



Especializada em treinamentos Java e J2EE

- Mais de 800 alunos treinados
- Mais de 4.000 em palestras e mini-cursos
- Instrutores certificados na plataforma Java;
- Em Florianópolis, na V.Office – f. (48) 224-8580

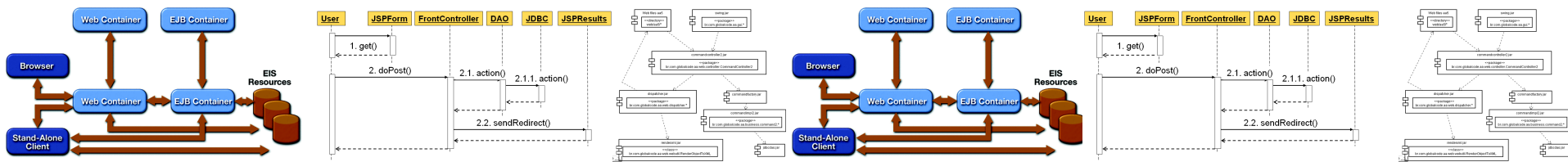
0800-704-5282

www.globalcode.com.br

JAREF

J2EE Architecture Research and Education Framework

From simple JSP to full J2EE in 10 steps...



Projeto Reconhecido pelo Sun JELC Java Education & Learning



Java Education & Learning (JELC) Advisory Board Meeting

Daniel Brookshier (September 10, 2004 2:18PM PT)

URL: <http://community.java.net/edu-jelc/>

» [Permalink](#)

This week I will be in Palo Alto meeting with Scott McNealy, John Gage, and [education leaders from all over the world](#). We will be talking about java.net's Education and Learning Community (JELC). The tasks will be to organize the advisory board for the first time, talk about open source in education, and to work towards creating a stronger community.

Featured Projects

One of my tasks is showing off what we have so far. We have 142 projects (including topic areas) and 975 members. That's quite a lot for a community that was relaunched just a few months ago. The projects range from student research to university infrastructure. Some are for research and other projects are real working software. Here are three of the projects that I will be showing off at the meeting:

Globalcode (globalcode.dev.java.net) is a dual-purpose project that combines software for educational institutions with computer science. With the global team based in Brazil, the project's mission is to teach Java 2 Platform Enterprise Edition (J2EETM) by example and to benchmark code to show how different strategies of architecture design compare. Their reference application is designed for a school's management of courses, classes, teachers, and students. With more than 80 members dedicated to its progress, Globalcode is already one of the largest projects in the JELC.

Agenda

- **Introdução a Arquiteturas**
- **JAREF: J2EE Architecture Research and Education Foundation**
- **10 Alternativas de Arquiteturas J2EE**
- **Futuro do projeto JAREF**
- **Conclusões**

Motivação

- Quais componentes, patterns, frameworks e melhores práticas J2EE devo adotar na minha empresa?
 - Quando usar EJB's?
 - Devo usar TODOS os patterns J2EE?
 - Qual o impacto de performance ao adotar M.V.C.?
 - Conseguiremos manter a solução após a entrega da fábrica de software? Quanto custará?

Agenda

- **Introdução a Arquiteturas**
- JAREF: J2EE Architecture Research and Education Foundation
- 10 Alternativas de Arquiteturas J2EE
- Futuro do projeto JAREF
- Conclusões

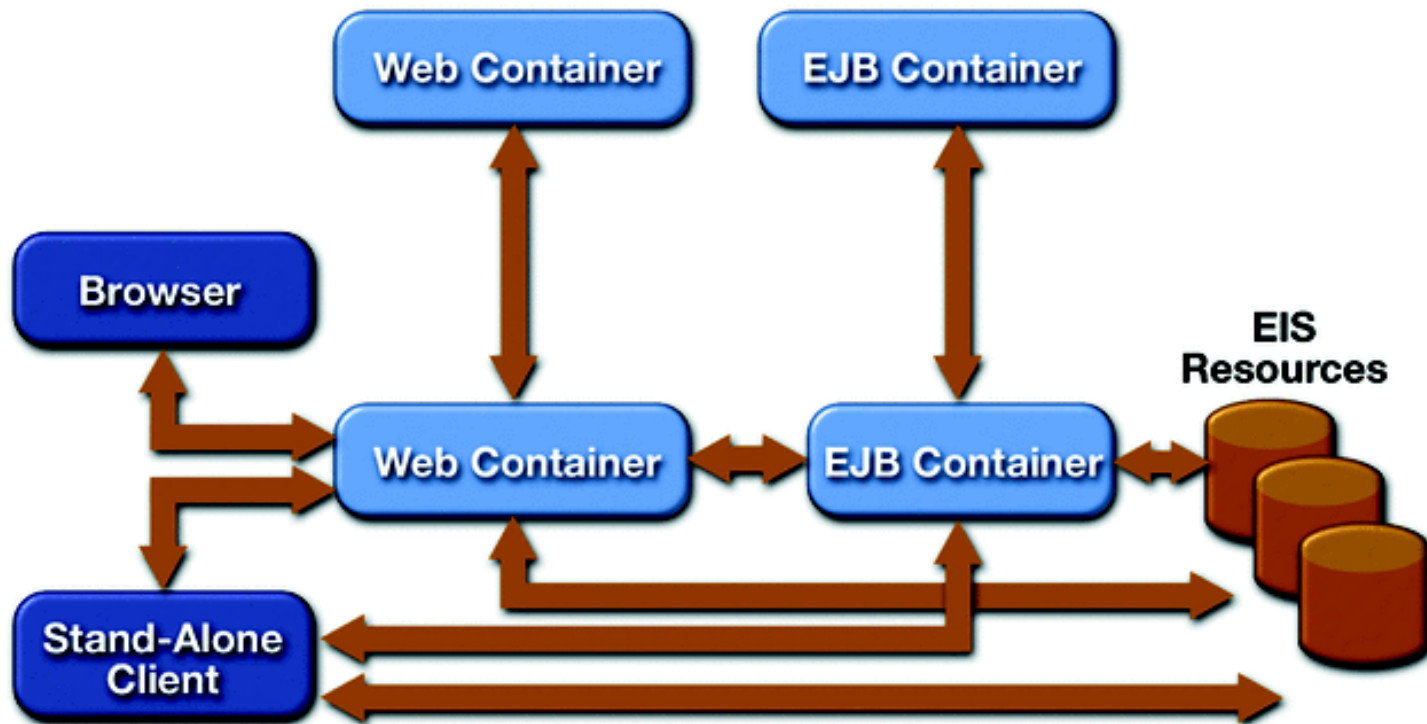
Introdução a Arquiteturas

- J2EE representa uma plataforma flexível para o desenvolvimento de softwares em múltiplas camadas de QUALQUER porte;
- Oferece diversos recursos: Servlets, JSP's, Tag Library, Filter, segurança, WebService, transações, persistência, pool de conexões e objetos, entre outros;
- Podemos aplicar design-patterns, melhores práticas e convenções para aumentarmos a qualidade da solução;

Introdução a Arquiteturas

- A definição de uma arquitetura J2EE deve reunir:
 - Tipos de componentes utilizados: Servlet, JSP, Custom Tag, Servlet Filter, EJB, Entity CMP, BMP, assíncrono, transações etc.;
 - Design-patterns e frameworks adotados;
 - Melhores práticas, convenções e métricas de desenvolvimento;
 - Diagramas UML de referência: use-cases, seqüência, classes, componentes e deployment;

Introdução a Arquiteturas



Introdução a Arquiteturas

- Uma **arquitetura** deve **atender** aos requisitos **funcionais** e **não-funcionais** do negócio:
 - Deve atender todas as **funcionalidades esperadas** pelos **usuários**;
 - Deve ser compatível com o **know-how da equipe** de desenvolvimento (interna ou fábrica externa);
 - **Requisitos não-funcionais** devem ser considerados: usuários simultaneos ou requisições por segundo, flexibilidade, segurança, extensibilidade, capacidade de crescimento, disponibilidade desejada, entre outros;

Introdução a Arquiteturas

- Considere os seguintes aspectos de ambiente ao decidir elementos de uma arquitetura J2EE:
 - Capacidade técnica da equipe;
 - Necessidade de integração com demais plataformas;
 - Nível de segurança;
 - “Elasticidade” do negócio;
 - Ferramentas disponíveis;
 - Prazo;
 - Budget;
- “More, is not necessarily better...”, Leonardo Galvão.

Introdução a Arquiteturas

- Melhores práticas de arquitetura J2EE:
 - Utilize UML intensamente: cada diagrama tem seu valor, fuja dos diagramas mudos;
 - Não deixe a paixão pela tecnologia da equipe tornar sua empresa um centro de testes de API's e frameworks;
 - Sempre que possível, efetue provas de conceitos: estatísticas não existem!
 - É mais difícil manter um padrão de arquitetura, do que definir um, cuide dos processos e metodologia!

Agenda

- Introdução a Arquiteturas
- **JAREF: J2EE Architecture Research and Education Foundation**
- 10 Alternativas de Arquiteturas J2EE
- Futuro do projeto JAREF
- Conclusões

JAREF

J2EE Architecture Research and Education Foundation

- É um **aplicativo** simples **J2EE** desenvolvido com mais de **10 diferentes arquiteturas**;
- Ilustra o uso dos principais **componentes, patterns e práticas J2EE** através de **arquiteturas progressivas**;
- Baseado em **2 anos de experiência** em consultorias em projetos para **bancos, seguradoras e fábricas de softwares**;
- Funcionalidades simples para manter o foco na arquitetura;

JAREF

Objetivos do projeto JAREF

- Educacional
 - O JAREF é a base para o ensino de arquiteturas nos treinamentos Globalcode;
 - Estará disponível para download gratuito com ampla documentação para auto-estudo de J2EE e certificação de arquiteto;
 - Uma forma prática e progressiva para o entendimento completo da plataforma J2EE;

JAREF

Objetivos do projeto JAREF

- Pesquisa
 - Efetuamos testes em laboratórios para analisar a performance de cada arquitetura;
 - Simulamos situações reais e obtemos dados concretos para provar matematicamente os benefícios de uma arquitetura;
 - Resultados de pesquisa de performance das arquiteturas publicado no JavaOne 2004;

JAREF

Equipe

- Mantido pelo Architecture Research Group, um grupo de pesquisas patrocinado pela Globalcode;
- Foi publicado no JavaOne 2004, um paper com testes de performance das arquiteturas referenciadas;
- Equipe composta por 7 profissionais de diferentes áreas, dedicados ao estudo, pesquisa e desenvolvimento de arquiteturas J2EE;

JAREF

Mais sobre o projeto JAREF

- Disponível para download: www.globalcode.com.br/jaref
- Trabalha por padrão com Jboss 3.2, MySQL e Eclipse;
(testado no IBM Websphere, BEA Weblogic e Sun Application Server)
- Catálogo atual conta com 6 arquiteturas Web e 4 com EJBs;
- Documentação completa UML e vídeo-aulas de cada arquitetura em novembro/2004;

JAREF

DEMO

JAREF - Catálogo de arquiteturas 1.0

1	JSP + Direct JDBC Access (inside JSPs) + Locator + Pooling
2	#1 + Data Access Object design-pattern
3	#2 + M.V.C. básico com Front Controller
4	#3 + Command & Command Factory design-patterns
5	#4 + View Dispatcher e clientes Swing web-based e Macromedia Flash
6	#5 + View Helper & Intercepting Filter design-patterns
7	#6 + Session Beans + Business Delegate
8	#7 + Entity Beans (BMP) + Session Façade
9	#8, Replacing BMP for CMP
10	#9 + Message-Driven Beans

