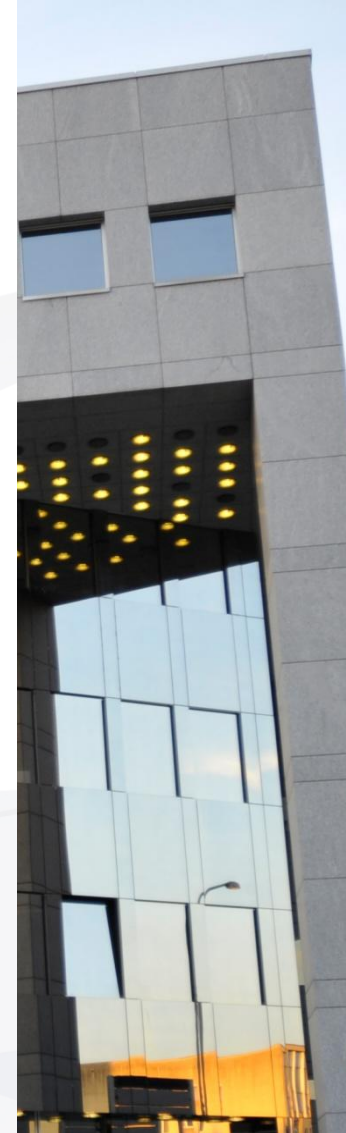


Migrating from commercial, proprietary J2EE Application Servers to JBoss

6 April 2009, Technopark Zürich

JUG'S

Java User Group Switzerland



Summary of this talk

- **Part I: Project management perspective**
 - The nature of migration projects
 - Risks and how to avoid them
 - First hand experience
- **Part II: Technological Insight**
 - Processes
 - Preparation
 - Implementation issues
 - Validation

Speakers



- Markus Grieder
 - Informatik Ingenieur FH
 - Senior Software Engineer



- Christoph Kuhn
 - Dr. oec. HSG & dipl. Inf-Ing. ETH
 - Senior VP

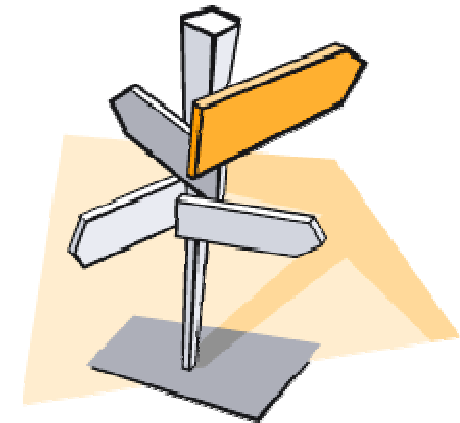
About Crealogix

- The CREALOGIX GROUP is one of the leading providers of e-business and ERP (Enterprise Resource Planning) solutions in Switzerland, Germany and Austria. CREALOGIX Holding AG shares (CLXN) are listed on the SWX Swiss Exchange.
- Founded in 1996
- JUGS'S Silver Sponsor
- JBoss Preferred Solution Partner

