# JavaEE

# architectures

I'm working as Java EE/SE developer, consultant, sometimes author, speaker and trainer with Java since 1995...

...and still really enjoying it!

Java Programming Language rocks!

**airhacks.io**

adam-bien.com

press.adam-bien.com

# Real World Java EE Night Hacks

## Dissecting the Business Tier

[ Iteration One ]

CDI REST JPA2 JMX

EJB 3.1 JAX-RS Maven 3 Java EE

ScalaTest    Hudson        DevOps        BASE

Arquillian    JUnit          CI            Stress Test

X - RAY

Adam Bien

Foreword by James Gosling

Adam Bien,
press.adam-bien.com

adam-bien.com

realworldpatterns.com

adam-bien.com

# Understanding CAP in Java EE context

# CAP vs. FLP (Fischer, Lynch and Paterson)

In an asynchronous network where messages may be delayed but not lost, there is no consensus algorithm that is guaranteed to terminate in every execution for all starting conditions, if at least one node may fail-stop.

# Who needs 24*7?

store.apple.com/de/go/macbookpro/

| Store | Mac | iPod | iPhone | iPad | iTunes | Support |

# We'll be back.

We're busy updating the Apple Store
for you and will be back soon.

Store    Mac    iPod    iPhone    iPad    iTunes    Support

# Wir sind bald wieder da.

Wir aktualisieren gerade das Angebot im Apple Online Store. Die Seite ist demnächst wieder verfügbar.