JAREF

Documentação das Arquiteturas

- Cada arquitetura é documentada com:
 - Ficha técnica: quando usar, quando evitar, recursos e know-how necessário, fator de custo de mão-de-obra, fator de risco, perfil de demanda e cenários típicos;
 - Diagramas UML: use-case, classes, componentes, seqüência e deployment;
 - Vejamos exemplo das documentações;

JAREF Doc

JAREF Architecture No. 3 – Ficha técnica

Recursos utilizados: Connection pool, Service Locator, JSP's com JavaBeans, Data Access Object com JDBC, M.V.C. básico e Front-controller;

Quando utilizar esta arquitetura

- Aplicativos de médio porte em empresas com conhecimento intermediário sobre a plataforma J2EE;

Quando evitar esta arquitetura

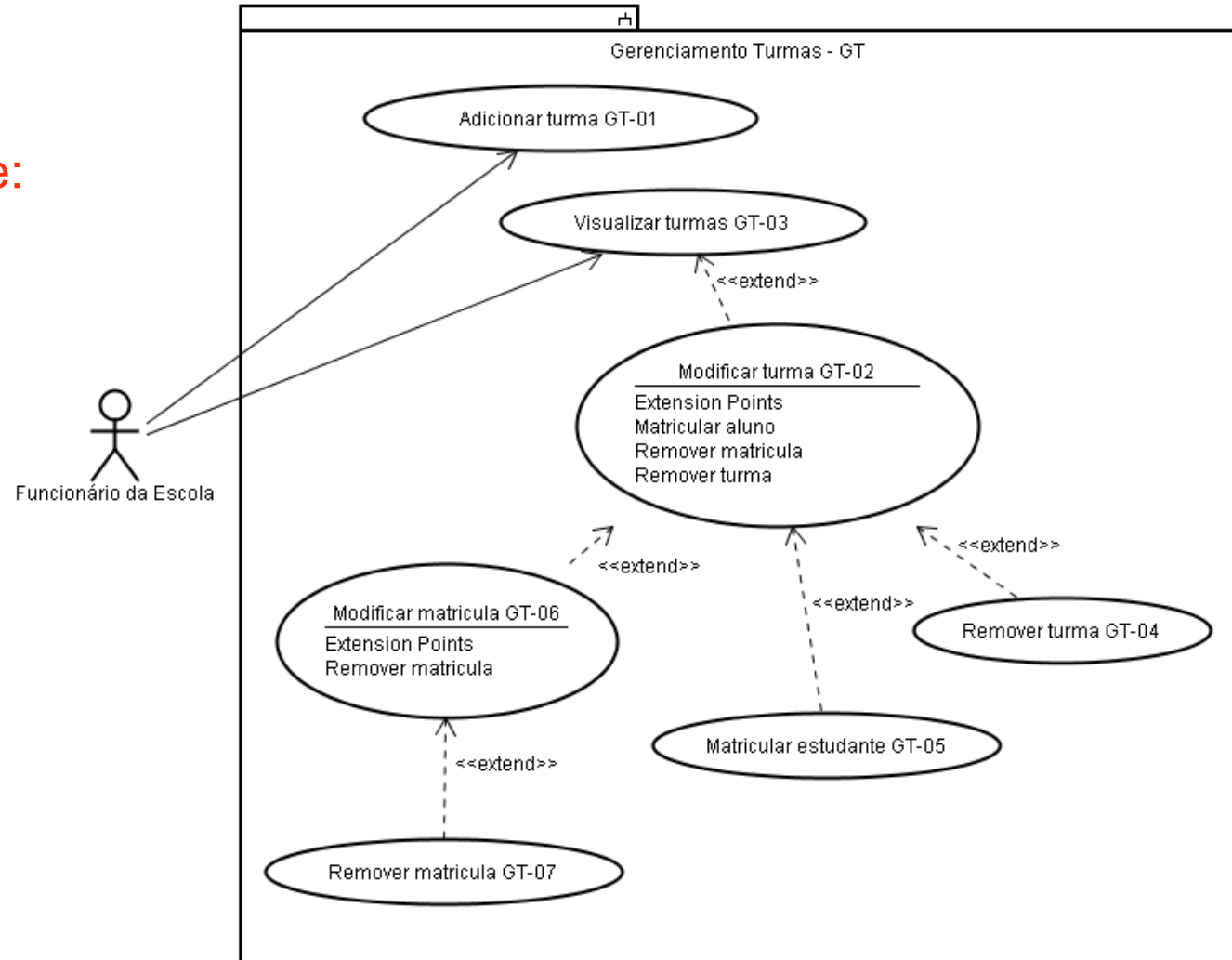
- Aplicativos com grande quantidade de processamento de dados em Java;
- Aplicativos web de médio e grande porte para longo prazo;
- Aplicativos e web-sites com grande número de funcionalidades;

Risco Técnico: médio
Fator custo: 1.15

Perfil de demanda: 1 até 4.000 usuários
Fator de qualidade: 1.3

JAREF – Exemplo da documentação

Diagramas use-case:
além do básico!



JAREF – Exemplo da documentação

Documento Use-case

1 Informações Gerais

Caso de uso	Visualizar <u>memberships</u>	Código	GM-03
Subsistema	Gerenciamento <u>Membership</u>	Código	GM

1.1 Descrição resumida

Através deste caso de uso o usuário poderá visualizar uma lista com todos os memberships cadastrados no sistema.

1.2 Atores

O ator principal neste caso de uso é o **Funcionário da Escola**.

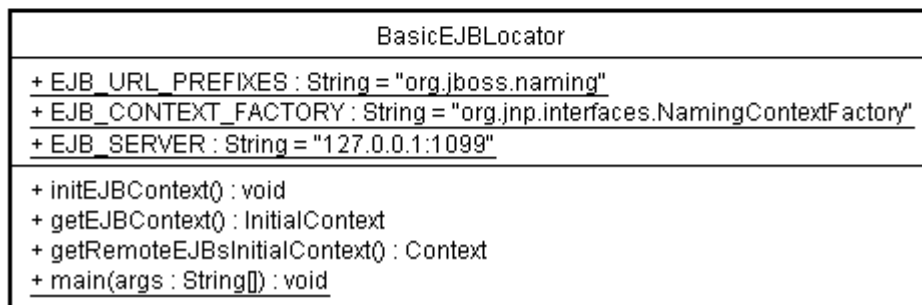
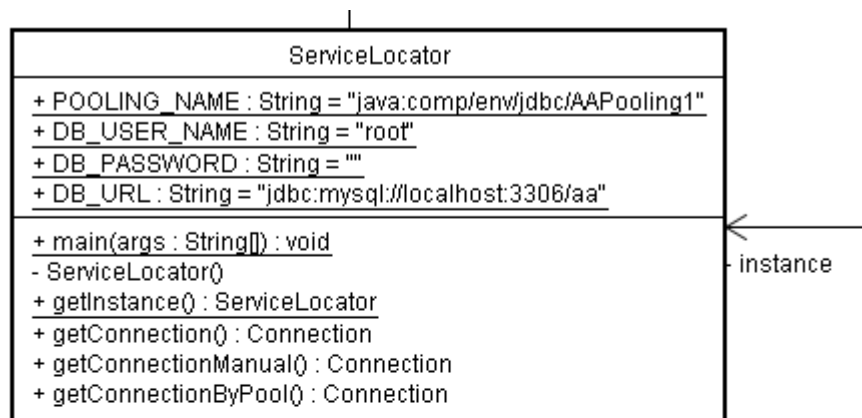
2 Fluxo de eventos

2.1 Fluxo primário

- 2.1.1 O usuário seleciona a opção “**Visualizar memberships**” no menu da aplicação.
- 2.1.2 É exibida uma lista com os memberships cadastrados contendo:
 - 2.1.2.1 Código do membership.
 - 2.1.2.2 Nome do membership (Hyperlink para edição do membership. **Fluxo alternativo 2.2.1**)
 - 2.1.2.3 Data de cadastro do membership.
 - 2.1.2.4 Endereço do membership.
- 2.1.3 Este caso de uso é encerrado