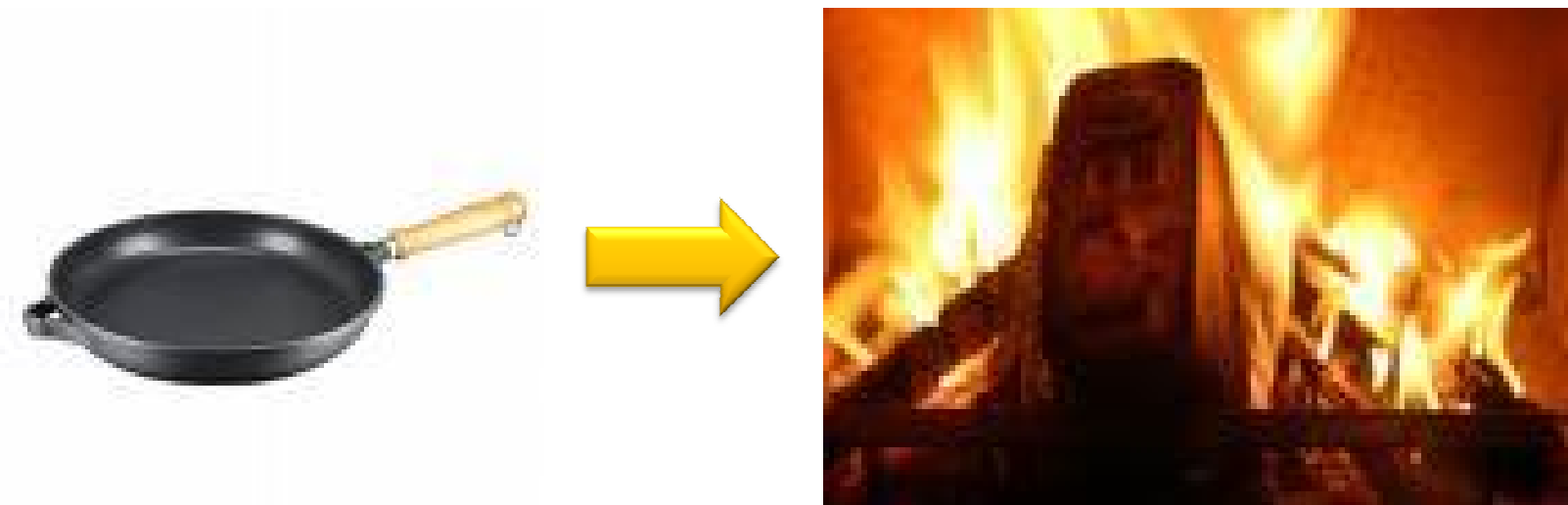


Part I: Project Management Issues

- Expectations
- Software Configuration
- Planning
- IT effectiveness



What you don't want to happen when you migrate



The driving force behind the migration

- What are reasons behind changing the application server ?
 - Costs
 - Product quality
 - Purchase policy
 - Security

The reasons have an impact on the migration project
→ expectation management

Product quality

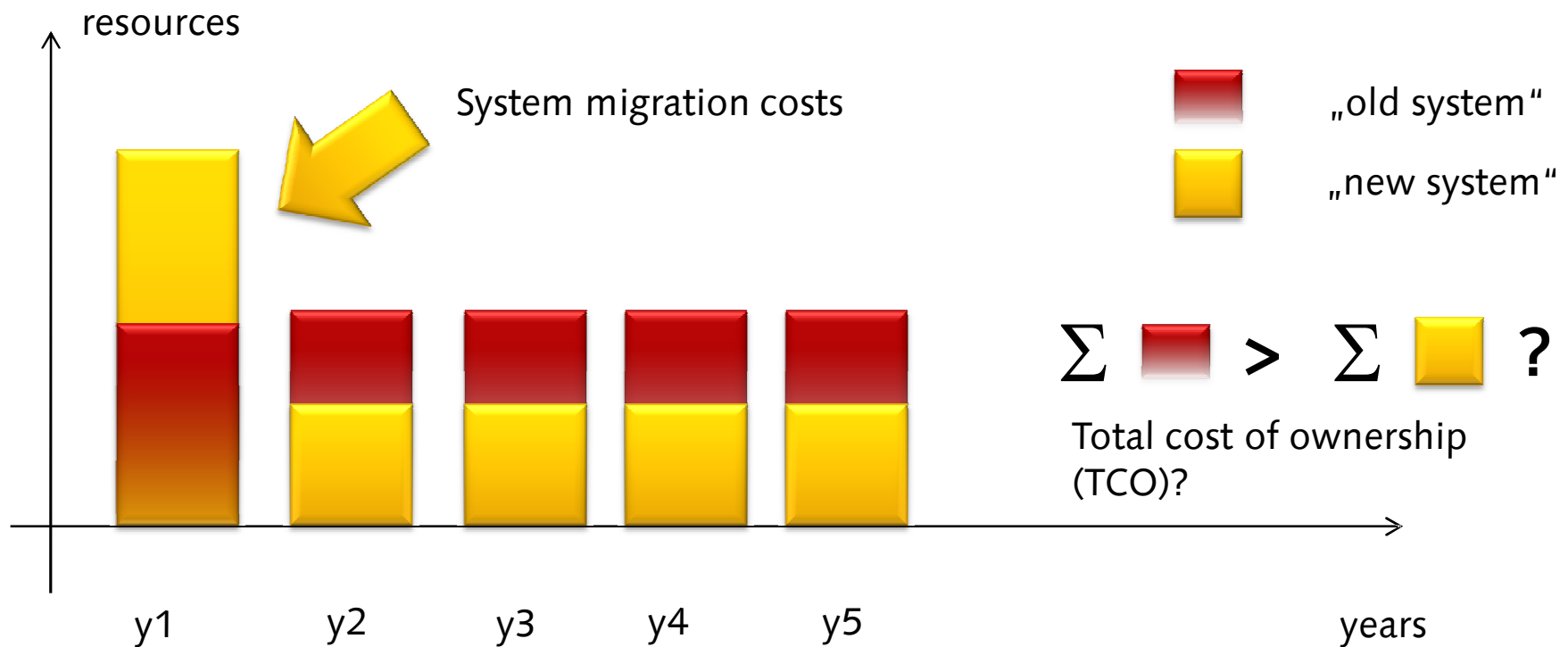
- Lack of features
 - What is used in addition to the J2EE Application Server?
 - BPM Components, Transaction Engines, O/R Mapping, Portals
- Compatibility issues
 - “Openness” to community standards
- Bugs
 - Big bugs?
- Performance
 - Is the App Server the performance bottleneck? → cf. Part two of talk
- Support
 - How well is the product supported?
- Operational requirements
 - Integration into monitoring tools
 - Clustering, fail-over, hot-deployment features

