

jBPM5 in Java EE 6

Marek Iwaszkiewicz
Pascal Schaerf
akquinet AG



BERLIN EXPERT DAYS

{join the talk}



Developer @ akquinet AG



Marek Iwaszkiewicz
marek.iwaszkiewicz@akquinet.de

JBoss Competence Center



Pascal Schaerf
pascal.schaerf@akquinet.de

JBoss Competence Center

Wer sind wir?



UI Design

OSGi

Trainings

Android

Web Portals

JBoss

Development

Java / JEE

Consulting

akquinet



User Experience

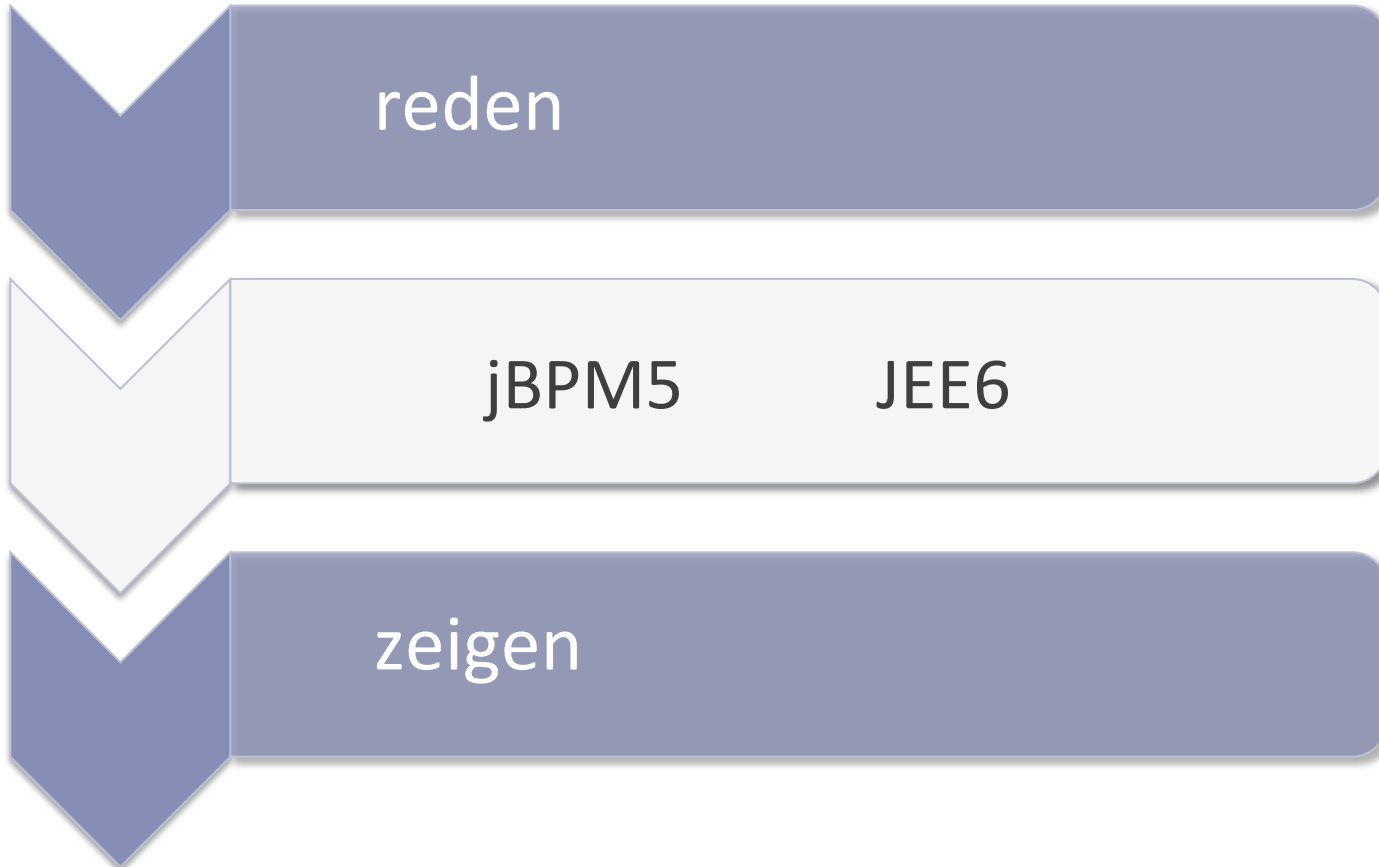
Modular Systems

Mobile Systems

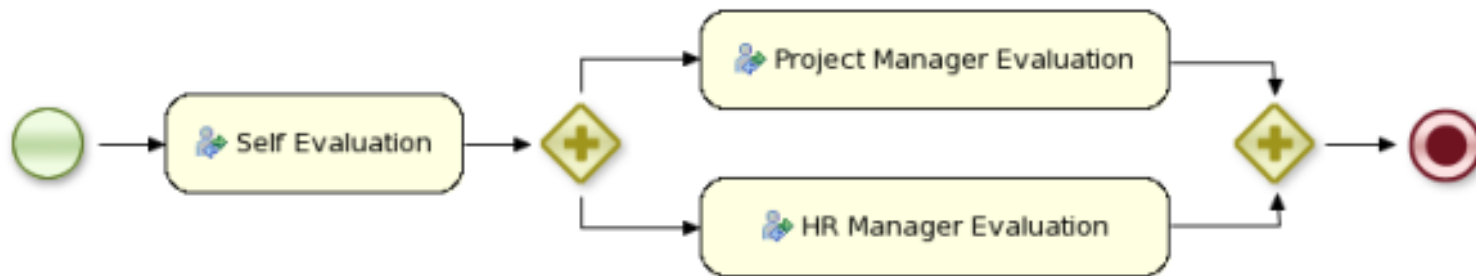
Integration

Open Source

<http://blog.akquinet.de>



Was ist jBPM5?