2.2 Fluxos alternativos (extensões)

JAREF – Exemplo da documentação



Digramas de classes

JAREF – Exemplo da documentação

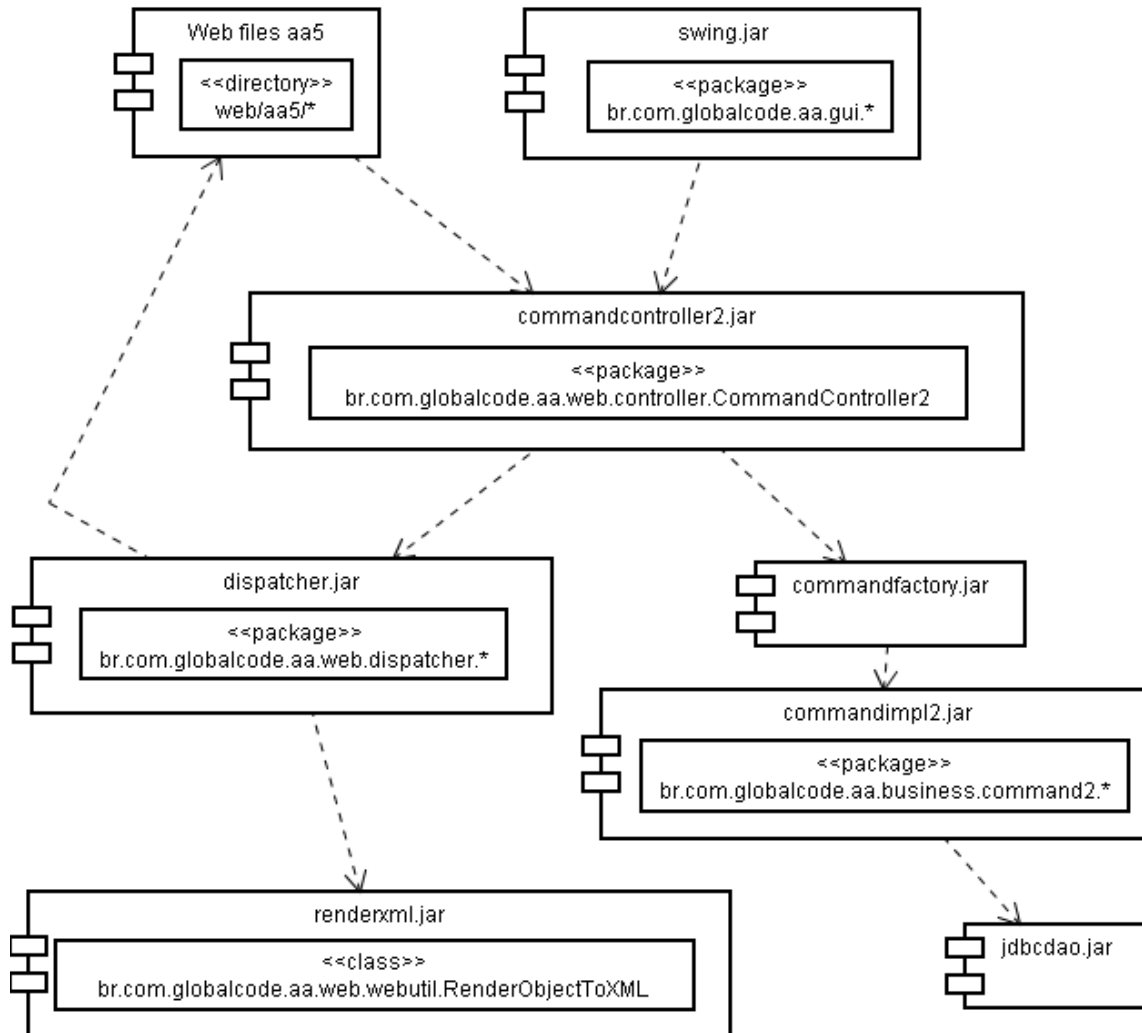
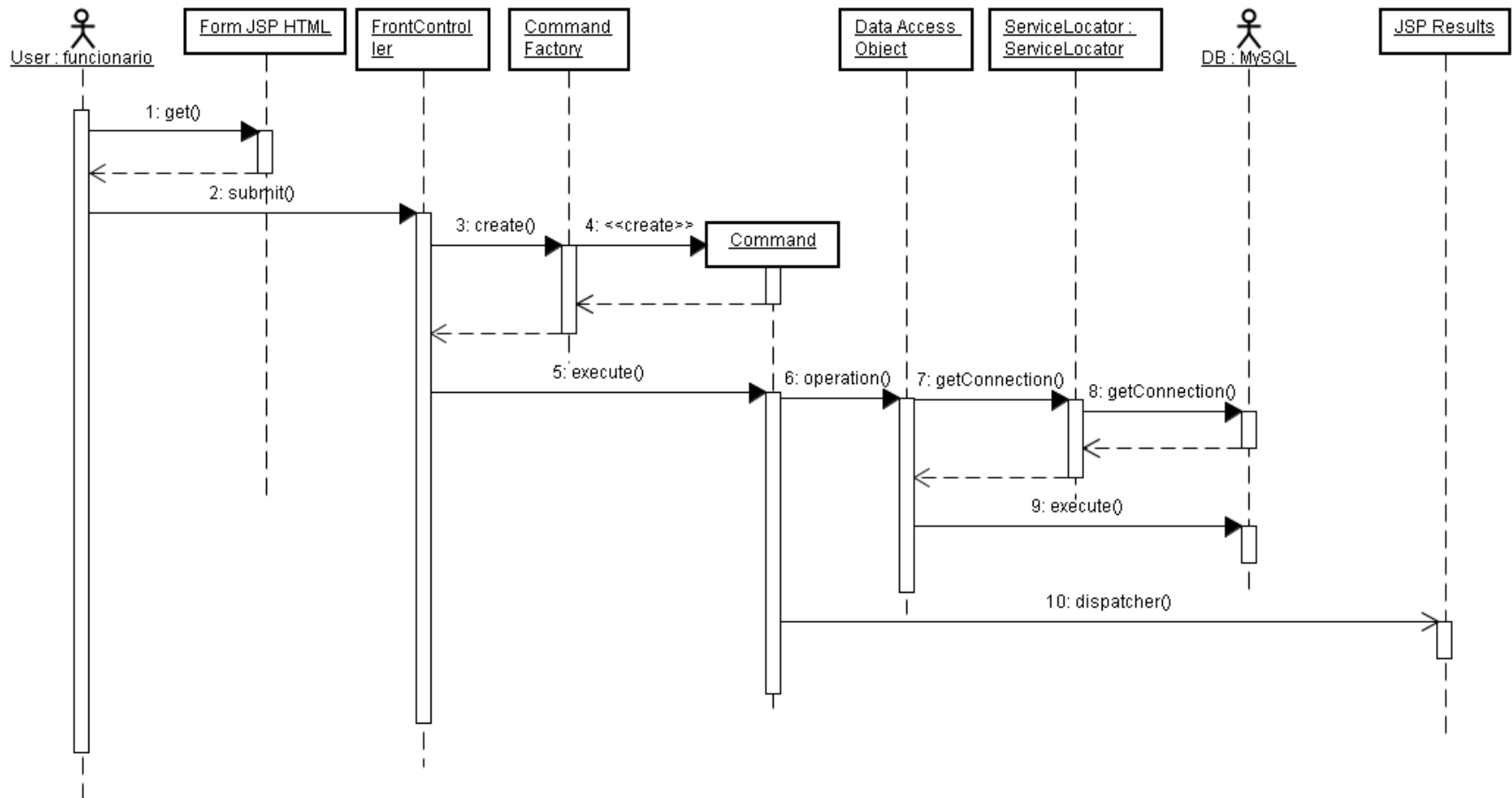


Diagrama de componentes: ilustra como é formado cada jar de cada arquitetura

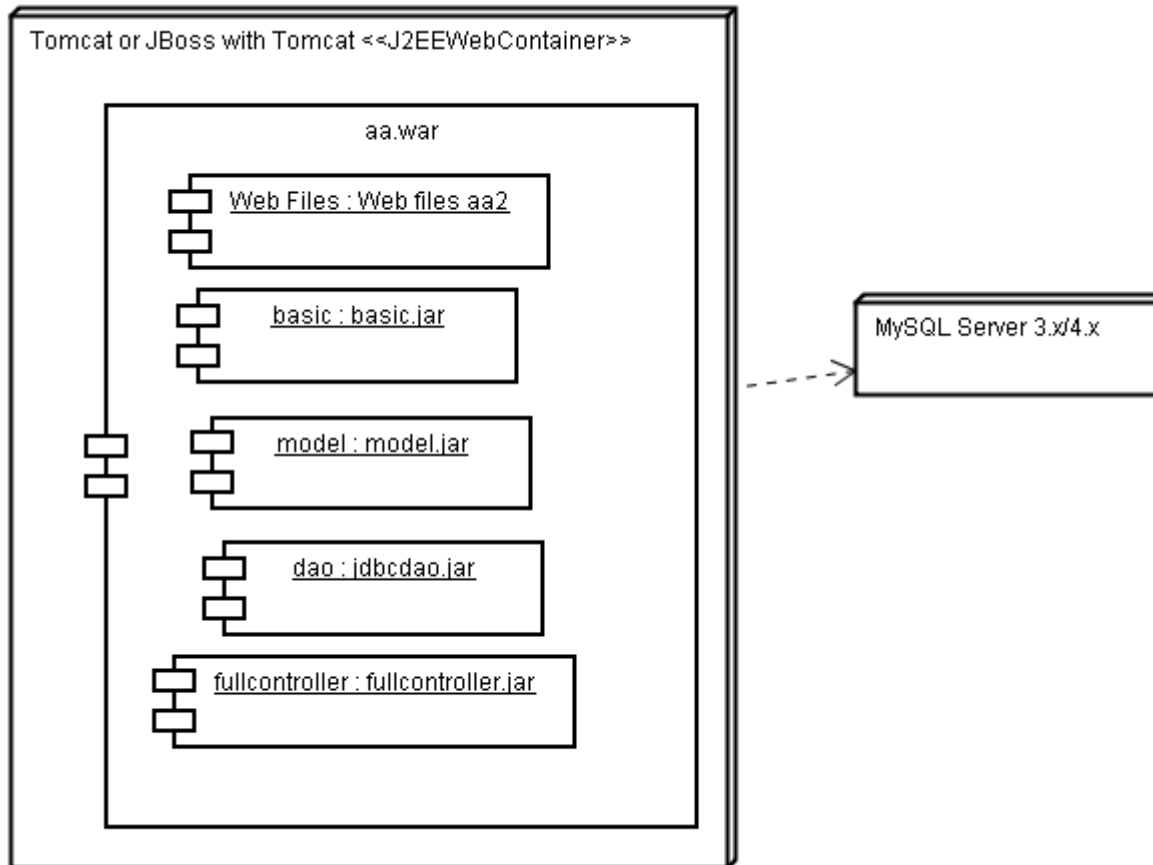
JAREF – Exemplo da documentação

Diagrama de seqüência



JAREF – Exemplo da documentação

Diagrama de deployment



Agenda

- Introdução a Arquiteturas
- JAREF: J2EE Architecture Research and Education Foundation
- **10 Alternativas de Arquiteturas J2EE**
- Futuro do projeto JAREF
- Conclusões

Architecture No. 1

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans;

Quando utilizar esta arquitetura

- Web-sites de pequeno e médio porte;
- Pequenos projetos de sistemas de administração para empresas com pouco conhecimento sobre Java e J2EE;
- Empresas com know-how em MS-ASP, PHP etc.

Quando evitar esta arquitetura

- Aplicativos de pequeno e médio porte que serão mantidos e modificados no médio e longo prazo;
- Empresas de médio porte em geral;

Risco Técnico: baixo

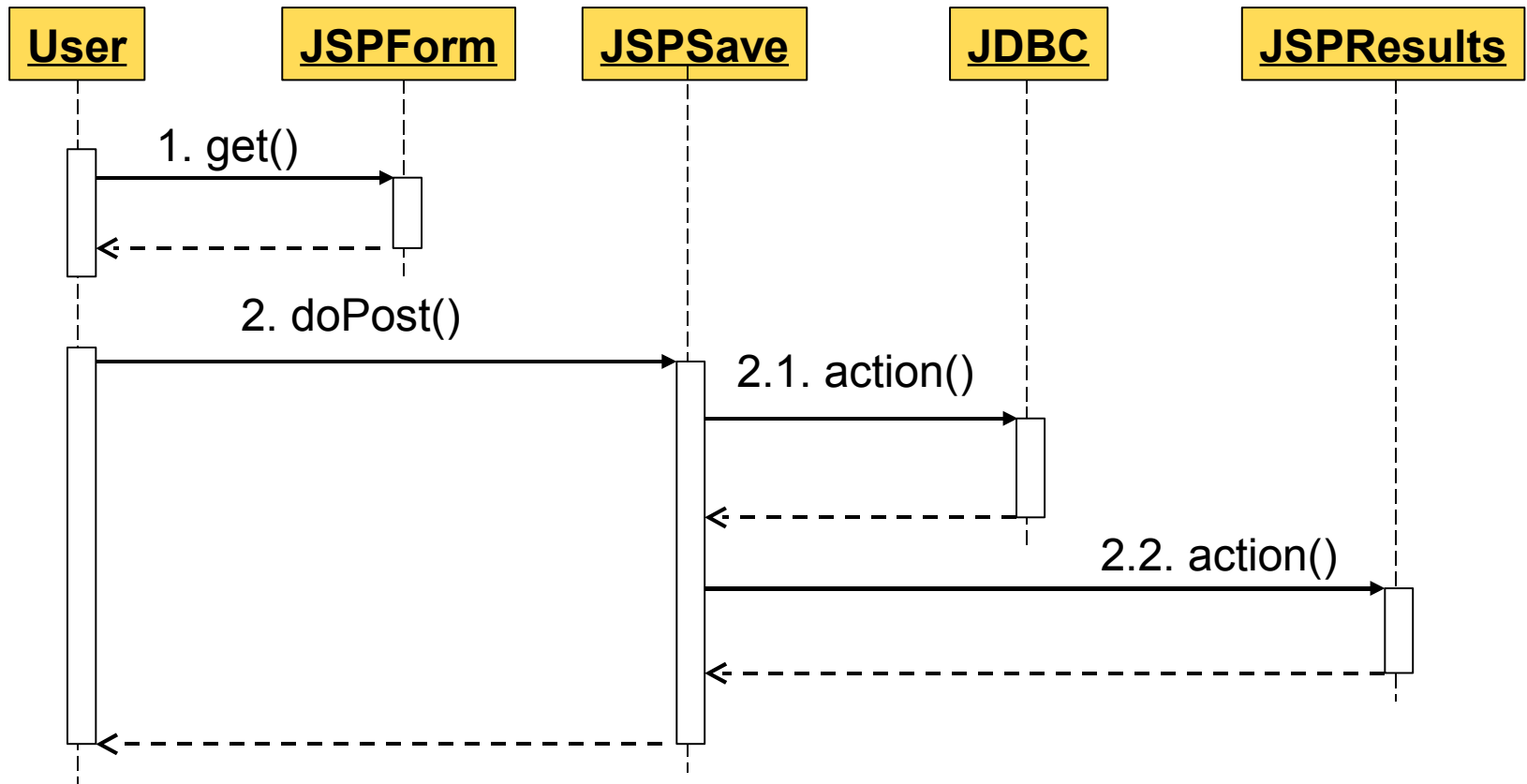
Fator custo: 1.0

Perfil de demanda: 1 até 1.000 usuários

Fator de qualidade: 1.0

Architecture No. 1

#1—JSP + Direct JDBC access



Architecture No. 2

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, **Data Access Object**;