Purchase policy

- The dependency of a product goes way beyond the basic software license:
 - software maintenance
 - How is the economic scalability? Per cpu, per system, per ...?
 - distribution channels
 - Can you purchase the software directly from the vendor?
 - professional services
 - Is there a well established professional service organization? Are there any local subsidiaries, partners, third parties?
 - freelancer skills available in the market
 - Are there freelancers available?
 - Technology dependency $\leftarrow \rightarrow$ product dependency
 - How can a system be designed that all components can be bought from different vendors? (and still work)



Cost issues

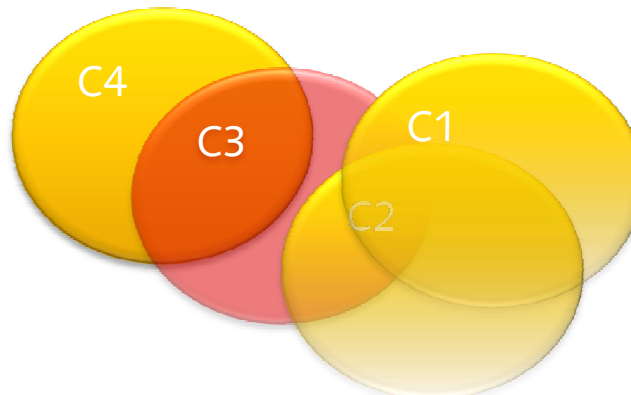
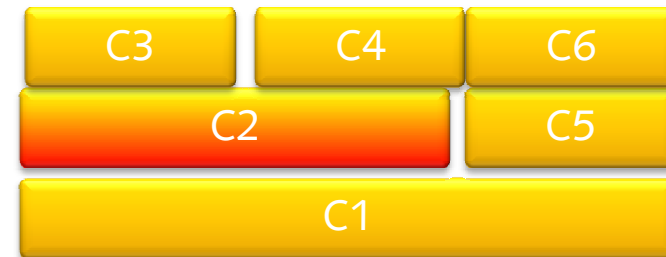
- Cost issues must always be put in context of the system's life cycle. License costs are paid once, support annually.



System architecture

- How does your system architecture look like?

-  System components to be changed
-  System components to remain unchanged



The nature of migration

- Migrating technologies in a system is to be seen as a part of life cycle management.
- The success in migrating technology components lies in a successful implementation of software configuration management (SCM).



