CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Runtime Reconfiguration of J2EE Applications

Jasminka Matevska-Meyer, Sascha Olliges, and Wilhelm Hasselbring

Software Engineering Group, Department of Computing Science, University of Oldenburg, Germany

Conference on Software Deployment and (Re) Configuration, 28-29 October 2004

Introduction

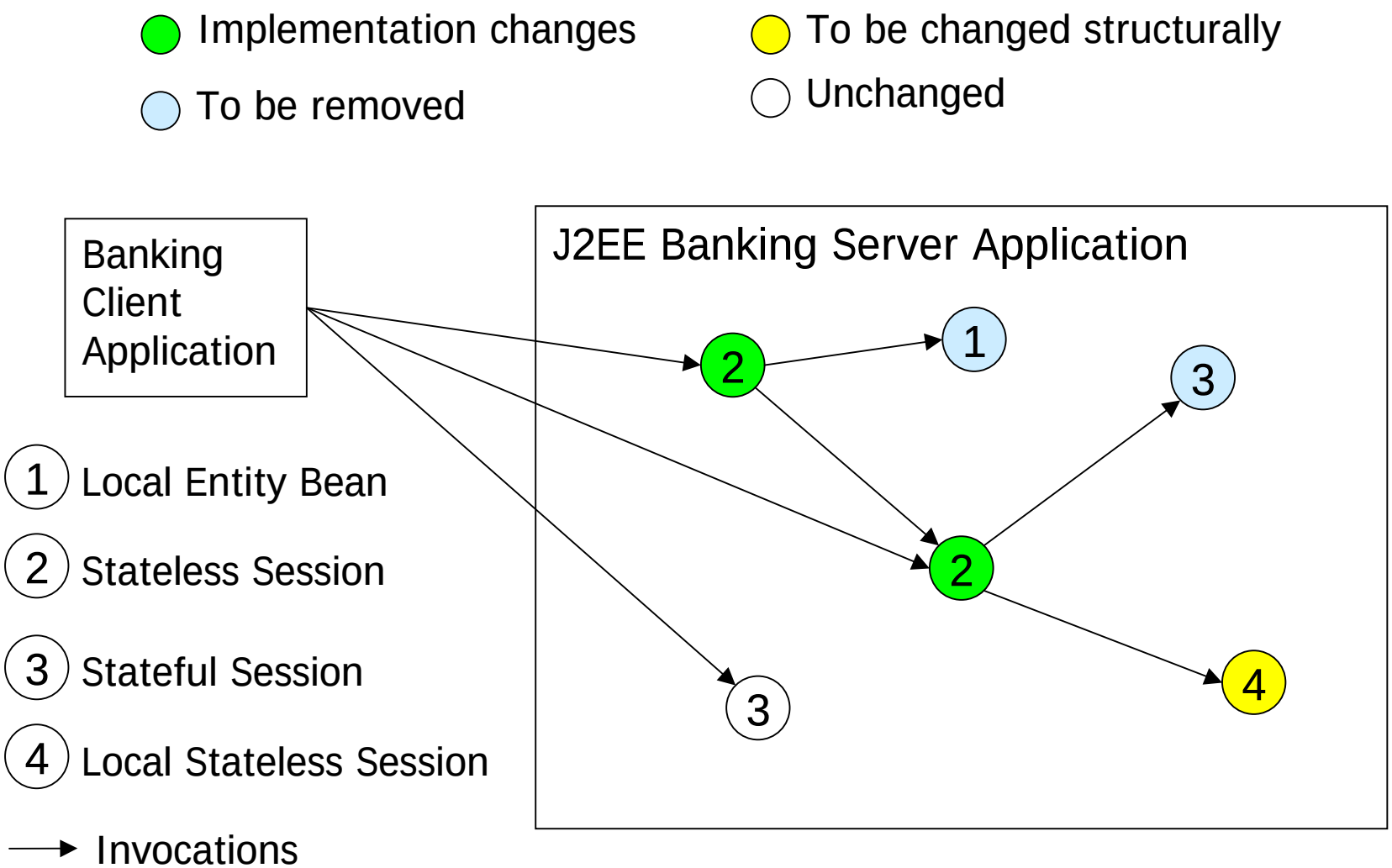
- Main concern
  - Reconfiguration of component-based systems at runtime
  - Implementation platform: J2EE
- Goals
  - Reducing the down-time of software systems
  - Maximising the set of available services
- Focus of this paper
  - Reconfiguration transparent to clients
  - Controlled runtime redeployment as an extension of hot deployment and dynamic reloading

CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Contents

CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Motivating Example



CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

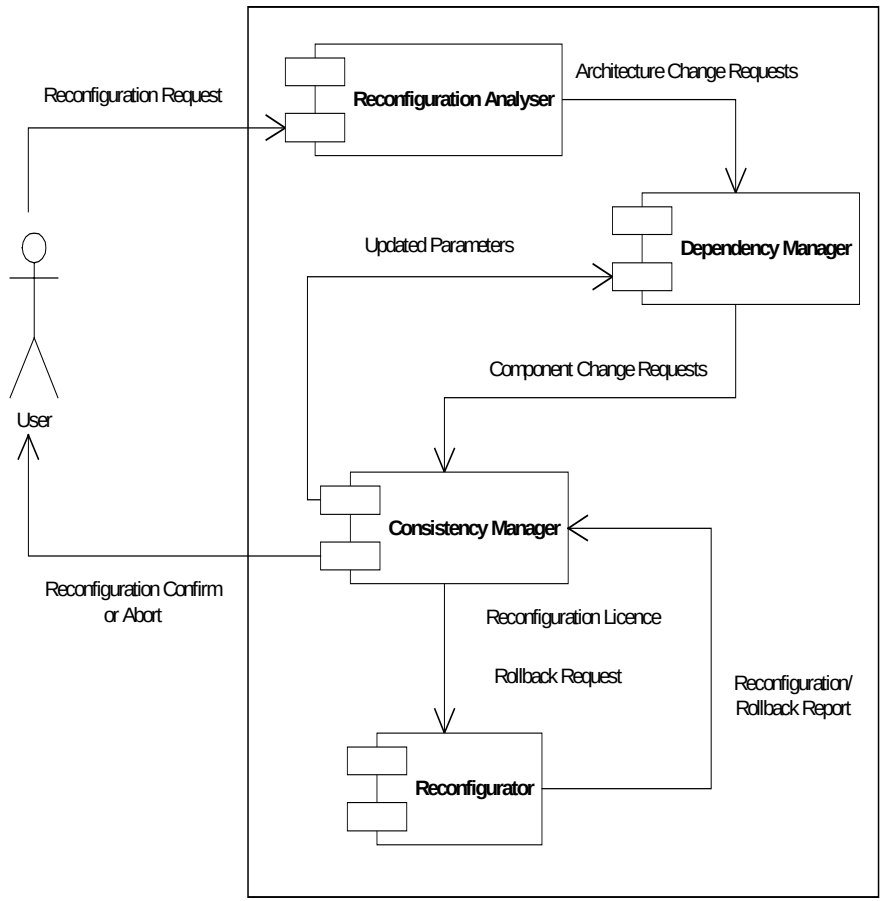
Runtime Reconfiguration

- Applying required changes to a running system
- Types of reconfiguration
  - Functional
  - Non-functional
  - Structural (considering both)
- An approach to enabling reconfiguration of component-based systems at runtime
  1. Analyse the running system for a particular time interval (at least the estimated reconfiguration duration)
  2. Create change-request-specific runtime dependency graphs
  3. Halt system parts based on the created graphs
  4. Observe the running system for the actual reconfiguration duration

CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Reconfiguration Manager - PIRMA

- Reconfiguration Analyser
  - Static dependency analysis
  - Component change identification
- Dependency Manager
  - Dynamic dependency analysis
  - Minimal dependency graph calculation
- Consistency Manager
  - Consistency checks
  - Determines reconfiguration point in time
  - Calculate Redeployment points
- Reconfigurator
  - Pause transaction processing at redeployment points
  - Exchange component byte code



CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Contents

Deployment API Functionality

- Enumeration of existing deployment units
- Start, stop, undeploy and deploy existing deployment units
- Creation and distribution of new deployment units
- Configuration of existing deployment units

- Redeployment transparent to users (optional)
  - No detailed concept in the specification
  - Not contained in any implementation yet
  - Focus of our project

Our Concept to Redeployment

- Hot deployment and dynamic reloading
  - Exchange Java byte code of component types at runtime
  - Essentially an Undeploy / Deploy operation sequence (container recreation)
- Our Extensions
  - Redeployment is a modification of the running container, rather than recreation
  - Therefore, the system remains reactive
- Controlled redeployment process steps
  1. Suspend transaction processing
  2. Perform module modifications
  3. Resume transaction processing

CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Contents

Redeployment Points and Discovery

Redeployment Point Discovery:

Redeployment points are enterprise bean methods that ...

- ... are configured to create, support (with no active transaction) or not support transactions
- ... are publicly available to client code

Redeployment Feasibility Check:

Redeployment point set per request...