Quando utilizar esta arquitetura

- Intranets com grande quantidade de acesso a banco de dados;
- Aplicativos de médio porte em empresas com pouco conhecimento J2EE;
- Aplicativos de médio porte com grande centralização de processamento no banco de dados e stored procedures;

Quando evitar esta arquitetura

- Aplicativos com grande quantidade de processamento em Java;
- Empresas de médio porte em geral;

Risco Técnico: baixo

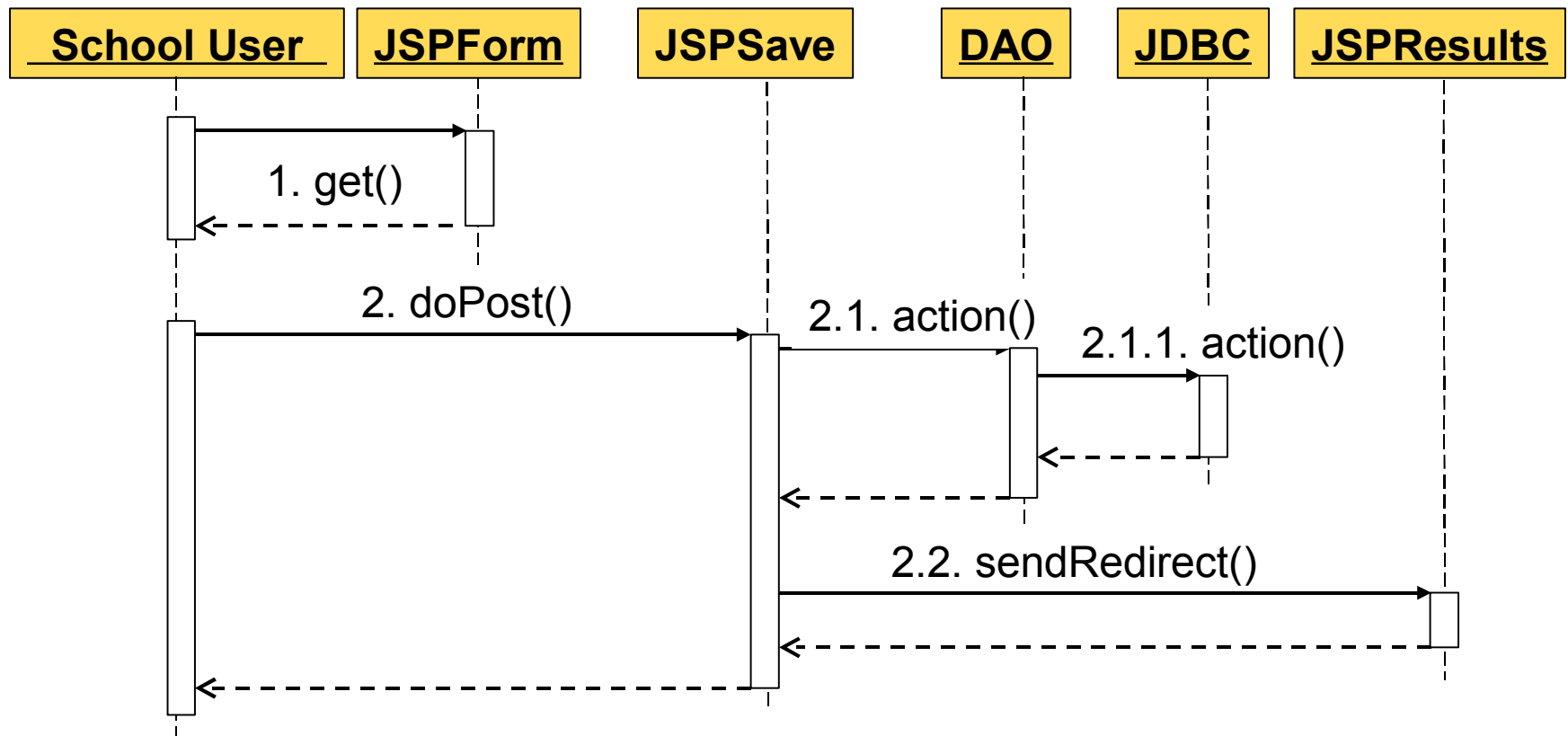
Fator custo: 1.05

Perfil de demanda: 1 até 1.000 usuários

Fator de qualidade: 1.1

Architecture No. 2

#2—JSP + DAOs



Architecture No. 3

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Data Access Object, **Model-view-controller básico, Front-Controller;**

Quando utilizar esta arquitetura

- Aplicativos de médio porte em empresas com conhecimento intermediário sobre Java e J2EE;

Quando evitar esta arquitetura

- Aplicativos com grande quantidade de processamento em Java;
- Aplicativos profissionais de médio porte para longo prazo;

Risco Técnico: médio

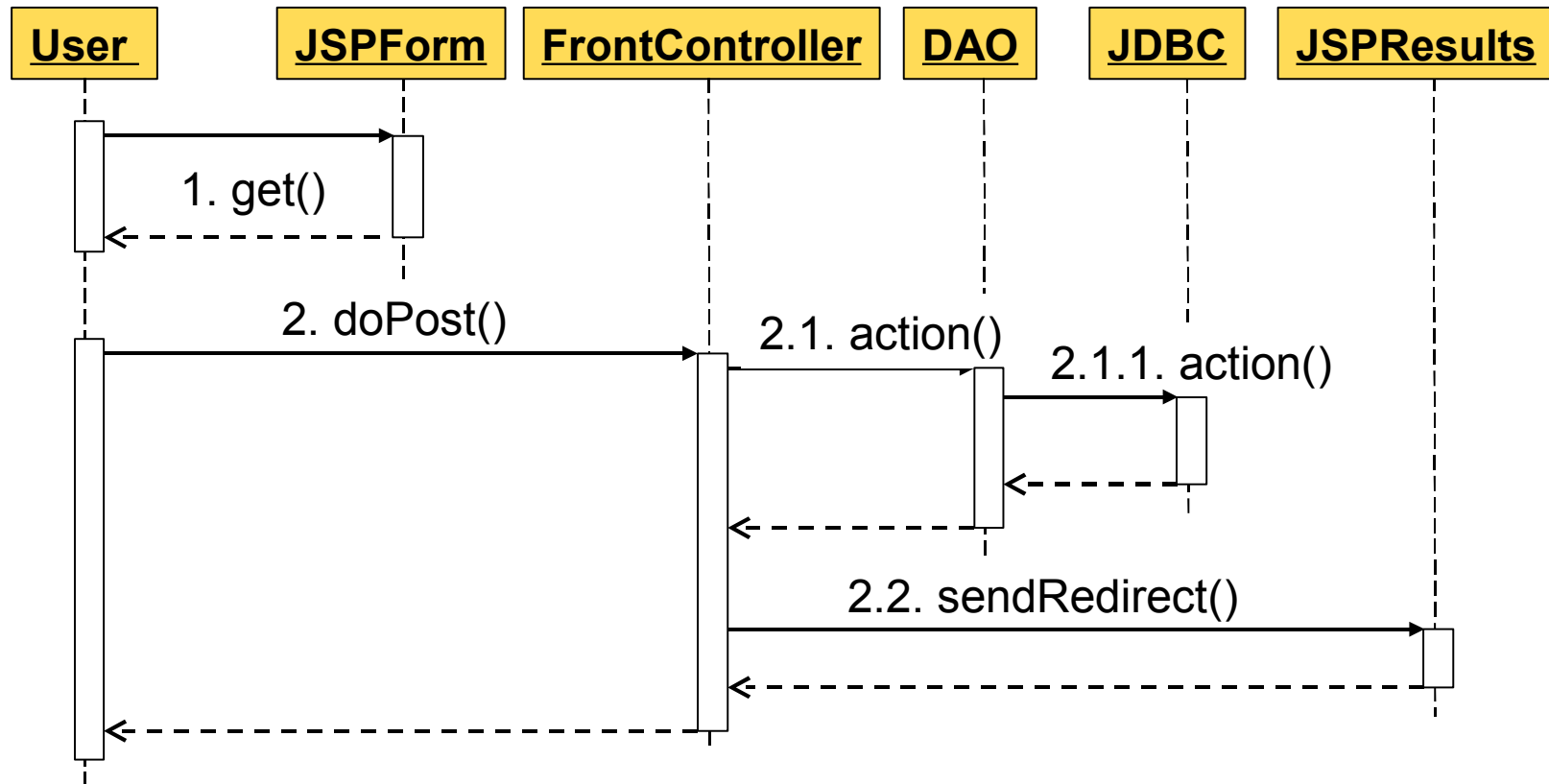
Fator custo: 1.15

Perfil de demanda: 1 até 1.000 usuários

Fator de qualidade: 1.3

Architecture No. 3

#3—#2 + Front Controller



Architecture No. 4

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Data Access Object, Model-view-controller básico, Front-Controller, **Command, Factory;**

Quando utilizar esta arquitetura

- Aplicativos de médio porte em empresas com conhecimento intermediário sobre J2EE e avançados sobre OO com Java;
- Cenários com tendências de crescimento e mudanças;

Quando evitar esta arquitetura

- Aplicativos profissionais de grande porte para longo prazo;
- Aplicativos que necessitam múltiplas visualizações para um mesmo processo;
- Cenários transacionais