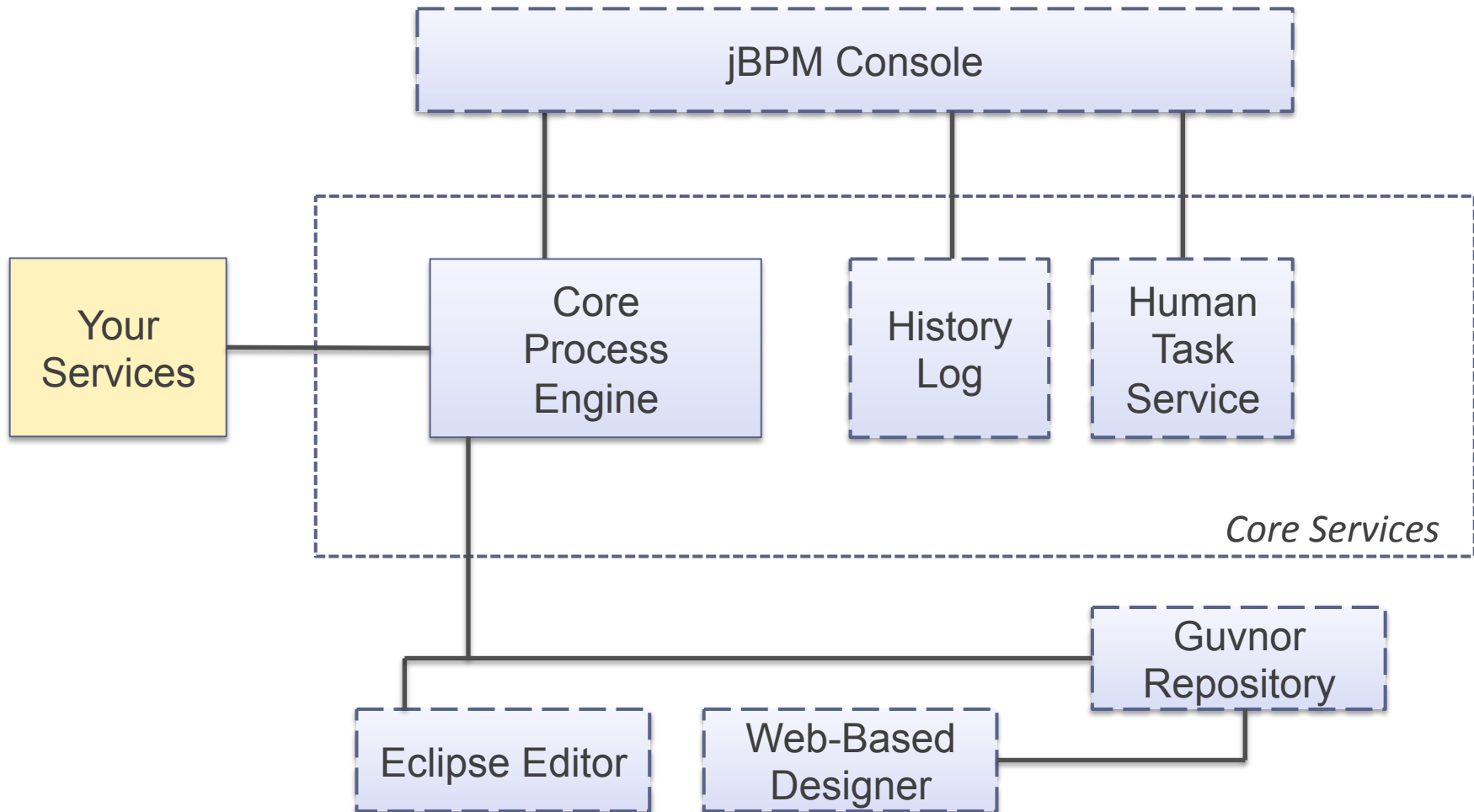


- BPMN2 Umgebung für Prozess-Modellierung und Ausführung
- BPM-Suite von JBoss
- Open Source – ASL (Apache Software Licence)
- Basiert auf Drools

