

# Curso Desenvolvedor Java



# EJB

# Servidores de Aplicação

## Java EE 5 Compatible Implementations



Apache Geronimo-2.1.4



WebLogic Server v10.0



Fujitsu  
Interstage Application Server  
Enterprise Edition 9.2



IBM WASCE 2.0



IBM WebSphere Application  
Server v7



JBoss  
JBossAS 5.0.0



Apusic Application  
Server (v5.0)



NEC  
WebOTX 8.1



Oracle Application  
Server 11



OW2 JOnAS 5.1



SAP NetWeaver 7.1



Sun GlassFish Enterprise  
Server 9.1



TmaxSoft JEUS 6

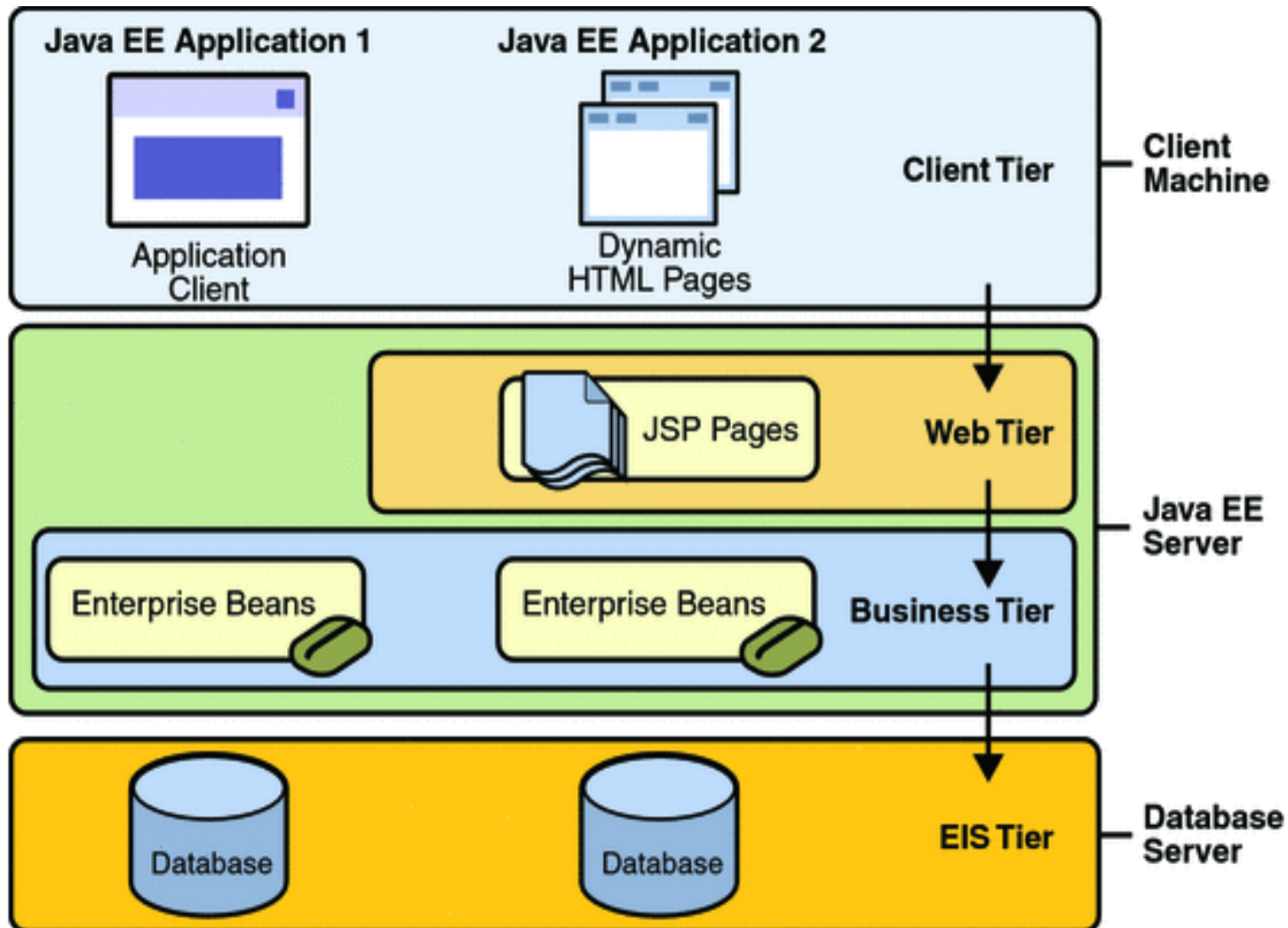


TongTech Co., Ltd  
TongWeb Application  
Server 5.0

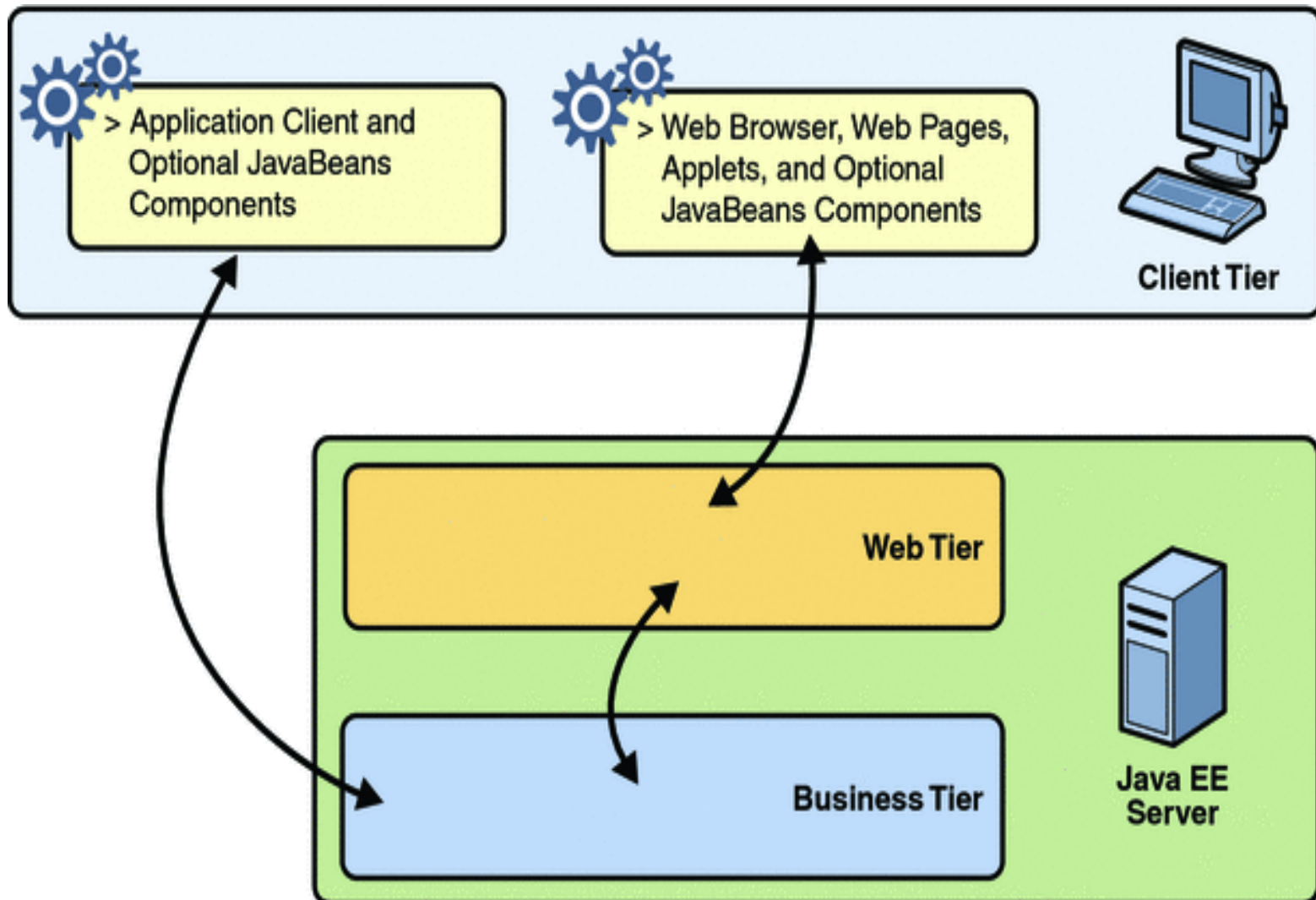


GlassFish Application  
Server v2

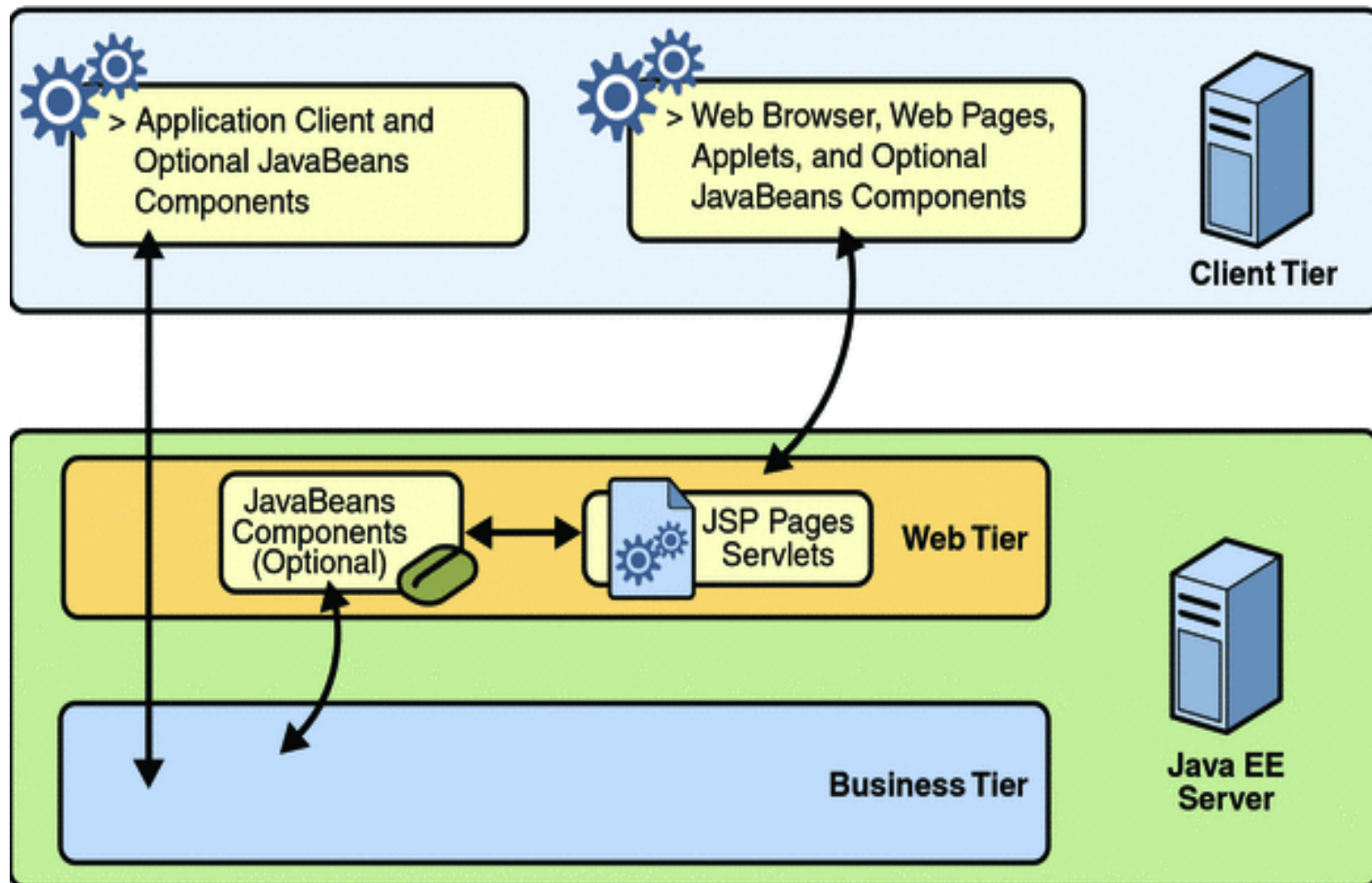
# Aplicações Multicamadas



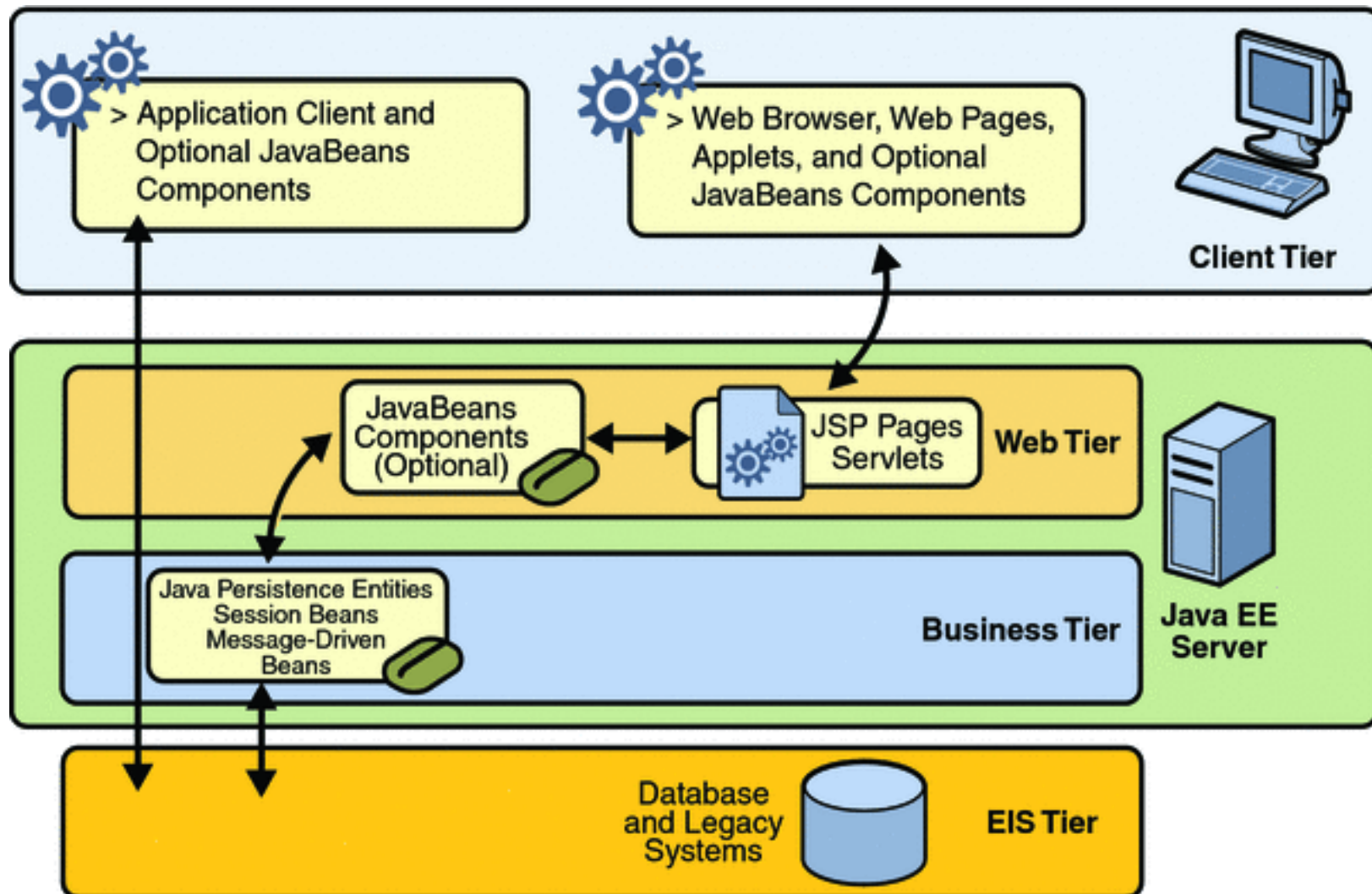
# Aplicação Desktop



# Aplicação Web

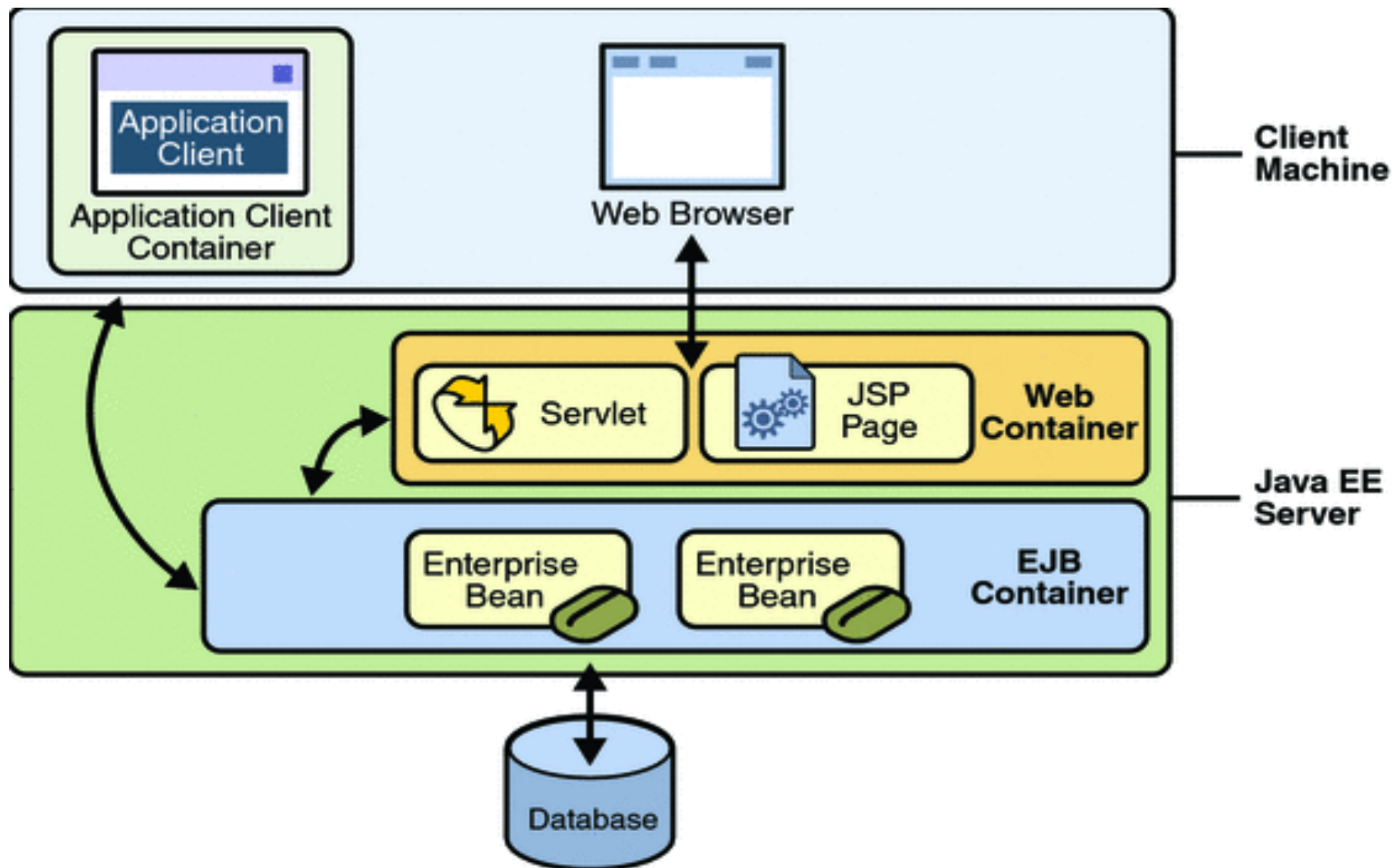


# Acesso aos Dados

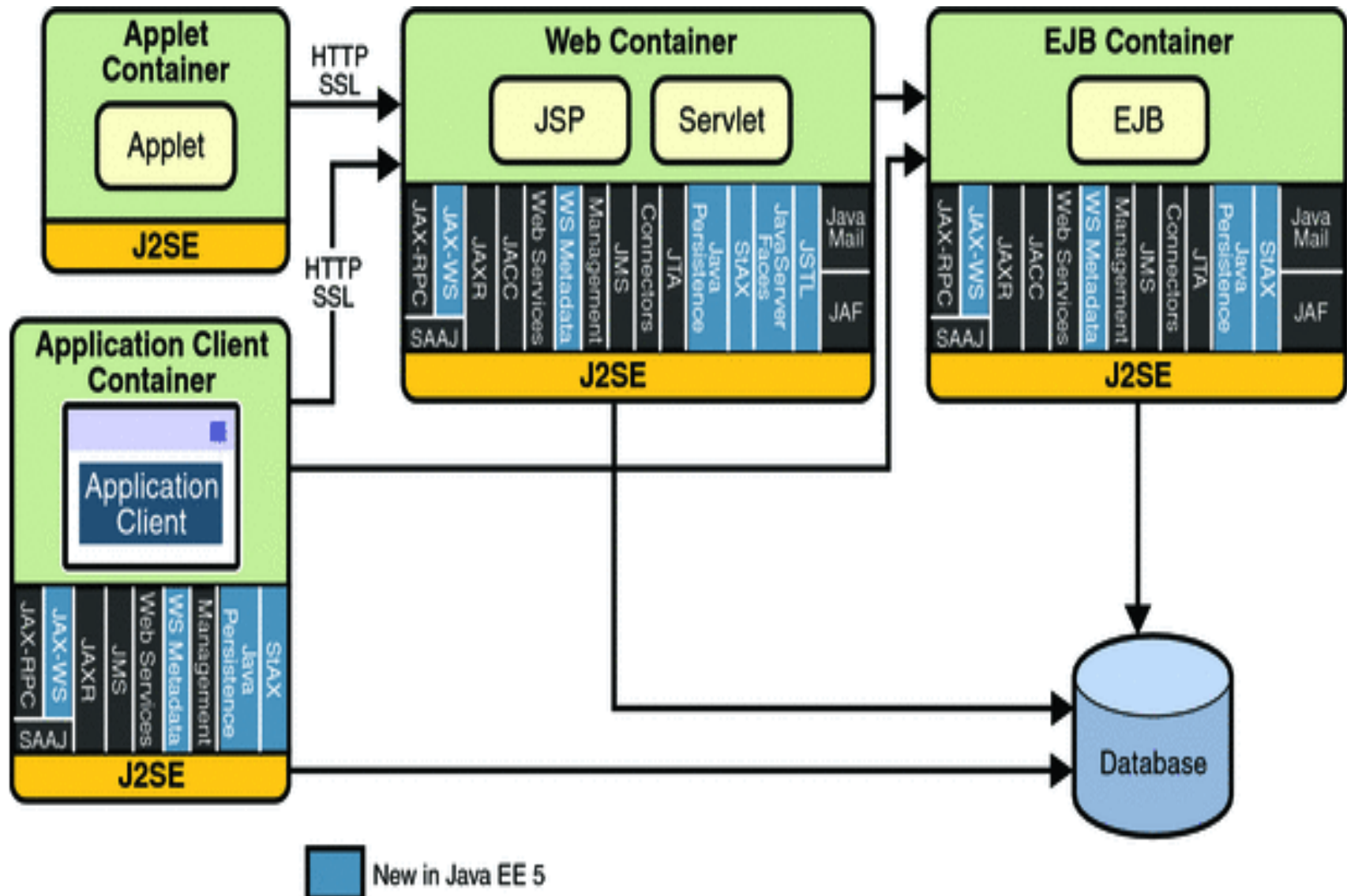