...and

Store | Mac | iPod | iPhone | iPad | iTunes | Support

## Apple Store

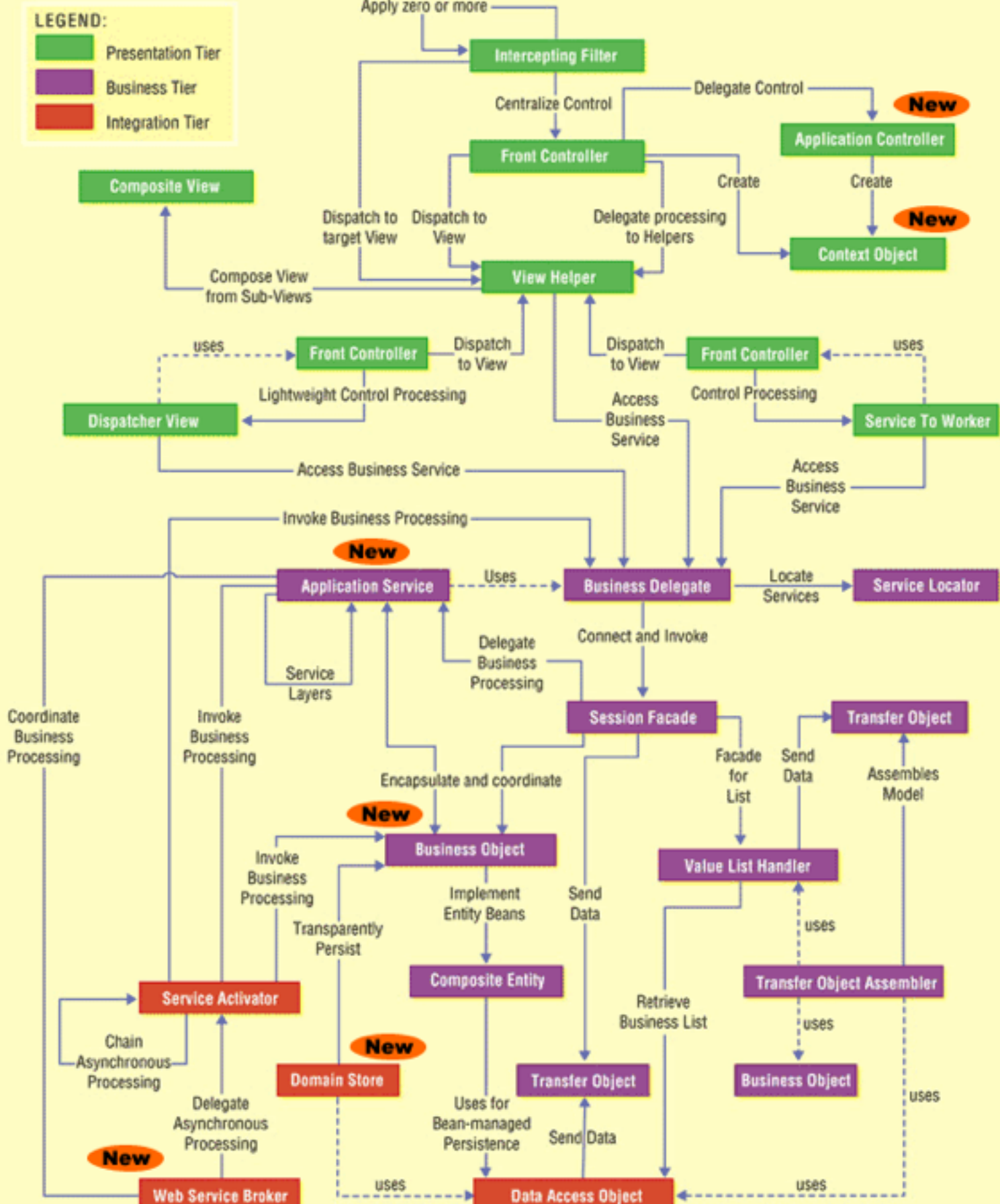📞 1-800-MY-APPLE | ⚙ Account | 🛒 Cart

# We're sorry.

We're sorry, but our Order Status services are extremely busy right now. Please try again later.

Apple uses industry-standard encryption to protect the confidentiality of your personal information.

# BASE vs. ACID, or how to start a project

# Deprecated J2EE Patterns

# Core J2EE Patterns, 2nd Edition

Apply zero or more → **Intercepting Filter**

Centralize Control

Delegate Control → **New** → **Application Controller**

**Front Controller**

Create → Create

**New** → **Context Object**

**Composite View**

Dispatch to target View | Dispatch to View | Delegate processing to Helpers

Compose View from Sub-Views → **View Helper**

uses → **Front Controller** → Dispatch to View

Dispatch to View → **Front Controller** ← uses

Lightweight Control Processing

Access Business Service

Control Processing

**Dispatcher View**

**Service To Worker**

Access Business Service

Access Business Service

Invoke Business Processing

**New** → **Application Service** --Uses--> **Business Delegate** → Locate Services → **Service Locator**

Service Layers

Delegate Business Processing

Connect and Invoke

Coordinate Business Processing

Invoke Business Processing

**Session Facade**

Facade for List

Send Data

**Transfer Object**

Encapsulate and coordinate

Assembles Model

Invoke Business Processing

**New** → **Business Object**

**Value List Handler**

Transparently Persist

Implement Entity Beans

Send Data

uses

**Service Activator**

**Composite Entity**

Send Data

Retrieve Business List

**Transfer Object Assembler**

Chain Asynchronous Processing

**New** → **Domain Store**

uses

**Business Object**

Delegate Asynchronous Processing

Uses for Bean-managed Persistence

**Transfer Object**

Send Data

**New** → **Web Service Broker**

uses → **Data Access Object** ← uses

uses

# GoF Patterns

Factory, Abstract Factory, Singleton, Builder, Prototype, Adapter, Bridge, Decorator, Facade, Flyweight, Composite, Proxy, Template, Observer, Visitor, Command, Strategy, State, Chain of Responsibility

# Documentation and Test

# Conway's Law

# Parkinson's Law of Triviality

# Deployment Strategies

- Blue-Green Deployment

- Hot Deployment

- Virtualization

- Containers

# GRASP

- High Cohesion, Minimal Coupling

- Protected Variations

- Information Expert

- Polymorphism

- Pure Fabrication

# Reactive Programming

# Pragmatic modularization of Java EE applications

# Command Query Responsibility Segregation (CQRS)

# Structuring

- The Vision

- Cohesion <-> Coupling

- Essential Entities and Business Components

- "Maximal Cohesion, Minimal Coupling"

# Designing plugins, templates and hooks

# Caveats of asynchronous applications

# RESTful architectures vs. SOAP

# Service governance and versioning

- API Versioning

- Dealing with extensions

- (In)Compatiblities

- SCM

# Stateful vs. stateless Java EE architectures

- Where is the state?

- What about the scalability?

# MoM and Java EE with JMS

- API Compatibility

- Deliver Once And Only Once

- Persistent Queues / Durable Subscribers

- Poisoned Messages

# Logging, errorhandling

- Who needs logs?

- Error classification

- Audits and traces

# Management and monitoring

- JMX, REST

- "Business monitoring"

- Automated setup and deployment

# Continuous Integration and Continuous Deployment

# Introduction to clustering, shared nothing and grid architectures

# Integration of legacy resources with and without JCA 1.6: code walk-throughs

# Thanks && see you:

Java EE User Interfaces,
Java EE 7 Delta

March 2014
  or / and
Java Persistence, April 2014

# Thank You!

blog.adam-bien.com
twitter.com/AdamBien