Risco Técnico: médio

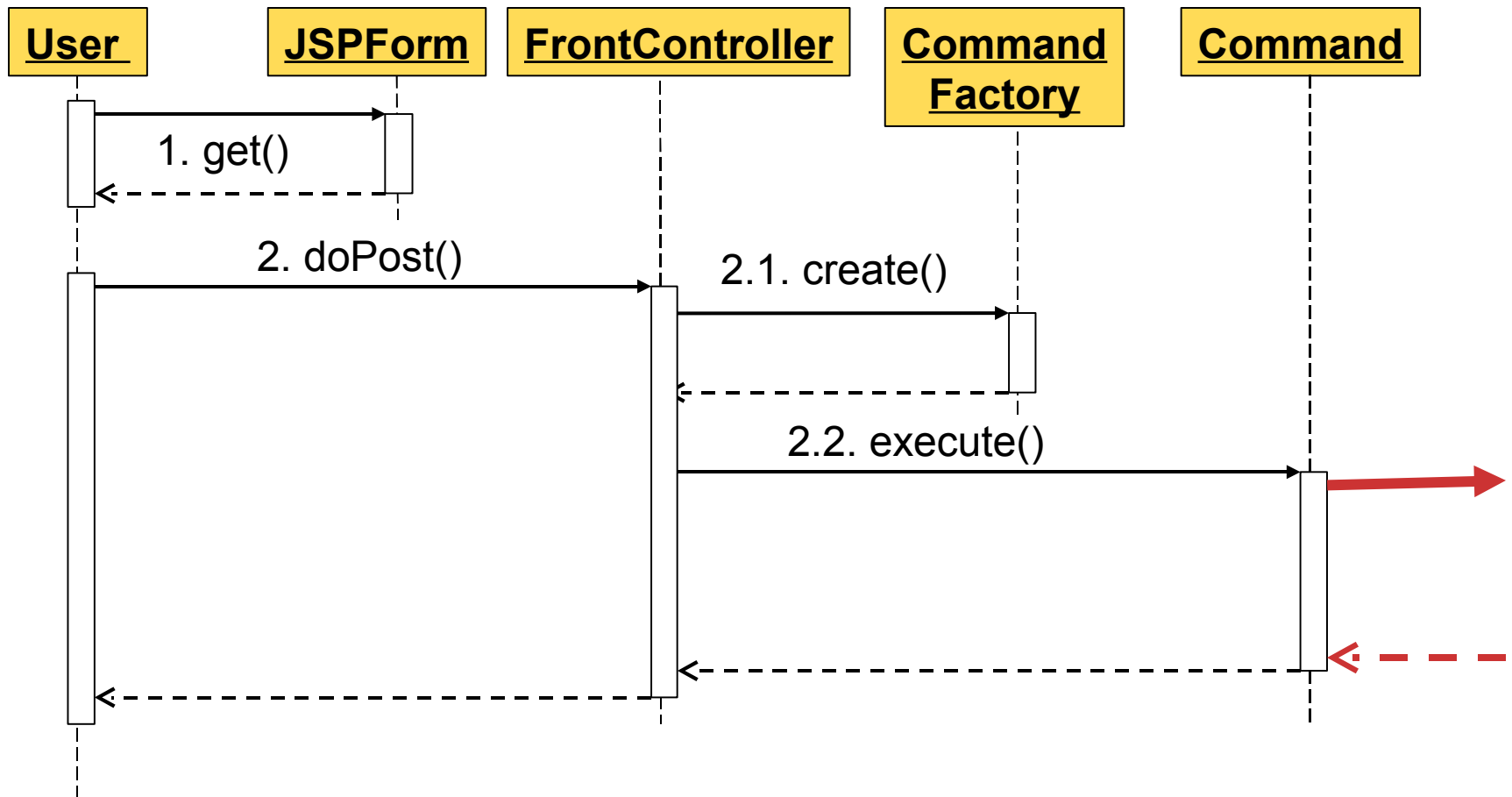
Fator custo: 1.25

Perfil de demanda: 1 até 1.000 usuários

Fator de qualidade: 1.5

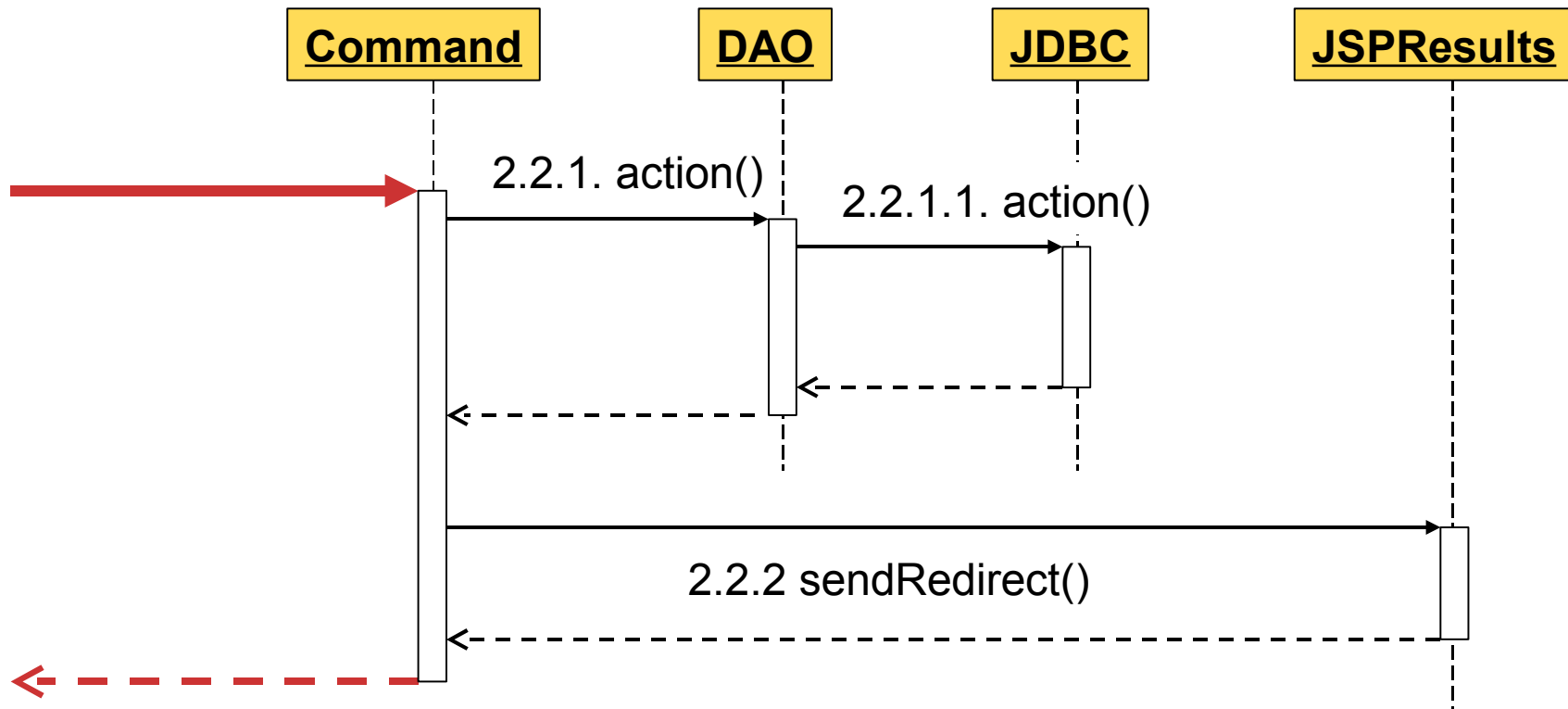
Architecture No. 4

#4—#3 + Command + Command Factory (part 1)



Architecture No. 4

#4—#3 + Command + Command Factory (part 2)



Architecture No. 5

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Data Access Object, Model-view-controller básico, Front-Controller, Command, Factory, **View Dispatcher, Swing-client com HTTP tunneling, Flash Client;**

Quando utilizar esta arquitetura

- Aplicativos Web com necessidade de múltiplos clients em empresas com domínio total da plataforma Web J2EE e Java;

Quando evitar esta arquitetura

- Cenários transacionais
- Necessidade de Integração;