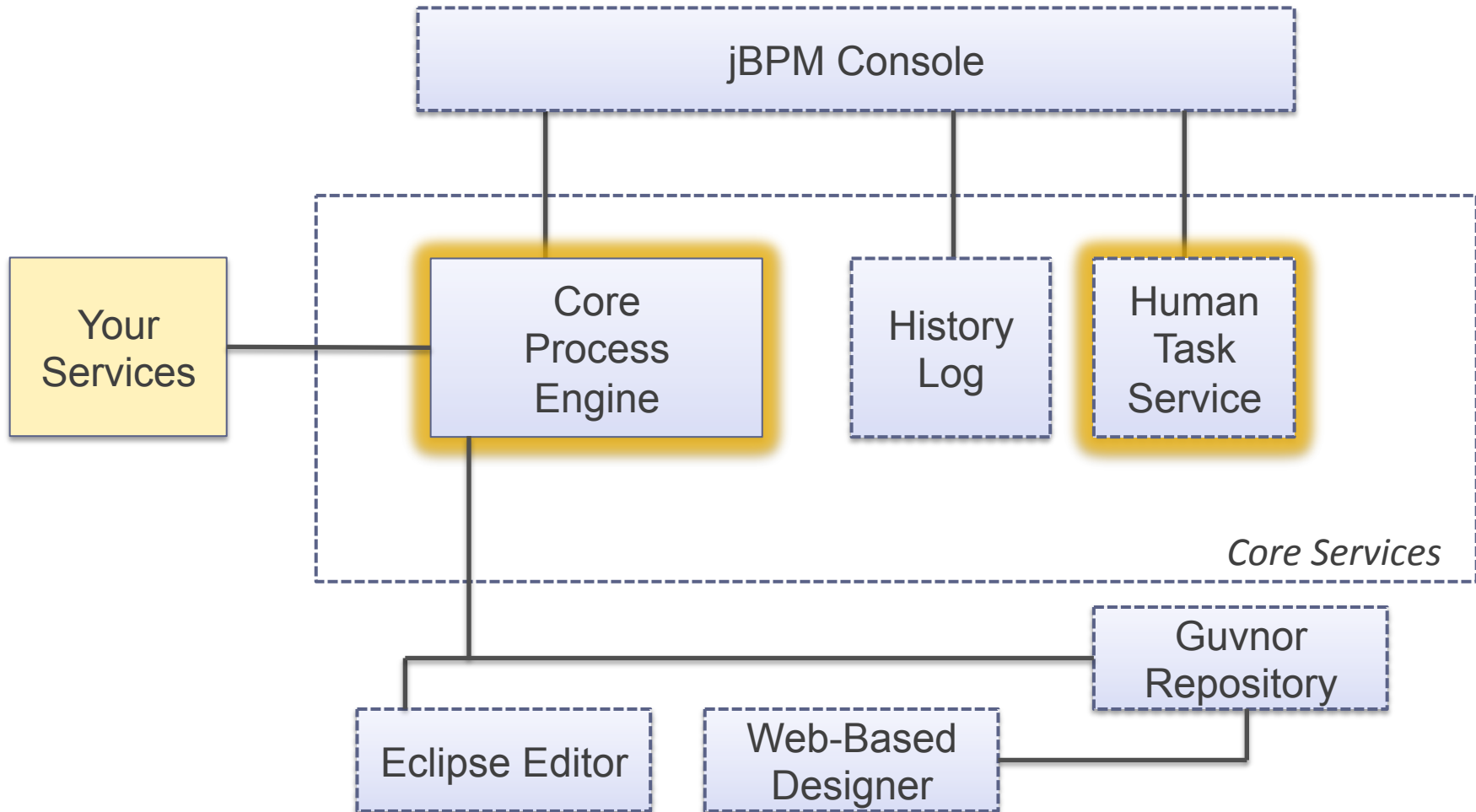
„beanspruchte“ Attribute

- flexible
- A lot of tools
- extensible workflow engine
- light-weight process engine
- Runs embedded or as a service
- Pluggable persistence with JPA
- JTA support
- Test support
- ...