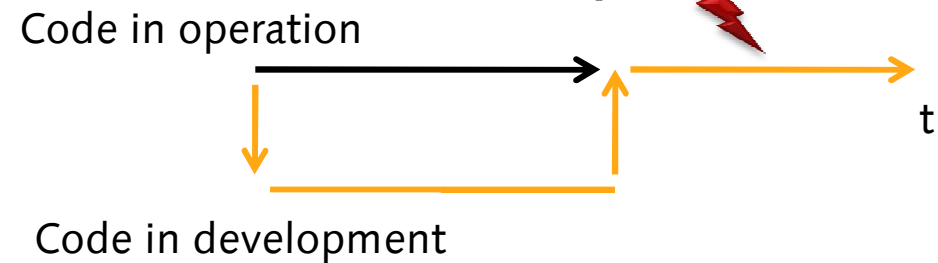
Migration = Change

Planning the system migration I

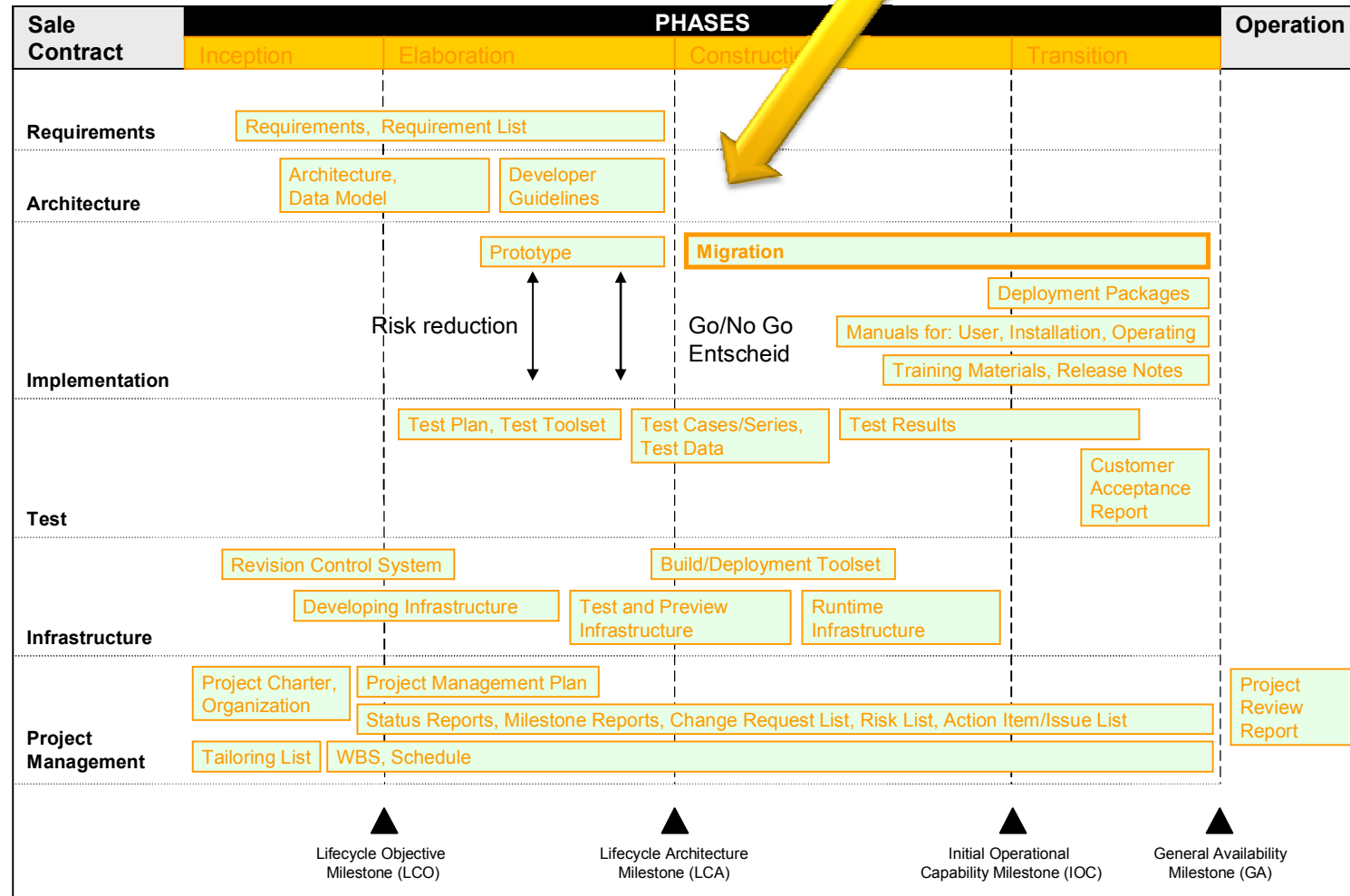
- What are the expectations?
 - These will give you the priorities
- Reduce risks
 - Analyze known differences between products
 - Check interdependencies
 - Make a prototype / pilot migration
 - Review pilot project with expectations

Planning the system migration II

- Implementation of migration
 - parallel deployment, „big bang“?
 - System testing
- Roll-Out
 - Partially, „big-bang“?
 - Fallback scenarios
 - Parallel op mode
 - → system management complexity



Project plan outline



Adaption of the Rational Unified Processes™ by Crealogix

On IT effectiveness

- Will your IT be more effective after the migration?
 - Better functionality to the end-user?
 - Smoother operational performance?
 - Costs?