# Os Containeres

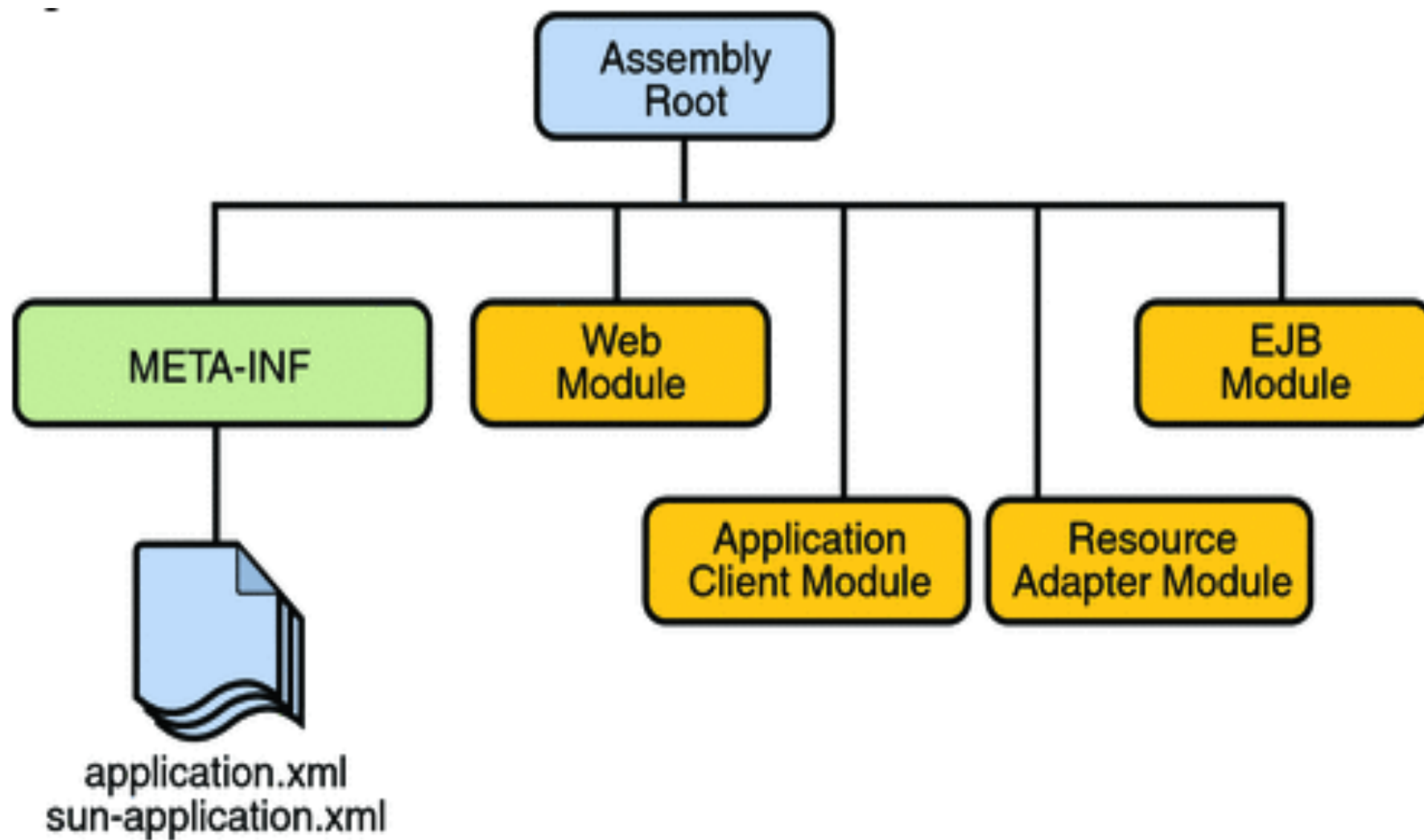


# Visão da API Java EE 5

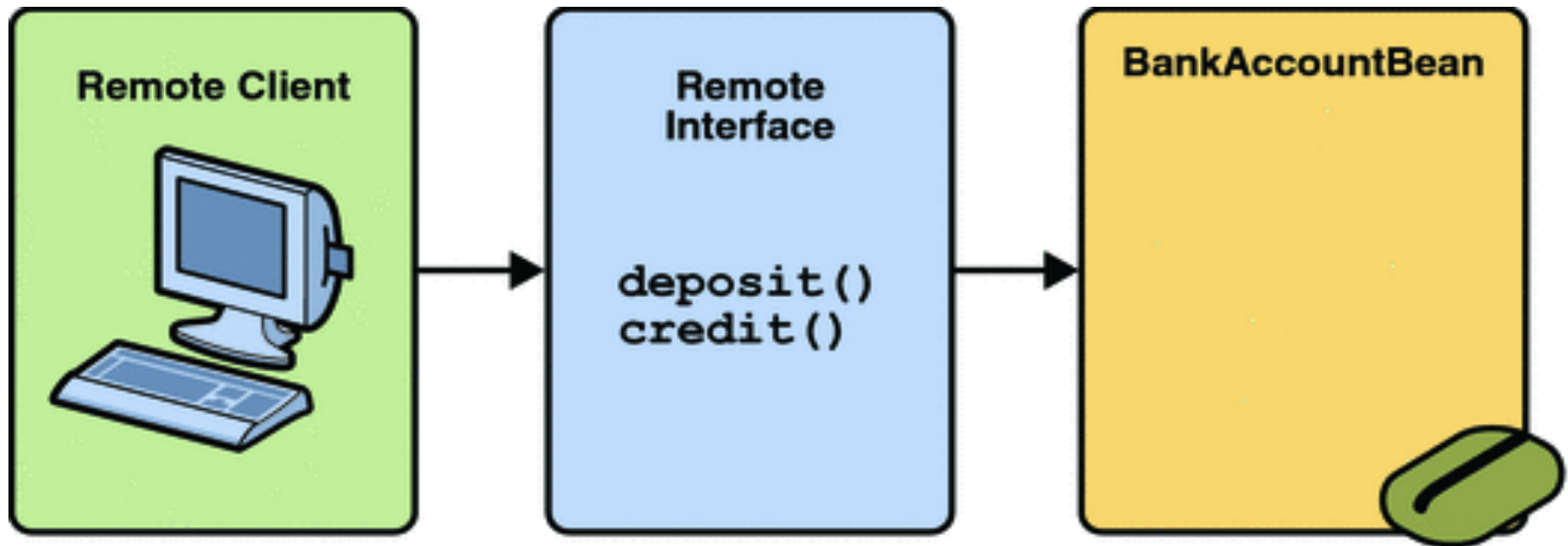




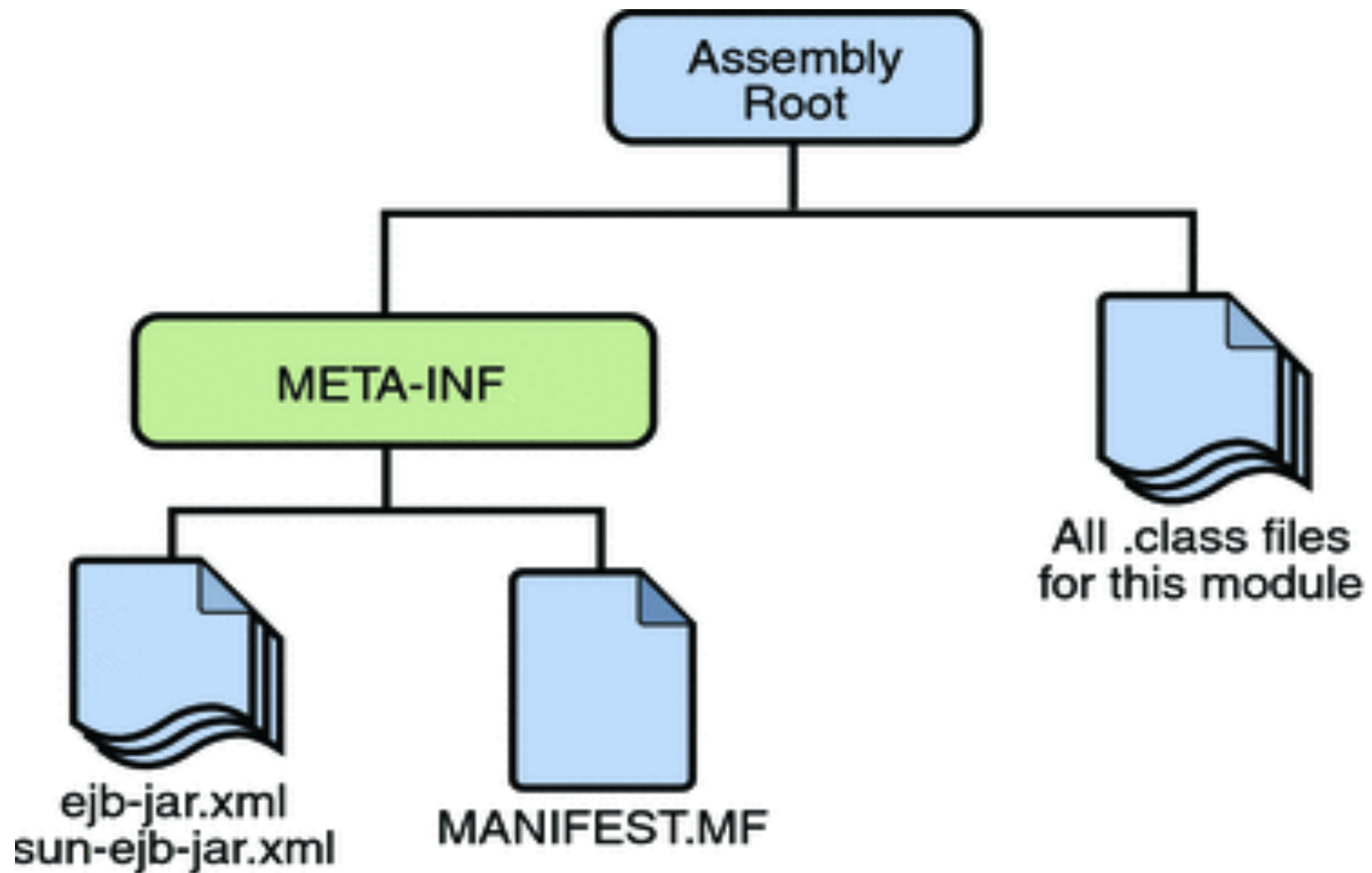
# Estrutura de um EAR



# A Interface ao Serviço Remoto **SENAI**



# A Estrutura de um EJB JAR



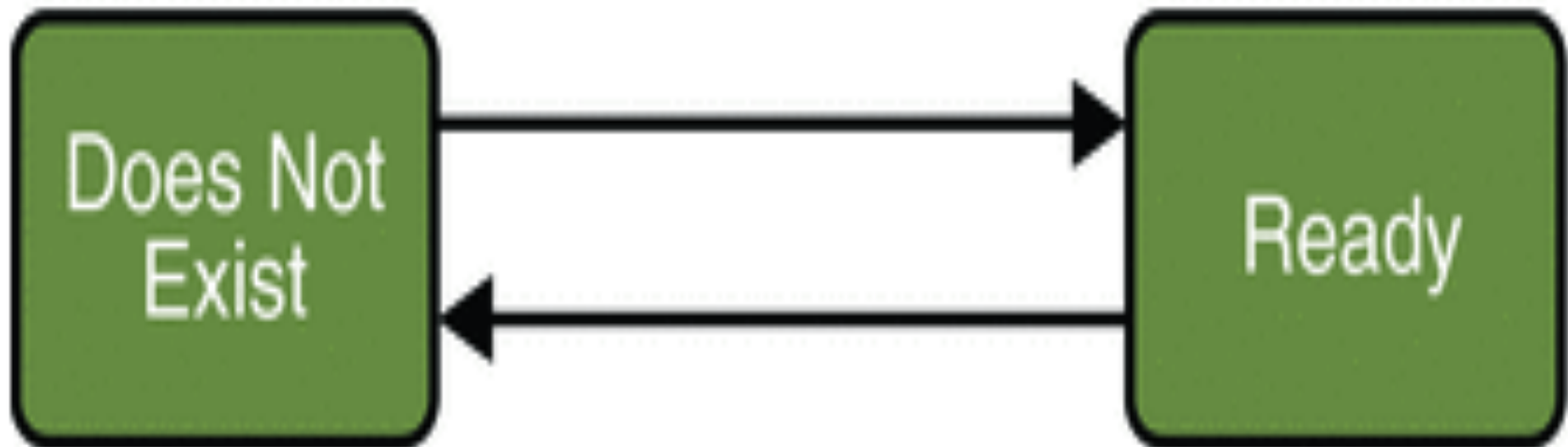
# Ciclo de Vida de um EJB Statefull

1. Create
2. Dependency injection, if any
3. PostConstruct callback, if any