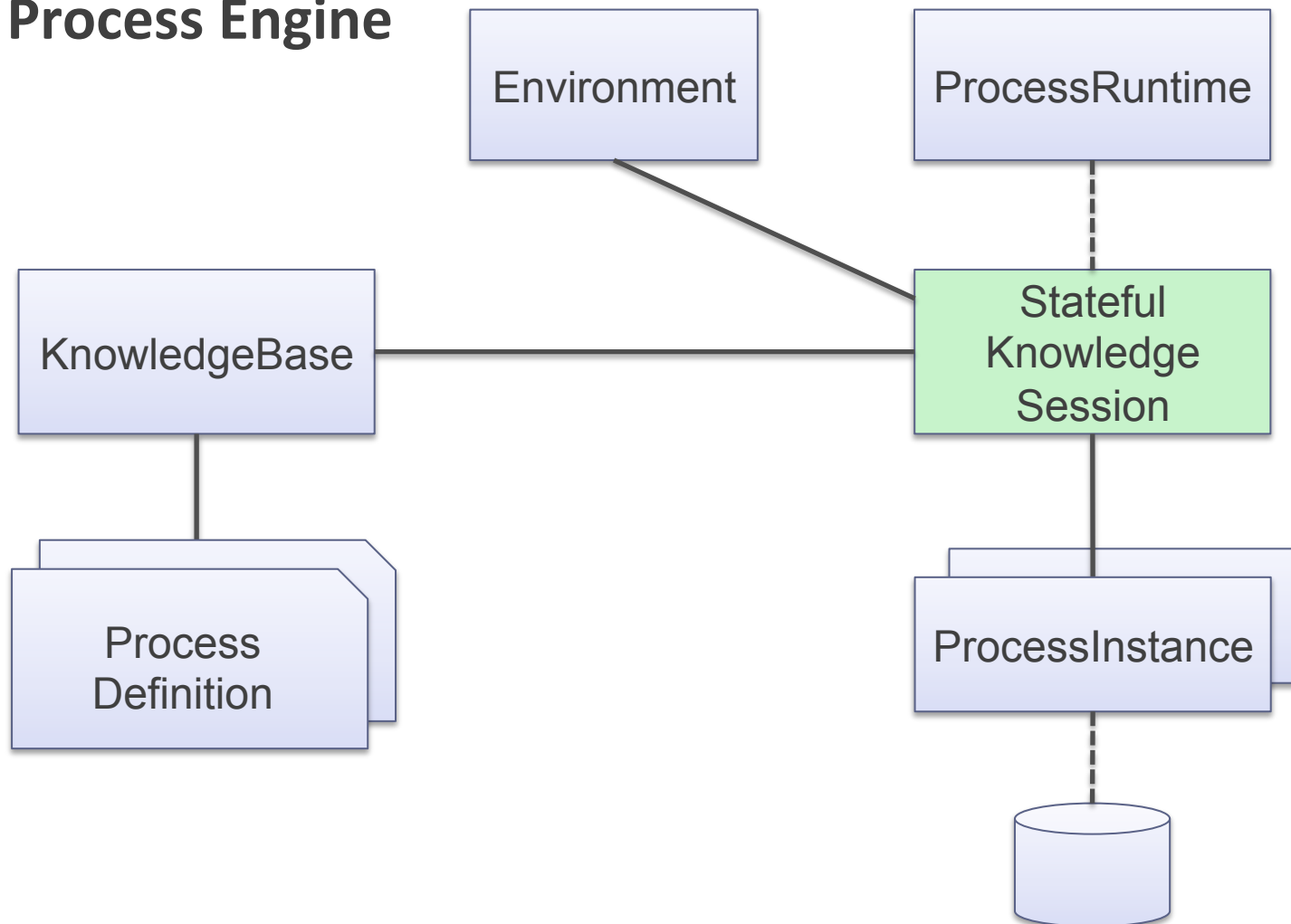
Komponenten / Makroarchitektur



Komponenten / Makroarchitektur



Core Process Engine



Code

KnowledgeBase erstellen und Prozessdefinition hinzufügen

```
KnowledgeBuilder kbuilder =  
    KnowledgeBuilderFactory.newKnowledgeBuilder();  
  
kbuilder.add(new ByteArrayResource(einProzess), ResourceType.BPMN2);  
  
KnowledgeBase kbase = kbuilder.newKnowledgeBase();
```

Code

Environment erstellen

```
@PersistenceContext(unitName = "org.jbpm.persistence.jpa")
private EntityManager em;

@Resource(mappedName = "java:jboss/TransactionManager")
private TransactionManager txm;

Environment environment = KnowledgeBaseFactory.newEnvironment();

JpaProcessPersistenceContextManager pcManager =
    new JpaProcessPersistenceContextManager(environment);

JtaTransactionManager droolsJtaTxManager =
    new JtaTransactionManager(null, null, txm);

environment.set(EnvironmentName.PERSISTENCE_CONTEXT_MANAGER, pcManager);
environment.set(EnvironmentName.TRANSACTION_MANAGER,
    droolsJtaTxManager);
```

Code

KnowledgeSession erstellen

```
KnowledgeBase knowledgeBase = ...
Environment environment = ...

StatefulKnowledgeSession kSession =
    JPAKnowledgeService.newStatefulKnowledgeSession(knowledgeBase,
        null, environment);

...

// Prozess starten
kSession.startProcess („com.acme.bpm.someProcessId“);
```