- ... has to be complete, meaning it has to cover any execution path accessible from client code
- ... has to be stable. Transaction creation behaviour should not change at runtime

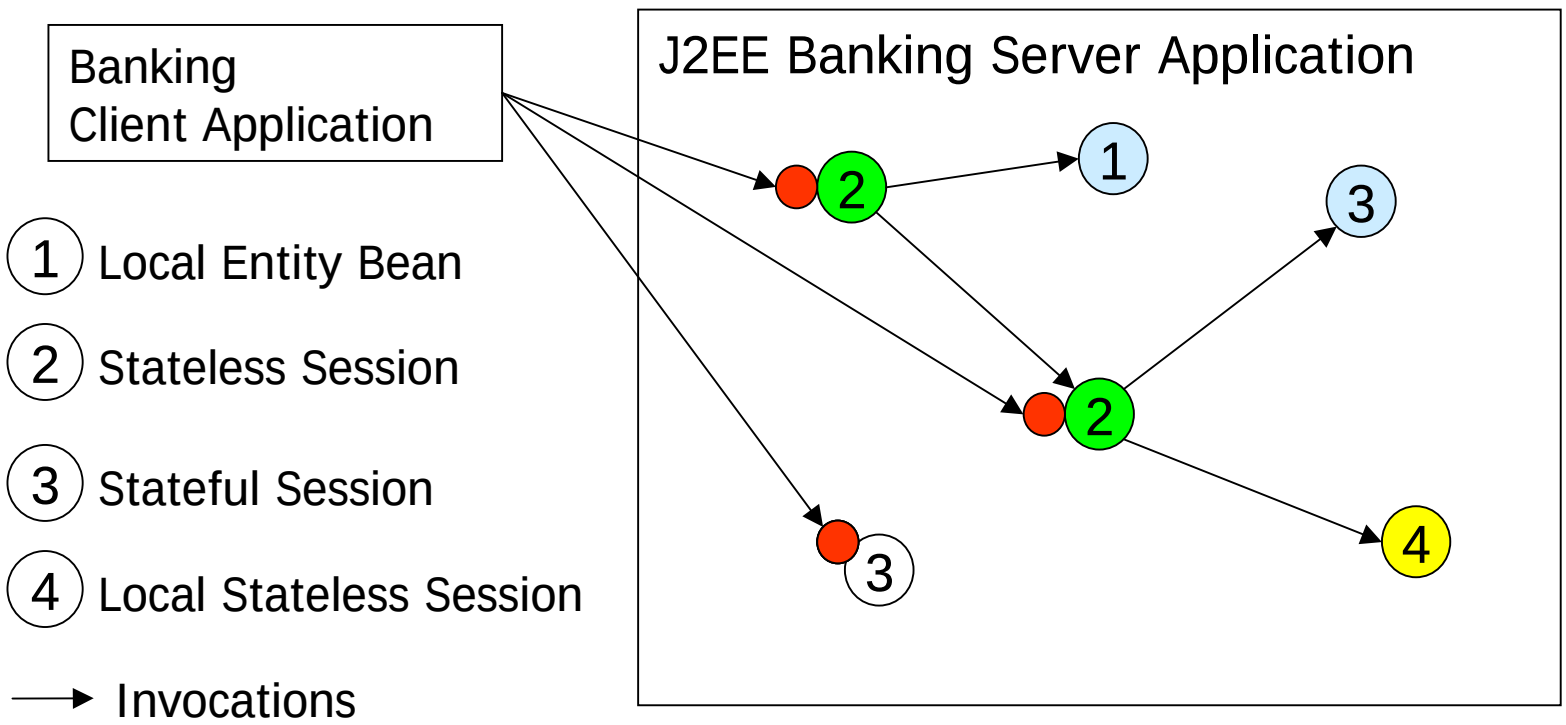
Redeployment Restrictions

- Provides-Interface of public EJB components may not be changed or removed
- Provides-Interface of Stateful Session EJB components may be changed, if the bean provider guarantees serialization compatibility
- Provides-Interface of Entity EJB components are dependent on the underlying data-schema and therefore changes at runtime should be avoided

CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

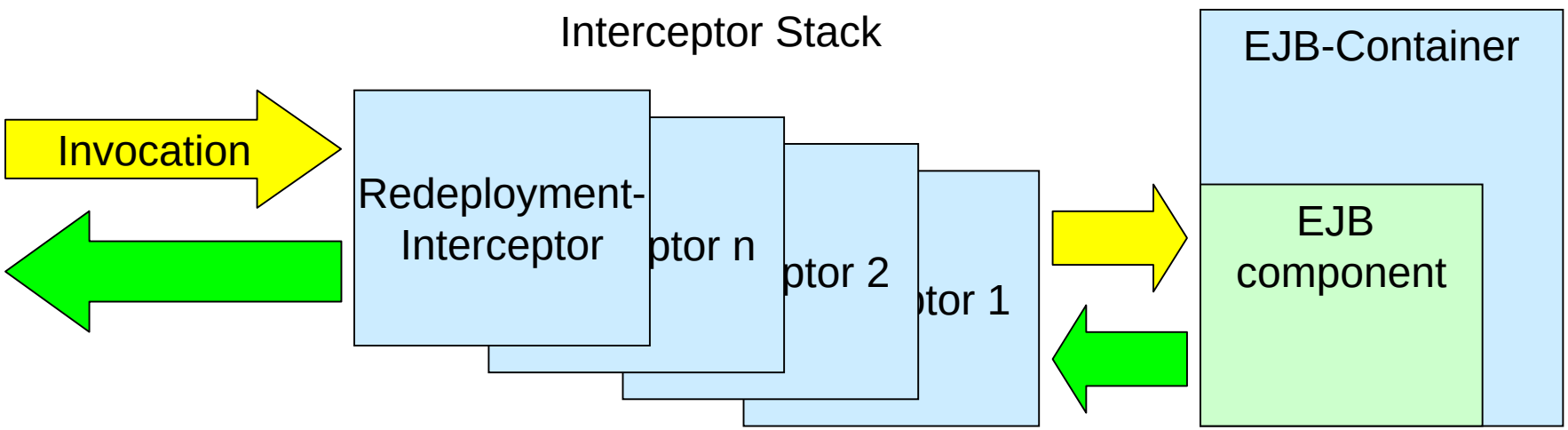
Redeployment Concept - Redeployment Points

- Implementation changes
- To be removed
- Redeployment Points
- To be changed structurally
- Unchanged



The JBoss Interceptor Stack

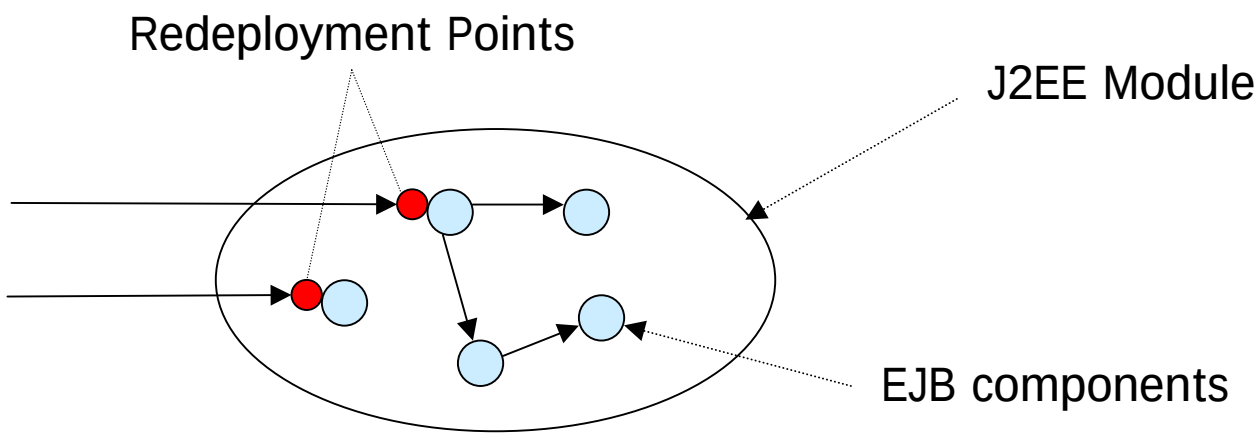
- Each single EJB component is hosted inside an EJB Container
- Each Container is configured with an interceptor stack
- Interceptors in JBoss implement non-functional attributes of EJB components as declared in the deployment descriptor



CARL VON OSSIETZKY universität OLDENBURG
FAKULTÄT II
INFORMATIK, WIRTSCHAFTS- UND RECHTSWISSENSCHAFTEN

Redeployment Interceptors

- MonitoringTxInterceptorCMT
  - Extension / Modification of the original CMT interceptor
  - Logs transactional behaviour to support redeployment point discovery
  - Suspend transaction processing at redeployment points
  - Trigger replacement of component class definition



Current Work / Open Issues / Future Work

- Current work
  - Minimal runtime dependency graph calculation
  - Designing scenarios for reconfiguration using design patterns
  - Implementation of the Redeployment Interceptors

- Open Issues / Future work
  - Predicting optimal point in time for redeployment to take place
  - Handling long-running transactions
  - Handling loop-back calls efficiently
  - Evaluation by larger case-study

Contributions

- Controlled runtime redeployment as an extension of hot deployment and dynamic reloading
- Redeployment is a modification of the running container, rather than recreation
- Therefore, the system remains reactive
- Reconfiguration is transparent to clients
- This fulfils the requirements of the Deployment API optional functionality