs		SIM
Web Services EndPoints		SIM
Compatibilidade EJB 2.x		SIM
Interoperabilidade RMI-IIOP		SIM
Timer Service		SIM
Entity Beans		SIM
CMT/BMT	SIM	SIM
Segurança Declarativa	SIM	SIM
Interceptors	SIM	SIM

# Deploy de EJBs na camada Web

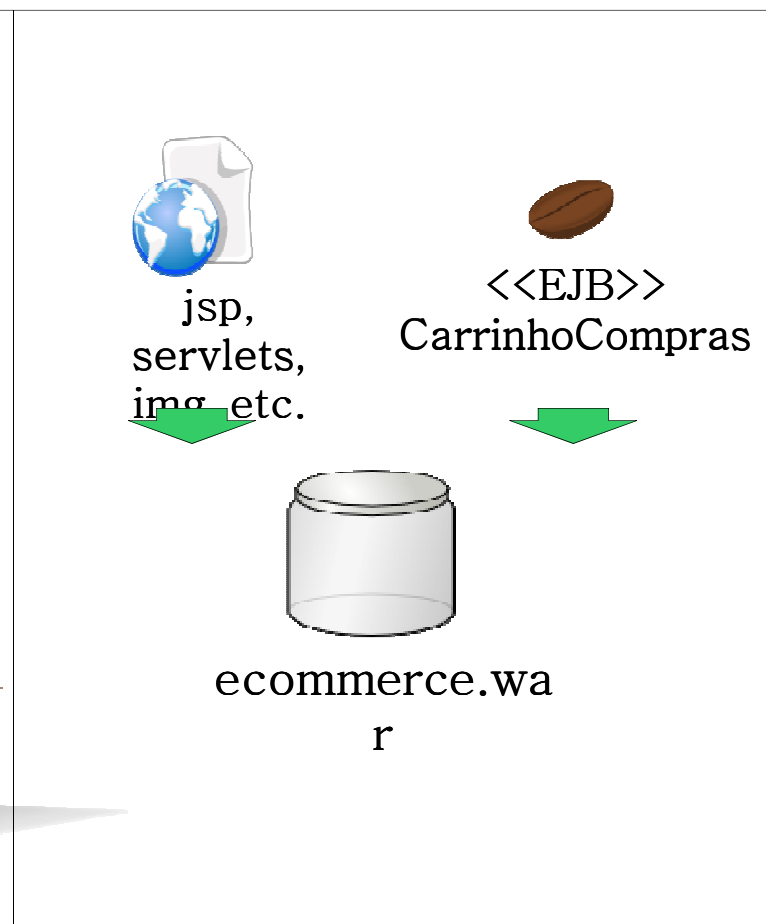
- Facilita o uso de EJBs com aplicações baseadas na Web
- Menor curva de aprendizado
- Simplicidade
- Agora é possível empacotar EJBs diretamente no arquivo .war
- Em aplicações n-camadas recomenda-se o uso de módulos EJBs, como na 3.0.

# Deploy de EJBs na camada Web

JAVA EE 5



JAVA EE 6



- Java EE 5 e 6 apresentaram muitas melhorias em relação às versões anteriores, facilitando bastante o desenvolvimento de aplicativos;
- Alguns recursos não especificados são complementados por frameworks de mercado como Spring
- Parte dos recursos que faltavam foram acrescentados no Java EE 6
- Java EE 7 terá recursos para Cloud:  
@MultiTenancy

# Mais Informações

---

<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

<http://docs.oracle.com/javaee/5/tutorial/doc/>

<http://docs.oracle.com/javaee/6/tutorial/doc/>