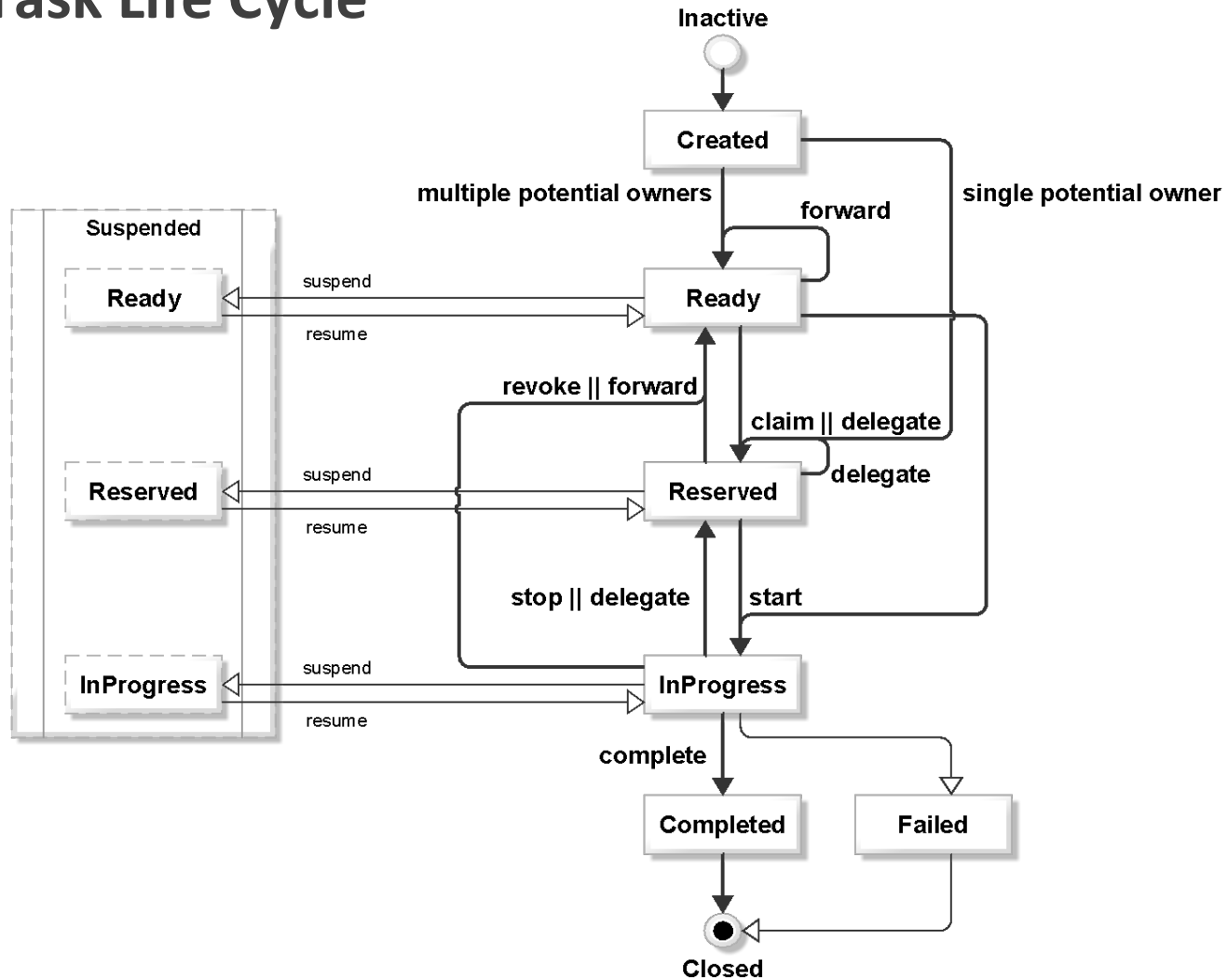
Human Tasks

- Arbeitsschritte (User Tasks), die vom Menschen ausgeführt werden
- Benutzer- und Rollenzuweisung
- In der Regel als Formular sichtbar
- Data-Mapping: Übergabe von Daten in und aus Tasks