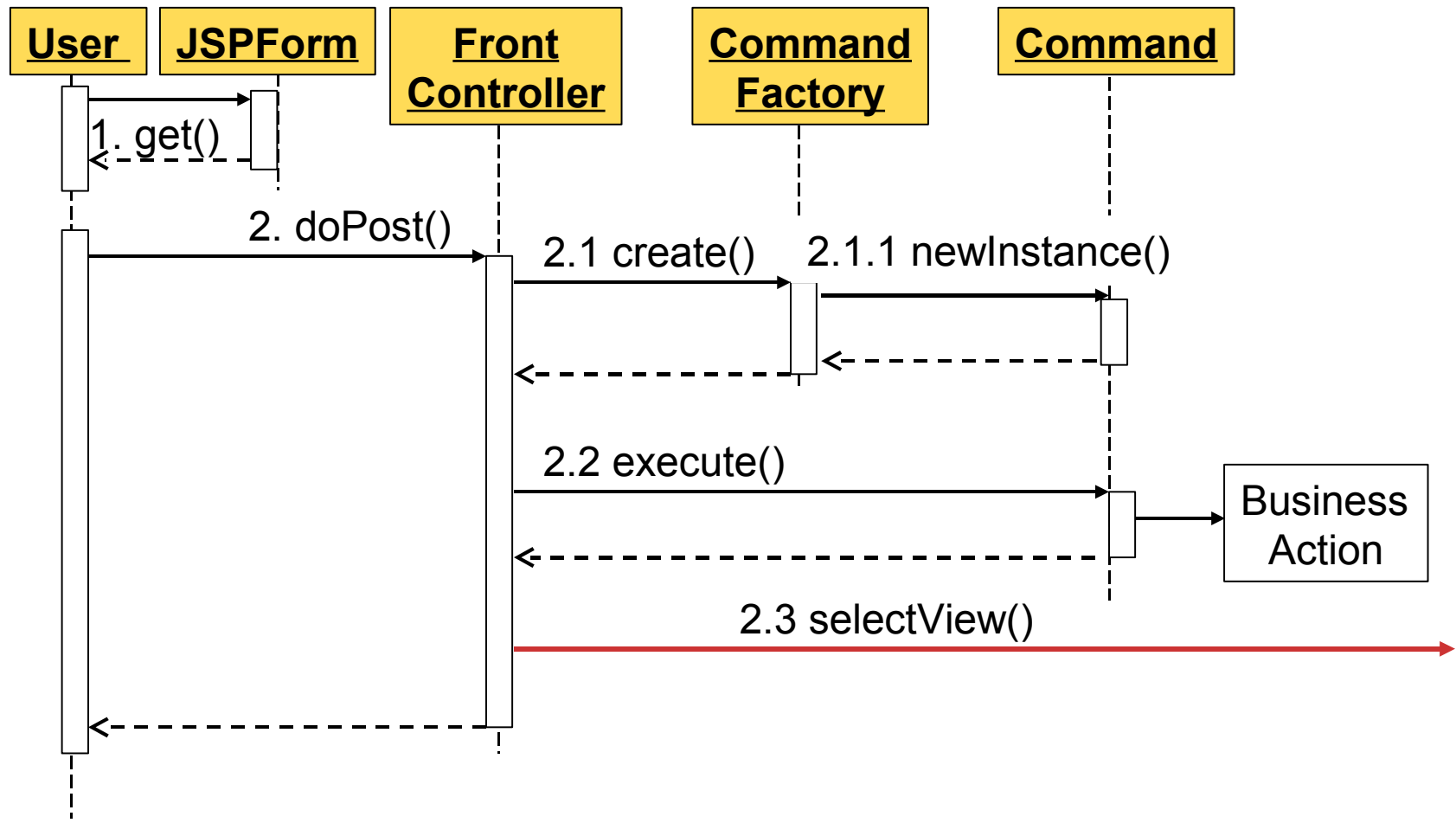
Risco Técnico: médio/alto

Fator custo: 1.70

Perfil de demanda: 1 até 1.000 usuários

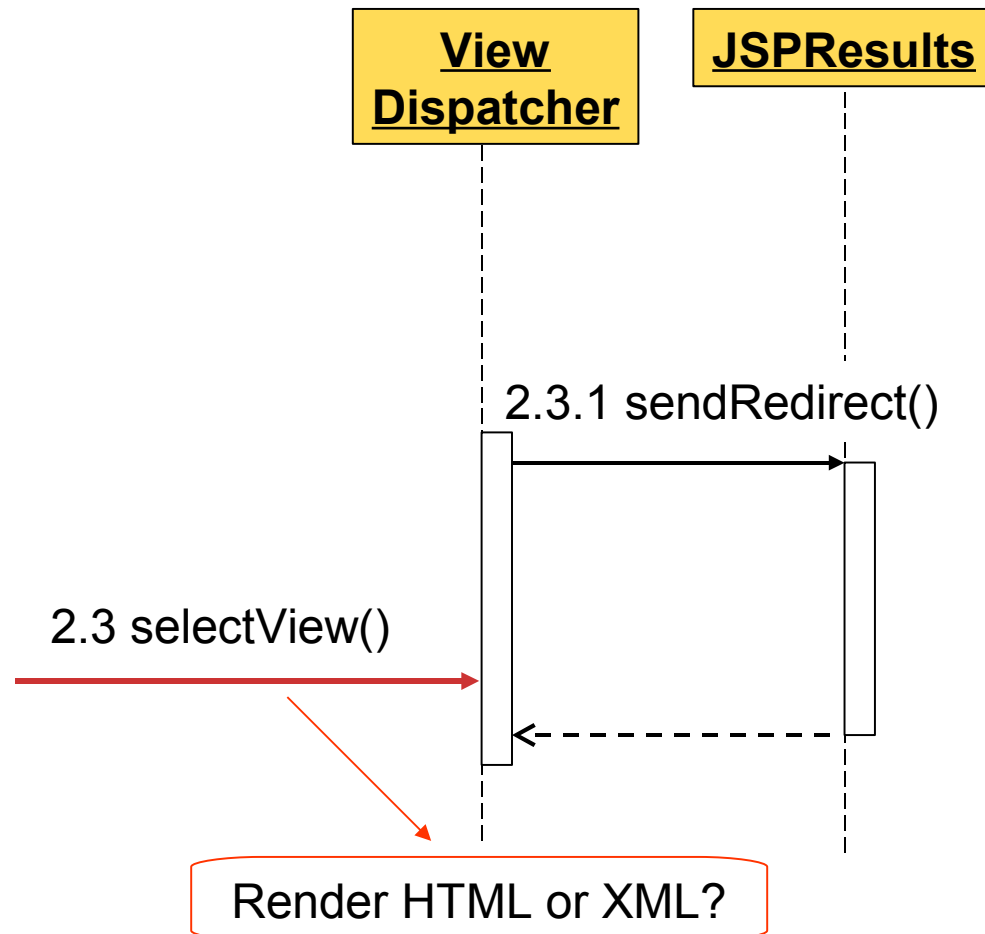
Fator de qualidade: 2.3

#5—#4 + View Dispatcher (part 1)



Architecture No. 5

#5—#4 + View Helper + View Dispatcher (part 2)



Architecture No. 6

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Data Access Object, Model-view-controller básico, Front-Controller, Command, Factory, View Dispatcher, Swing-client com HTTP tunneling, Flash Client, **Intercepting Filter, View Helper;**

Quando utilizar esta arquitetura

- Aplicativos Web de grande porte;

Quando evitar esta arquitetura

- Cenários transacionais
- Necessidade de Integração;

Risco Técnico: alto

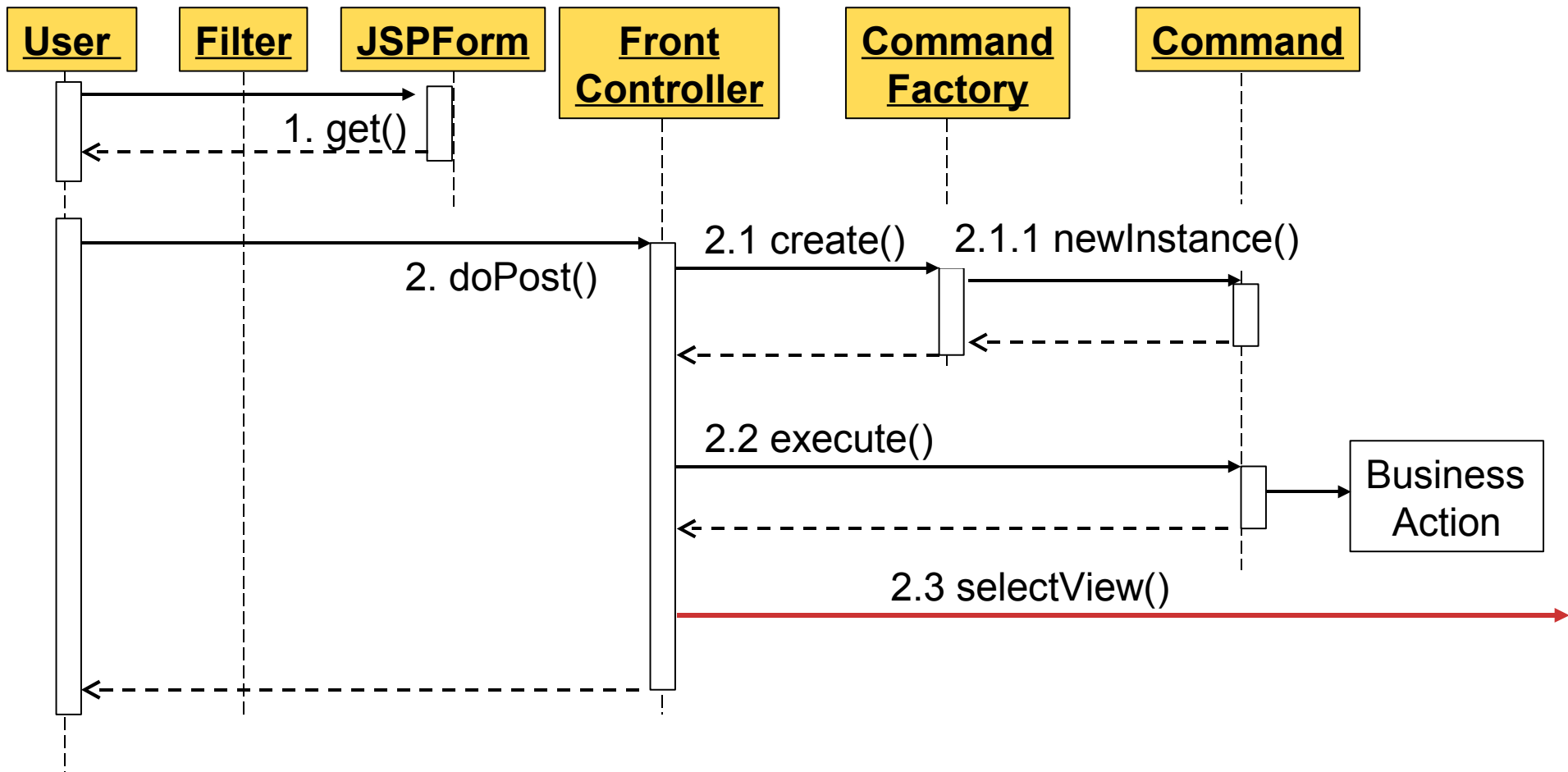
Fator custo: 1.90

Perfil de demanda: 1 até 1.000 usuários

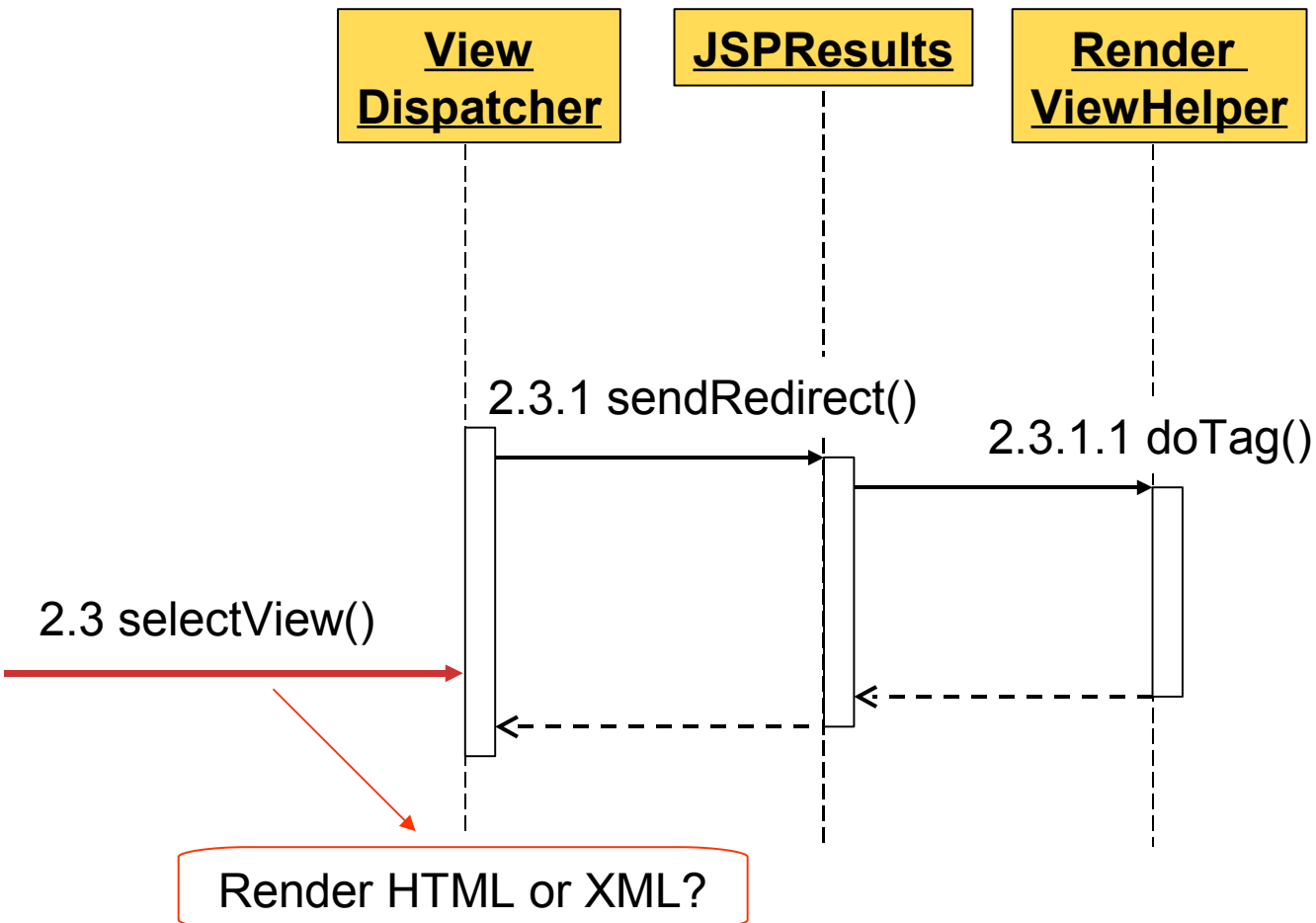
Fator de qualidade: 2.5

Architecture No. 6

#6— + Intercepting Filter + View Helper + (part 1)



Architecture No. 6



Architecture No. 7

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Data Access Object, Model-view-controller básico, Front-Controller, Command, Factory, View Dispatcher, Swing-client com HTTP tunneling, Flash Client, Intercepting Filter, View Helper, **EJB Session bean, Business Delegate;**