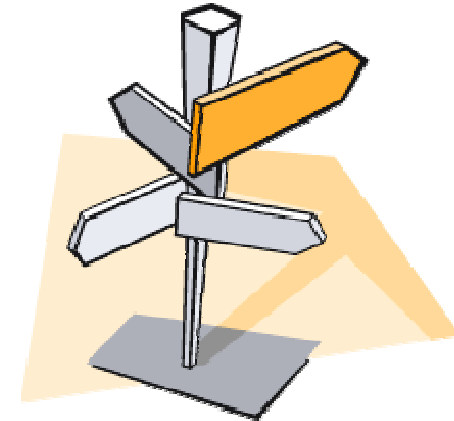
- → Life Cycle!

Summary

- An important success factor of a migration project is to get the right expectations and to set project priorities accordingly
- Costs are best to be seen in the context of total cost of ownership of all system components affected by the migration.
- Timing of the migration project should align smoothly with the system's life cycle

Part II: Technological Insight

- Migration Process
- Preparation
- Migration
 - Tools
 - Problems and Solutions
 - Support
- Validate
 - Performance, Tuning
 - System Management



All information in this presentation is mainly for JBoss 4.2 EAP and above.

Migration Process

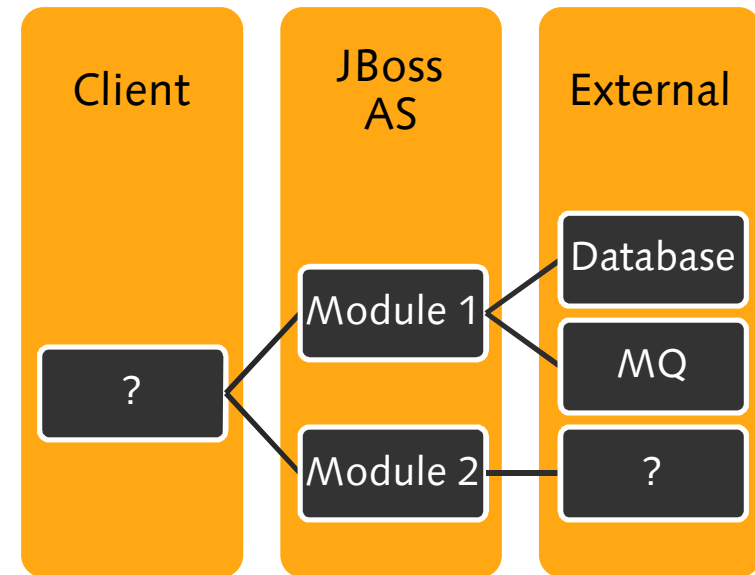
- Identify code or configuration which need modifications
 - Review your software
 - Prototype
- Perform migration
 - Setup JBoss EAP
 - Code modifications
 - Build process
- Validate
 - Systemtests
 - Monitor software



Preparation

Know your Application

- Access Points (Web, EJB Client,)
- Dependencies to third party libraries
- Use of any non-standard APIs?
- Clustering Support?



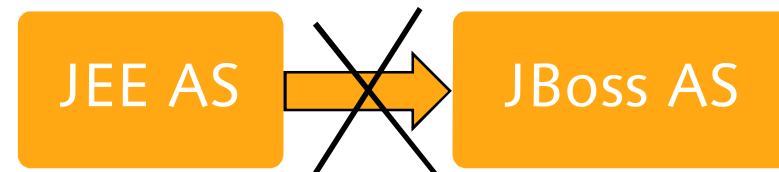
Does the migration impose other changes? Expect problems!