WS-Human-Task Spezifikation

- *Rollenkonzept*
- *Lebenszyklus von Human Tasks*

Human Task Life Cycle



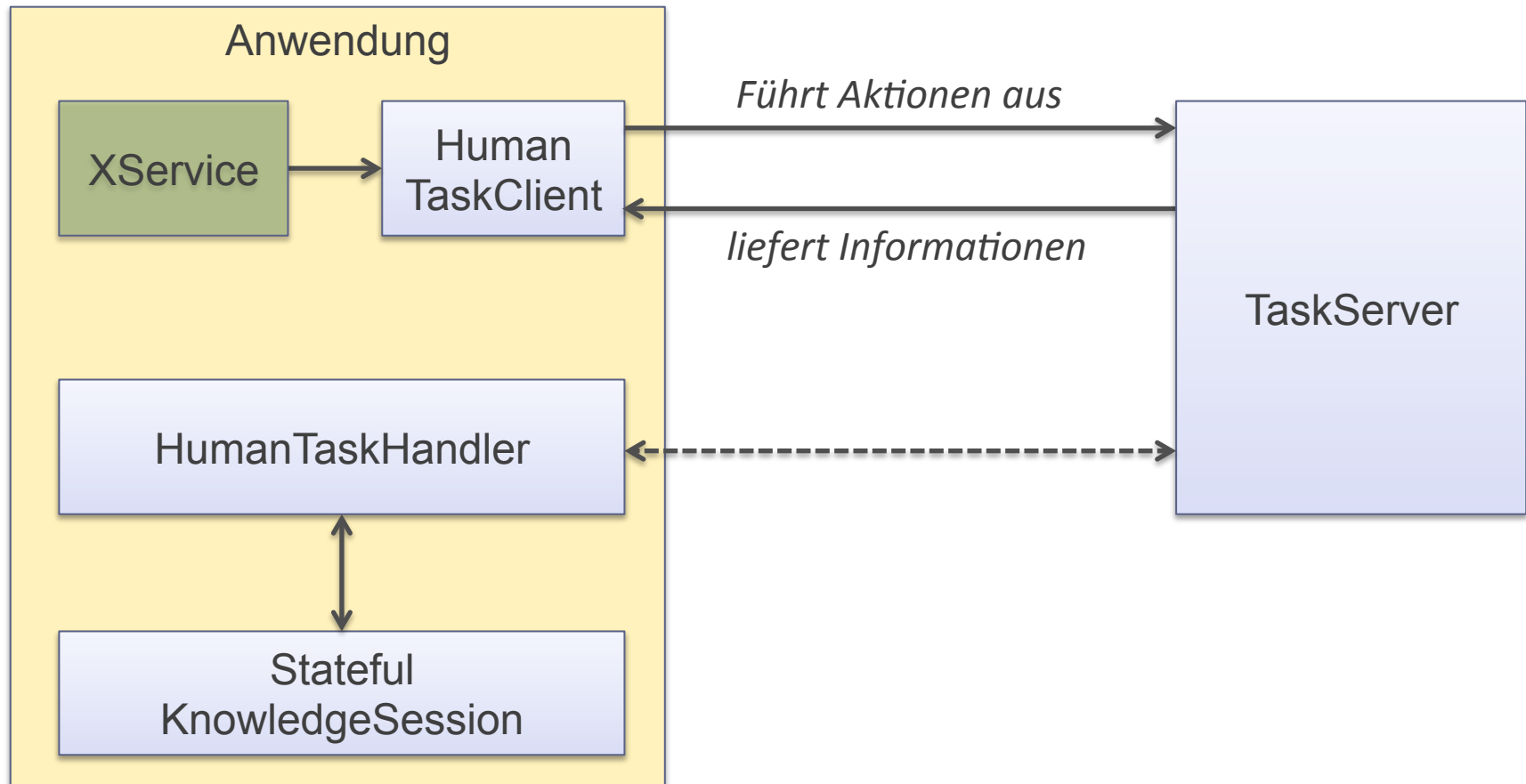
Human Task Service

- Task-Service konform zur Spezifikation
- Verwaltet Human Tasks
- Stellt Informationen zu Human Tasks bereit

- Schaltet die Prozesse **nicht** selbst weiter!!!
 - ➔ delegiert an Prozess-Engine

- Mehrere Ausprägungen:
 - Remote (Mina, JMS, HornetQ)
 - Local (embeddable Service)

Human Task Service



jBPM5 ...