Quando utilizar esta arquitetura

- Aplicativos Web e não web de alta escalabilidade e flexibilidade de arquitetura;

Quando evitar esta arquitetura

- Aplicativos de médio porte;
- Empresas sem conhecimento de EJB's;

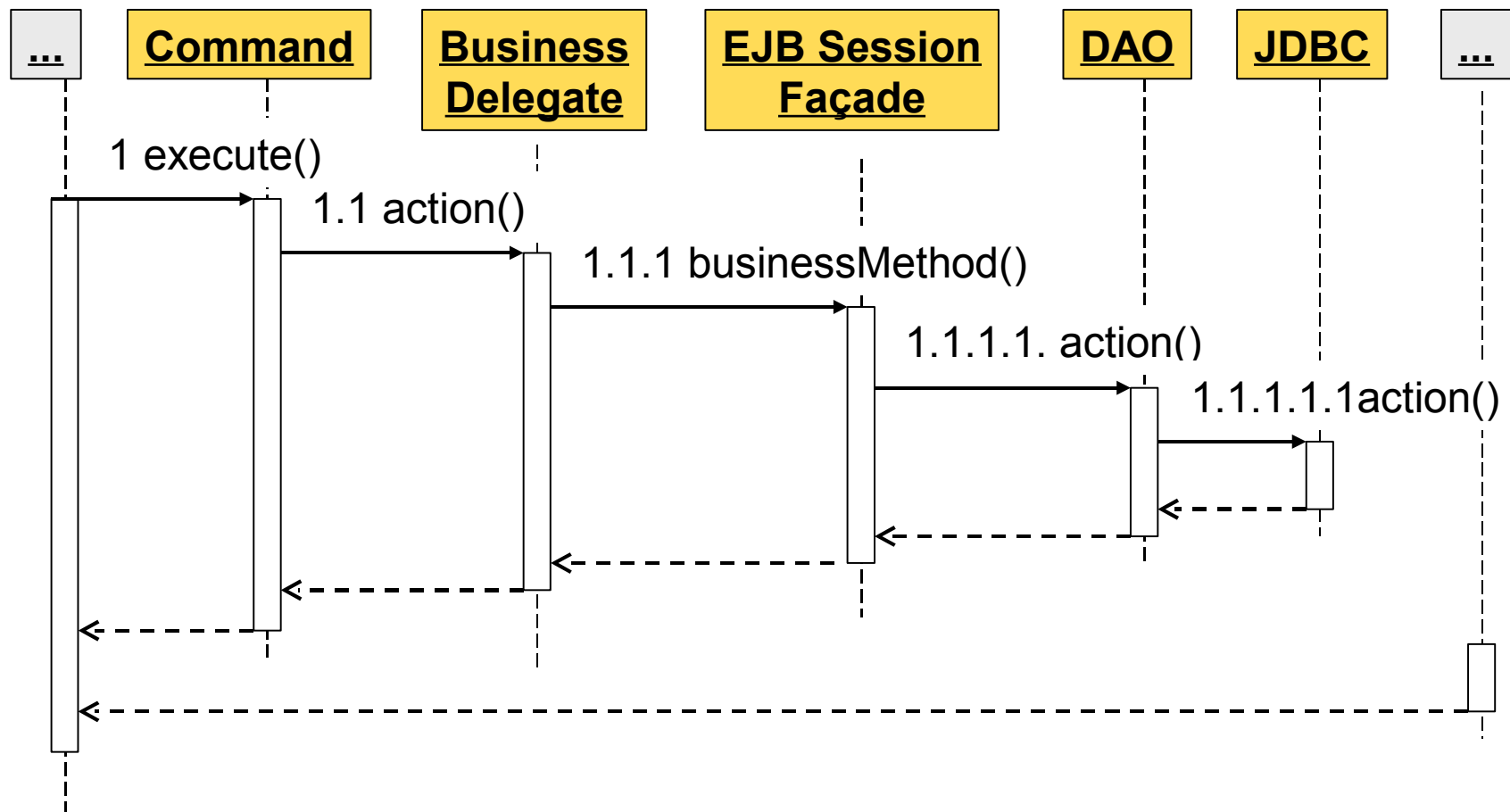
Risco Técnico: alto

Fator custo: 2.50

Perfil de demanda: 1 a 10.000 usuários

Fator de qualidade: 3.0

Architecture No. 7



Architecture No. 8

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Data Access Object, Model-view-controller básico, Front-Controller, Command, Factory, View Dispatcher, Swing-client com HTTP tunneling, Flash Client, Intercepting Filter, View Helper, EJB Session bean, Business Delegate, **Entity BMP, Session Façade;**

Quando utilizar esta arquitetura

- Aplicativos Web e não web de alta escalabilidade e flexibilidade de arquitetura;
- Cenários onde o principal consumidor do database é o container Java;
- Cenários justifica-se fazer cachê do banco de dados;

Quando evitar esta arquitetura

- Aplicativos de médio porte;
- Empresas sem conhecimento **avançados** de EJB's;

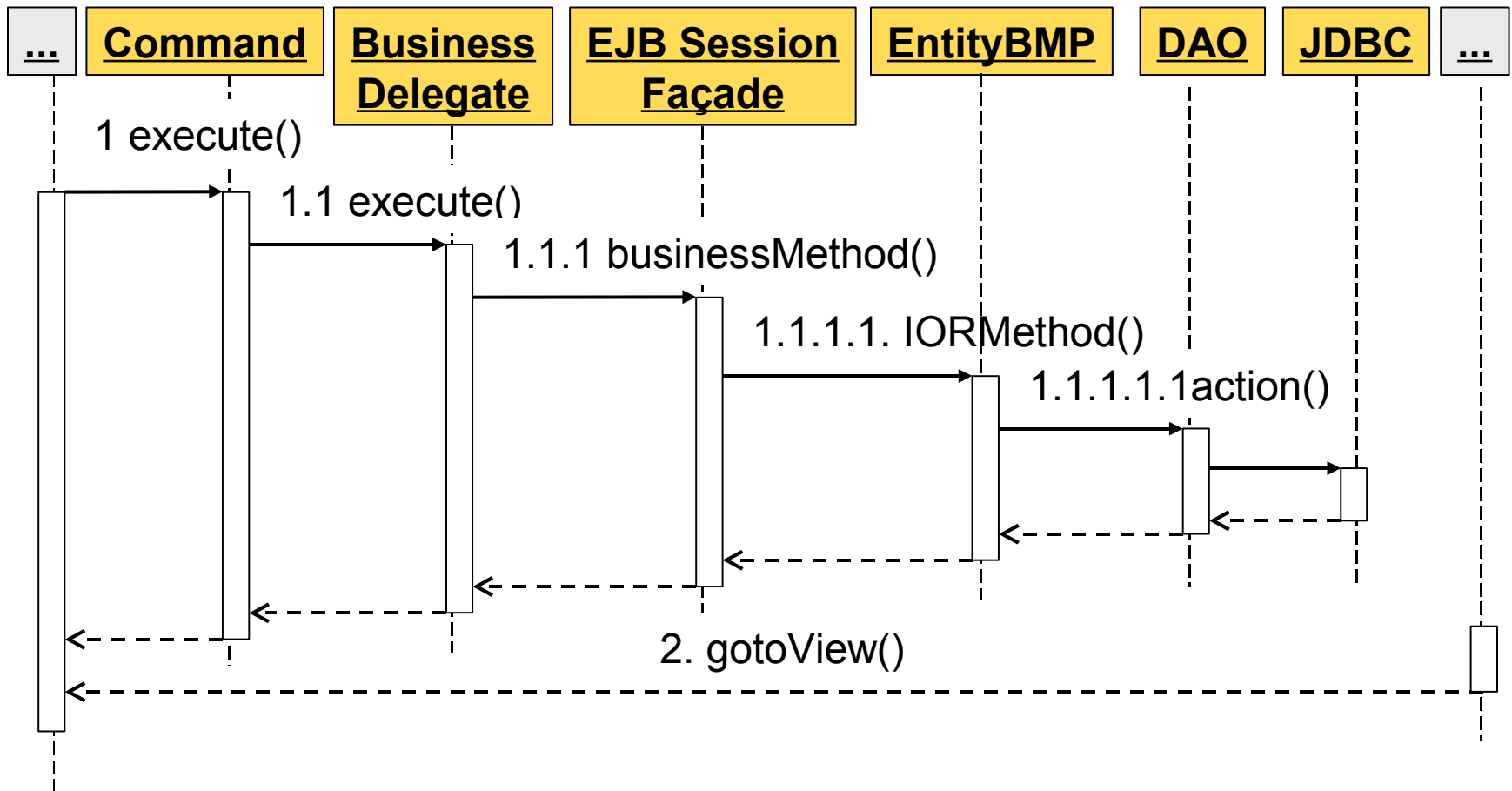
Risco Técnico: altíssimo

Fator custo: 3.20

Perfil de demanda: 1 a 10.000 usuários

Fator de qualidade: 3.3

Architecture No. 8



Architecture No. 9

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Model-view-controller básico, Front-Controller, Command, Factory, View Dispatcher, Swing-client com HTTP tunneling, Flash Client, Intercepting Filter, View Helper, EJB Session bean, Business Delegate, **Entity CMP, Session Façade;**

Quando utilizar esta arquitetura

- Aplicativos Web e não web de alta escalabilidade e flexibilidade de arquitetura;
- Cenários onde o principal consumidor do database é o container Java;
- Cenários justifica-se fazer cachê do banco de dados;

Quando evitar esta arquitetura

- Aplicativos de médio porte;
- Empresas sem conhecimento **avançados** de EJB's;