- Migrate to another JVM?
- OS

Application Server differences

Could not be migrated

- Precompiling EJB → Use Eclipse WTP/EJB3.0
- „System Management“-Console → wait for Embedded JOPR
- Automatic discovery of web-contexts → wait for mod_cluster



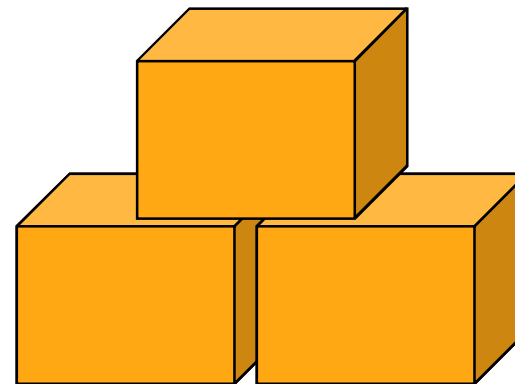
New options

- Precompiling of JSPs with Tomcat Compiler.
- Production ready EJB 3.0 Support since JBoss 4.2 EAP.



Setup JBoss Plattform

- Use production-configuration provided from JBoss EAP.
- Disable JMS if you not need it.
- Replace HSQL-DB.
- Security! Never run JBoss in public without some work.
- On general: other tunings are not needed.



Tools

Migration

- Automatic tools not (yet) available.
- Writing descriptors for JBoss is not the main problem.
- Dependency Analyzer (JBoss Tattletale, Maven, Ivy, Mavenizer)

Tests / Debug

- JBoss Tools for Eclipse



Tools – descriptor validation

- <http://www.jboss.org/j2ee/dtd/>
- <http://www.jboss.org/j2ee/schema/>



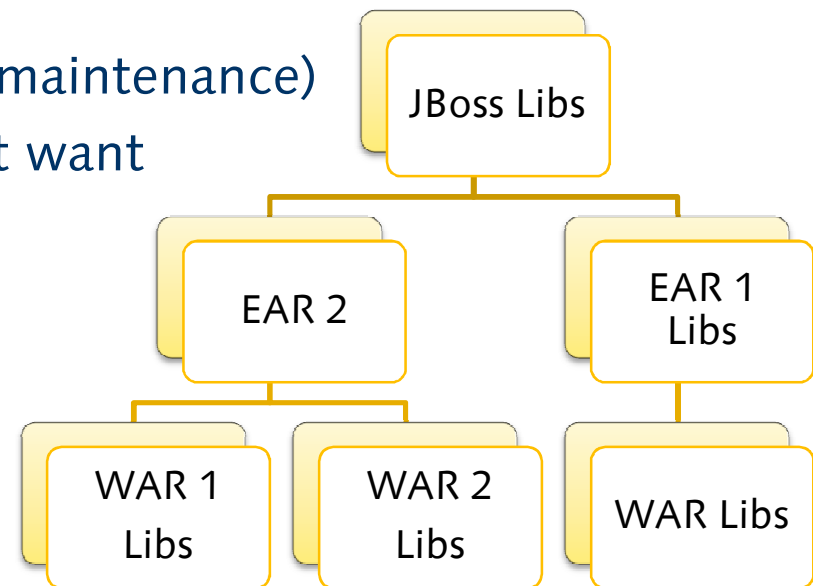
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jboss PUBLIC "-//JBoss//DTD JBOSS 4.2//EN" "http://www.jboss.org/j2ee/dtd/jboss_4_2.dtd">
<jboss>
  <enterprise-beans>
    <session>
      <wrong-name>EjbName</wrong-name>
      <resource-ref>
        <res-ref-name>mail/DefaultMail</res-ref-name>
        <jndi-name>java:/Mail</jndi-name>
      </resource-ref>
    </session>
  </enterprise-beans>
</jboss>
```

The screenshot shows an IDE window titled 'jboss.xml'. The XML content is displayed with syntax highlighting. A red 'X' icon in the left margin indicates a validation error. The error is located on the line containing the tag '<wrong-name>EjbName</wrong-name>', which is highlighted in blue. The rest of the XML structure is as follows:

Problems and Solutions I

Main Problems

- Classloading
 - Use scoped classloading for EARs is usually a good choice
 - Remove JBoss, JSE/JEE-Libs from the EAR/WAR.
- Log4j
 - We preferred scoped logging (better maintenance)
 - Use a RepositorySelector if you don't want scoped classloading

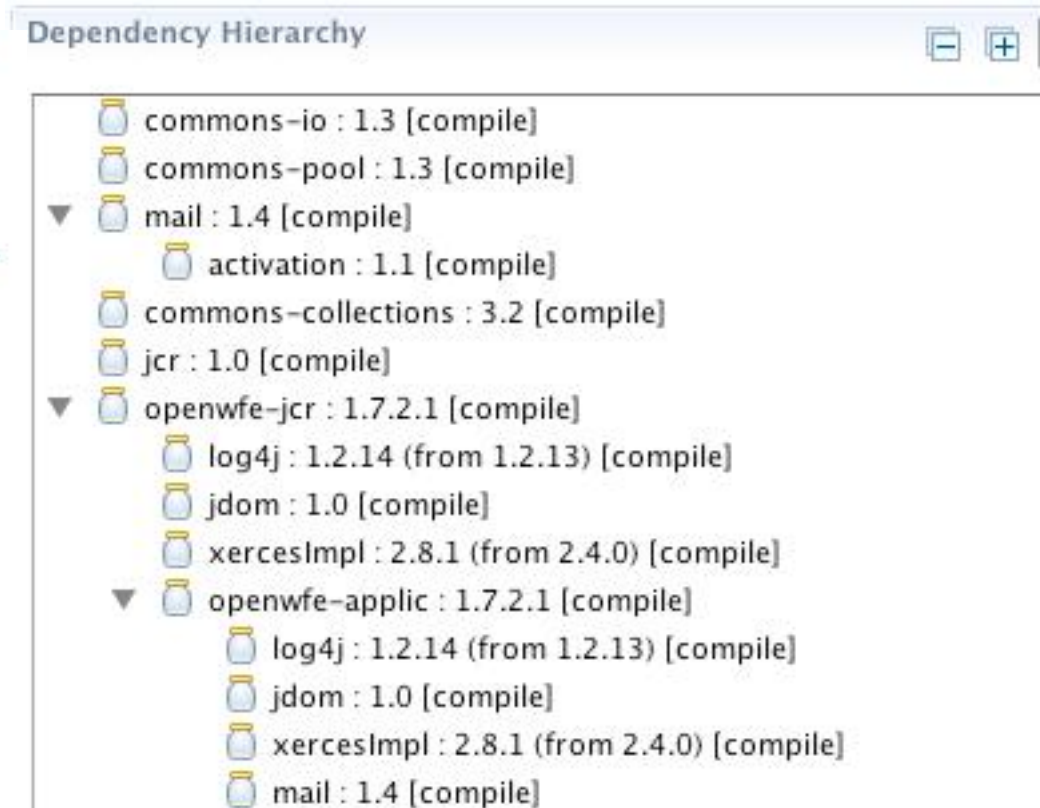


Problems and Solutions – Detect Classloading