- eingebettet in eine JEE6-Webanwendung
- auf dem JBoss AS 7
- JPA2.0
- JTA
- Testumgebung (inkl. Persistenz)

Es gab einige Probleme...

- Schlechte Dokumentation
- Teilweise gar keine Dokumentation (z.B. JTA-Anbindung)
- Keine Best-Practices
- Vieles sehr mühselig (z.B. Testumgebung)
- Kein JPA 2.0 Support

- ➔ tiefer Blick in die Sourcen notwendig
- ➔ eigene Patches für Fehler / Erweiterungen
- ➔ Dokumentation erweitern

ABER: Am Ende lief alles

Anwendung für Urlaubsanträge - Technologie-Stack

JBoss AS 7



jBPM5 (embedded mit JTA)



JPA2.0 / Hibernate 4



JSF2.0 / Richfaces 4



CDI / Weld

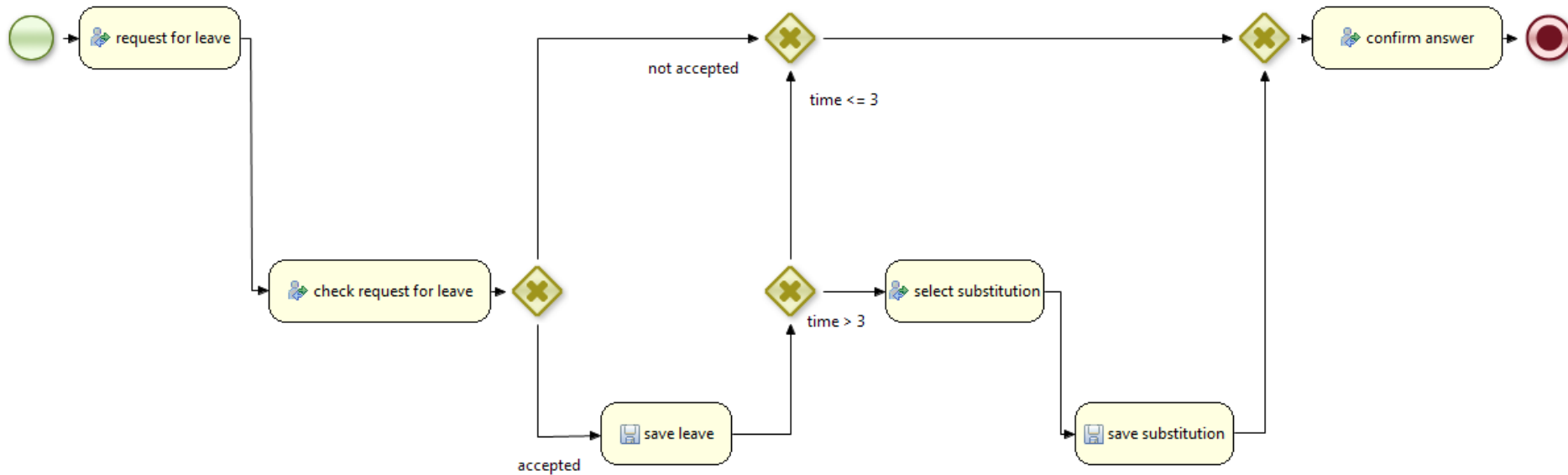


Seam 3

Maven 3



Anwendung für Urlaubsanträge – der Prozess



nicht fehlerfrei

schwache Doku

komplex → schwierig

Schwache Tools

verfrühte finale Version

modular

Doku wird schnell besser

mächtig

neue Tools in Aussicht

vielversprechend

EMail:

marek.iwaszkiewicz@akquinet.de

pascal.schaerf@akquinet.de

Tutorials/Blogs:

<http://blog.akquinet.de>

Source Code:

<https://github.com/akquinet>

Flux-Framework (jBPM5-CDI-Support):

<https://github.com/akquinet/flux>