4. Init method, or ejbCreate<METHOD>, if any



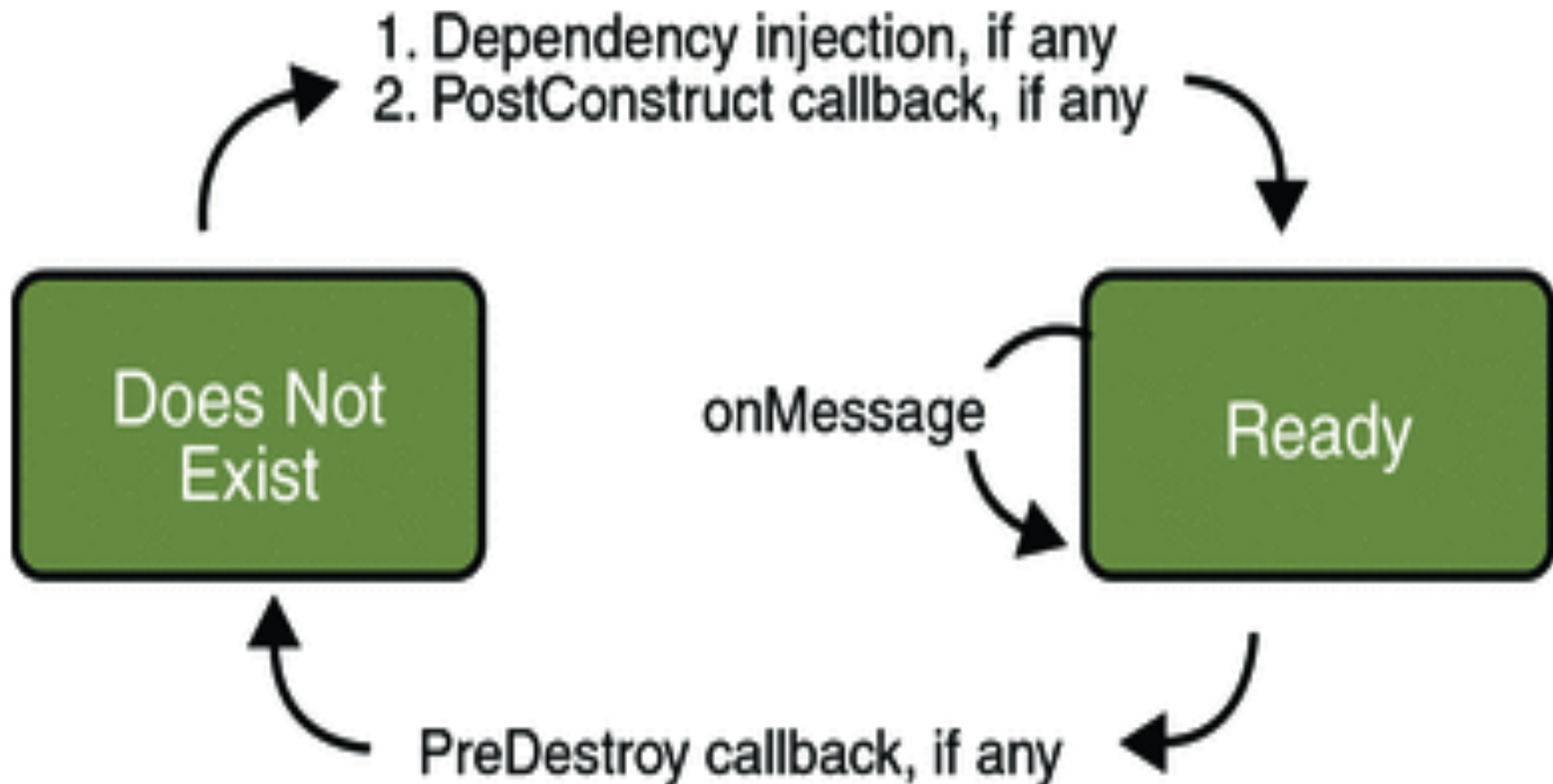
# Ciclo de Vida de um EJB Stateless

1. Dependency injection, if any
2. PostConstruct callbacks, if any



PreDestroy callbacks, if any

# Ciclo de Vida de um EJB Message-Driven





# A Declaração da Interface de Negócio



```
import java.math.BigDecimal;  
import javax.ejb.Remote;
```

```
@Remote  
public interface Converter {  
    public BigDecimal dollarToYen(BigDecimal dollars);  