```

<artifactId>magnolia-core</artifactId>
<name>magnolia-core</name>
<dependencies>
  <dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>1.3</version>
  </dependency>
  <dependency>
    <groupId>javax.mail</groupId>
    <artifactId>mail</artifactId>
    <scope>provided</scope>
    <version>1.4</version>
  </dependency>
</dependencies>

```

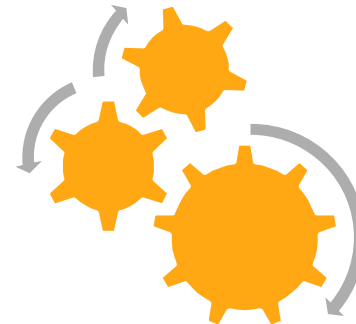


Usually you should always use a „provided“-scope for JEE/JBoss-Libraries.

Problems and Solutions II

Other Areas

- Performance: Usually depends on the JVM/OS. -> Do enough Performance-Tests if you also change the JVM/OS.
- JBoss has Solutions for Performance-Problems of the JVM: JBoss Serializing for Sun JVM
- Missing data/configuration after restart, because application writes to tmp/deploy-directory. -> expand module in the deploy-directory



Support

- Use JBoss Manuals from RedHat
http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/
- Read project documentations from www.jboss.org for modules in JBoss EAP like JBoss Remoting.
- Use Tomcat-Documentation for Web-Container.
- Good support from Red Hat / JBoss Support with an active subscription.
- Reading the source-code is always a good option.



Validation & System Management

- System testing is very important!
- Monitor your JBoss-Instances! A lot of general monitoring solutions could monitor JBoss per JMX, SNMP.
- From our experience JBoss runs very stable.
- Updates are not often needed, but that depends on your applications. If you use newer technologies like EJB 3 or WebServices you should update more often.