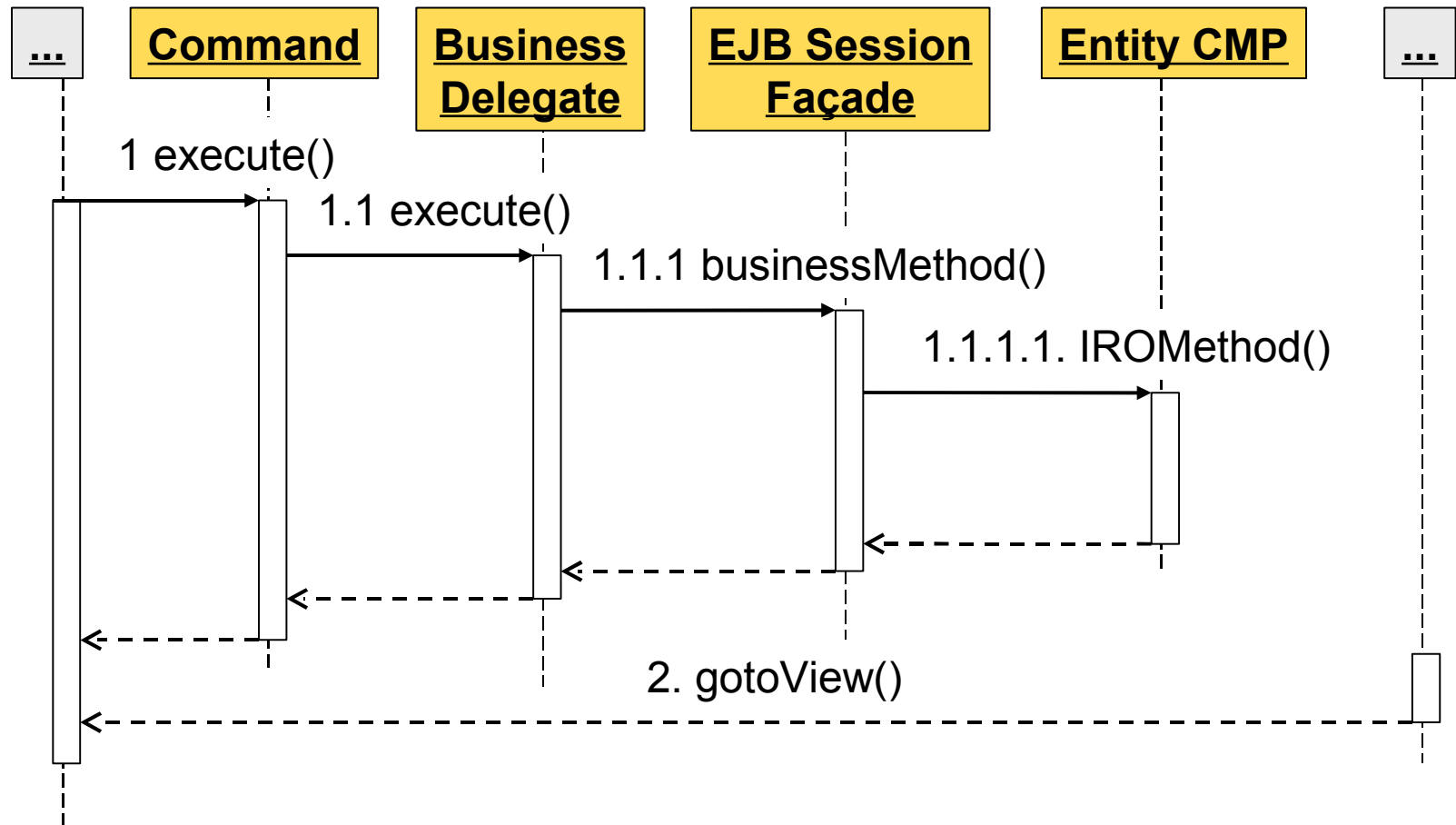
Risco Técnico: altíssimo

Fator custo: 3.10

Perfil de demanda: 1 a 10.000 usuários

Fator de qualidade: 3.5

Architecture No. 9



Architecture No. 10

Recursos utilizados: Connection pool, Service Locator, JSP's codificados com JavaBeans, Model-view-controller básico, Front-Controller, Command, Factory, View Dispatcher, Swing-client com HTTP tunneling, Flash Client, Intercepting Filter, View Helper, EJB Session bean, Business Delegate, Entity CMP, Session Façade, Message-driven Beans;

Quando utilizar esta arquitetura

- Necessidade de altíssima confiabilidade;
- Demandas “elásticas”;

Quando evitar esta arquitetura

- Utilizar somente em projetos “enormes”: NASA, Sistema de Pagamento Brasileiro entre outros utilizam esta arquitetura;

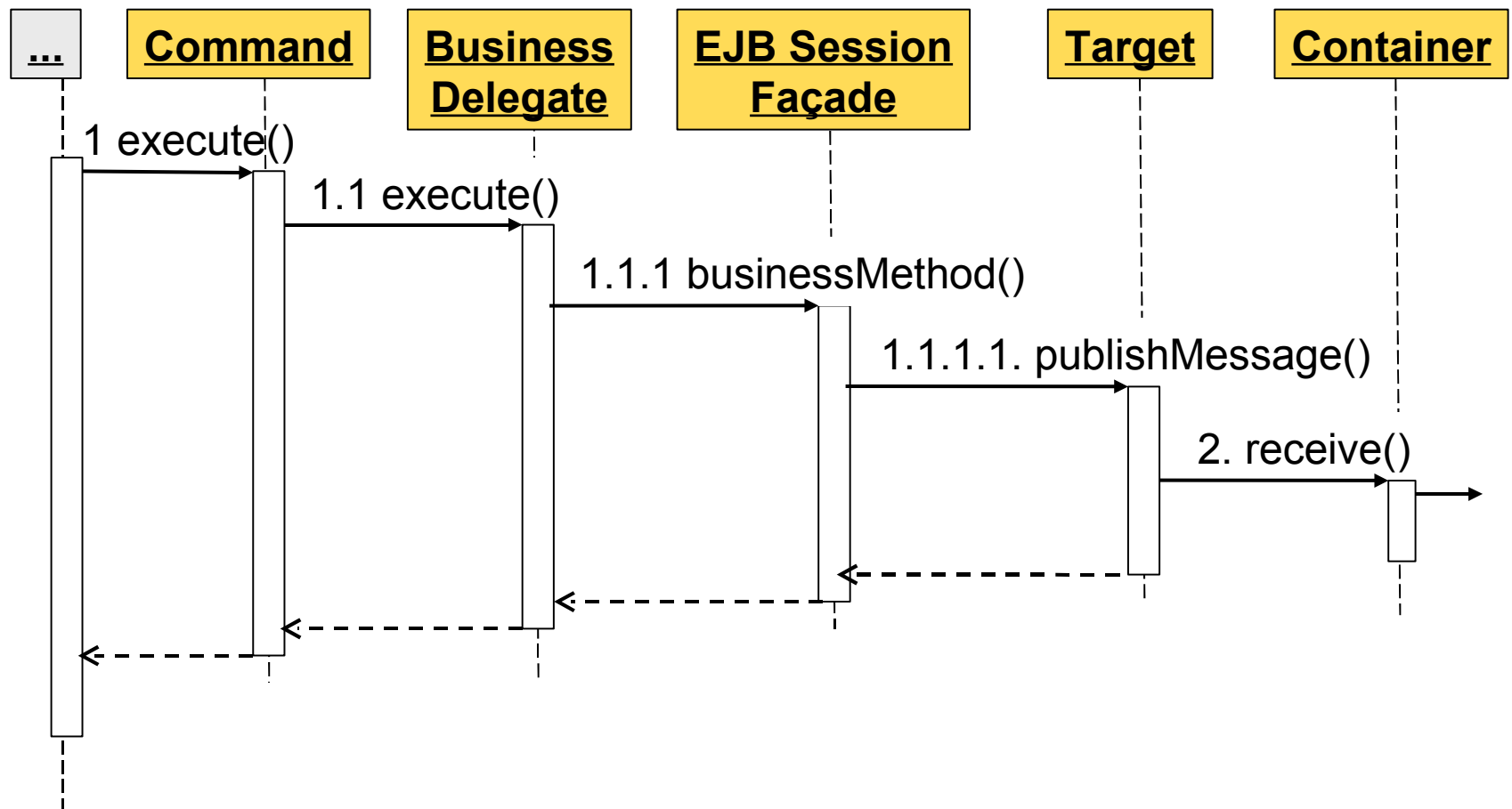
Risco Técnico: altíssimo

Fator custo: 3.80

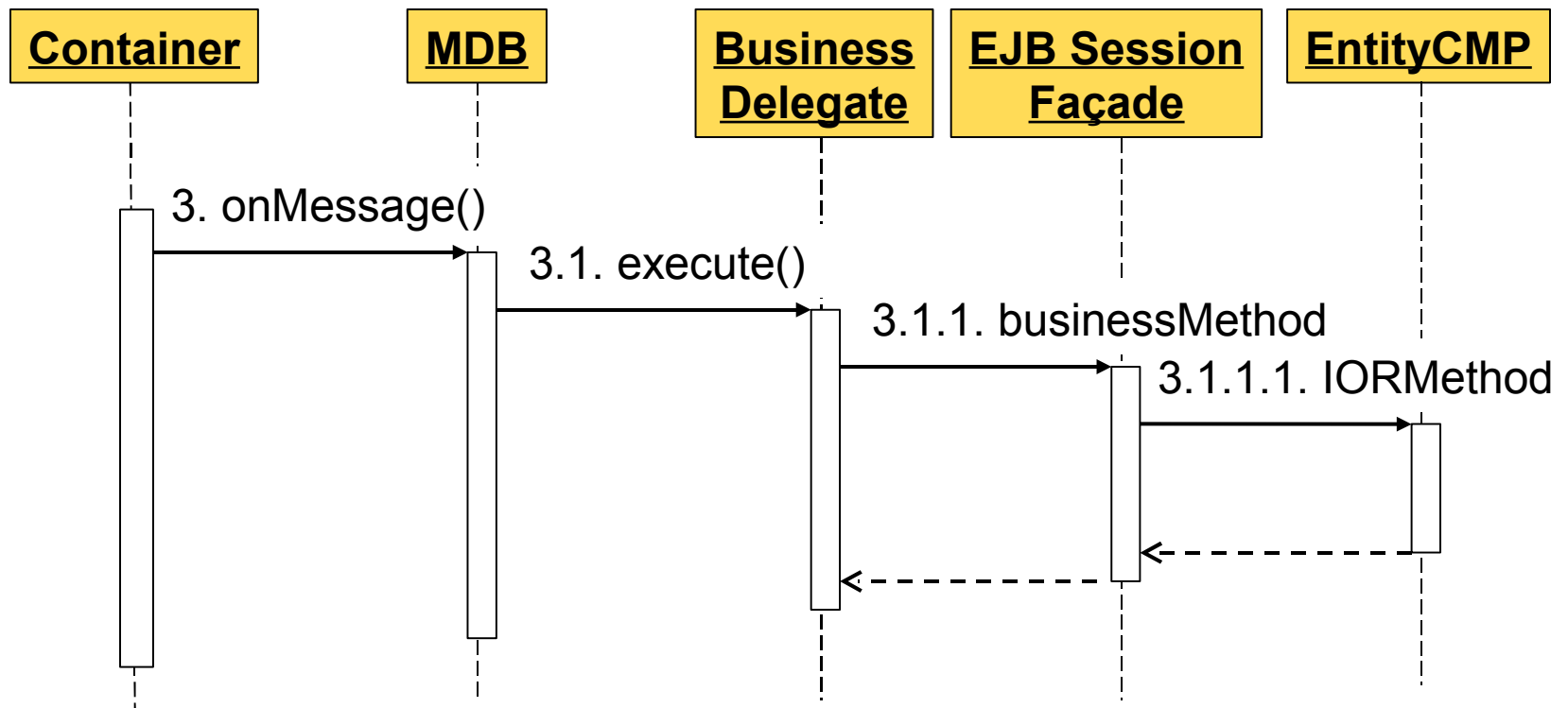
Perfil de demanda: acima de 10.000 usuários

Fator de qualidade: 5.0

Architecture No. 10



Architecture No. 10



Agenda

- Introdução a Arquiteturas
- JAREF: J2EE Architecture Research and Education Foundation
- 10 Alternativas de Arquiteturas J2EE
- **Futuro do projeto JAREF**
- Conclusões

JAREF

Futuro do projeto

- Incluir arquiteturas com os frameworks Struts, Hibernate, OJB e Java Server Faces;
- Incluir estudos sobre custo de transações, WebServices e logging;
- Estudos avançados sobre escalabilidade e EJB's;
- Fazer um benchmark das 10 arquiteturas com 4 application servers;
- Architecture Wizard;
- Lista de 12 pequenos refactorings para aumentar código de processamento de dados nos EJB's;

Agenda

- Introdução a Arquiteturas
- JAREF: J2EE Architecture Research and Education Foundation
- 10 Alternativas de Arquiteturas J2EE
- Futuro do projeto JAREF
- **Conclusões**

Conclusão: diferentes arquiteturas, diferentes necessidades