    public BigDecimal yenToEuro(BigDecimal yen);  
}
```

# O EJB

```
import java.math.BigDecimal;
import javax.ejb.*;

@Stateless
public class ConverterBean implements Converter {
    private BigDecimal yenRate = new BigDecimal("115.3100");
    private BigDecimal euroRate = new BigDecimal("0.0071");

    public BigDecimal dollarToYen(BigDecimal dollars) {
        BigDecimal result = dollars.multiply(yenRate);
        return result.setScale(2, BigDecimal.ROUND_UP);
    }

    public BigDecimal yenToEuro(BigDecimal yen) {
        BigDecimal result = yen.multiply(euroRate);
        return result.setScale(2, BigDecimal.ROUND_UP);
    }
}
```

# Uma Aplicação Cliente



```
import java.math.BigDecimal;
import javax.ejb.EJB;

public class ConverterClient {
    @EJB
    private static Converter converter;

    public ConverterClient(String[] args) {
    }

    public static void main(String[] args) {
        ConverterClient client = new ConverterClient(args);
        client.doConversion();
    }

    public void doConversion() {
        try {
            BigDecimal param = new BigDecimal("100.00");
            BigDecimal yenAmount = converter.dollarToYen(param);
        }
    }
}
```

# Uma Aplicação Web

```
<%@ page import="converter.ejb.Converter,  
    java.math.*, javax.naming.*"%>  
  
<%!  
    private Converter converter = null;  
    public void jspInit() {  
        try {  
            InitialContext ic = new InitialContext();  
            converter = (Converter) ic.lookup(Converter.class.getName());  
        } catch (Exception ex) {  
            System.out.println("Couldn't create converter bean." +  
                ex.getMessage());  
        }  
    }  
  
    public void jspDestroy() {  
        converter = null;  
    }  
%>
```

# Um Message Driven Bean

