

Entwurfsmusterkatalog für Komponentencontainer

Diplom

Zwischenbericht

Irina Soudnik

Gliederung

- Einleitung
- Containerdefinition
- Vorgehensweise
- Beispiel: Referenz-Implementierung von SUN
- Ausblick

Aufgabenstellung

- Auswahl der zu untersuchenden EJB oder CCM konformen Komponentencontainer
 - Quellcode
 - Dokumentation
- Untersuchung ausgewählter Implementierungen bzgl. der zugrundeliegenden Architektur- und Entwurfsmuster
- Beschreibung der Muster in der Form eines Musterkataloges

Probleme

- Containerdefinition
 - unterschiedliche Komponentenmodelle
 - Abgrenzung von den anderen Bestandteilen der Komponentenmodelle
- Vorgehensweise
 - die Klassenbeziehungen zu analysieren
 - die bestehenden Muster zu identifizieren
 - aber*
 - *die Menge der zu untersuchenden Klassen ist groß*
 - *die Zeit ist knapp*

Vorgehensweise

- Prinzipien
 - ausgehend von der Containerfunktionalität
 - auf unterschiedlichen Abstraktionsniveau zu untersuchen und darzustellen
 - bei jedem Schritt bekannte Muster und Musterkandidaten zu identifizieren

Vorgehensweise (2)

- Referenz-Implementierung
 - grobe Architektur
 - detailliertere Betrachtung der bestimmten Bestandteilen der Implementierung
- Weitere Implementierung(en)
 - Untersuchung die Lösungsansätze nur für die ausgewählten Problembereiche
 - *werden die Musterkandidaten zu echten Mustern?*

Containerdefinition

- Beschreibung aus unterschiedlichen Sichten
- Container-Komponente
 - Installation
 - Generierung der Stellvertreterklassen
 - Behandlung der Descriptoren
 - Lebenszyklusverwaltung
 - Pooling
 - Methodenaufruf
 - Interzeption
 - Unterstützung der Persistenz der Komponenten, Transaktionsverwaltung, Sicherheitsüberwachung und Realisierung des Namenraums

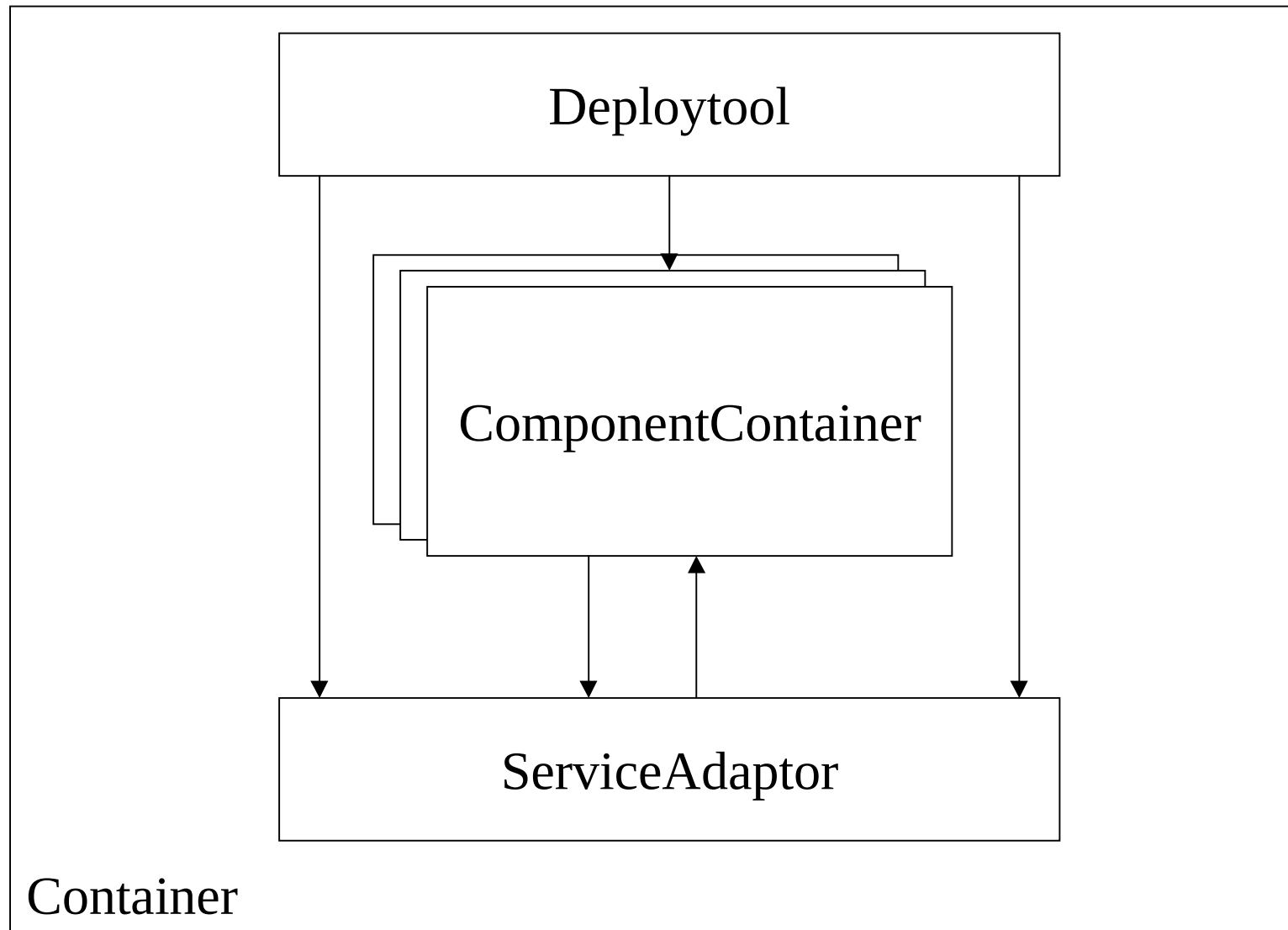
Containerdefinition (2)

- Container-Klient
 - „stubs“-Generierung
 - Erzeugung und Verwaltung der Objektreferenzen
 - Server-Absturz
- Container-Server
 - *nicht unmittelbar mit dem “Leben“ der Komponenten verbundene Probleme*
- Container-Dienste
 - Nutzung der standardisierten Schnittstellen von JNDI, JTS, CORBA, RMI, Datenbanksysteme usw.

Referenz-Implementierung von SUN

- J2EE Platform-Architecture
 - Komponenten: **EJBs**, Servlets und JSPs
 - Technologies: JNDI, JTS, RMI, CORBA, JDBC...
- J2EE Referenz-Implementierung von SUN
 - J2EE **Application-Server**, Web-Server, relationale Datenbank, J2EE APIs, Entwicklungs- und Installationswerkzeuge
 - kostenlos und als source code verfügbar
 - gut zu handhaben
 - + folgt genau der Spezifikation
 - nicht nur EJB-Container
 - SDK 1.2.1

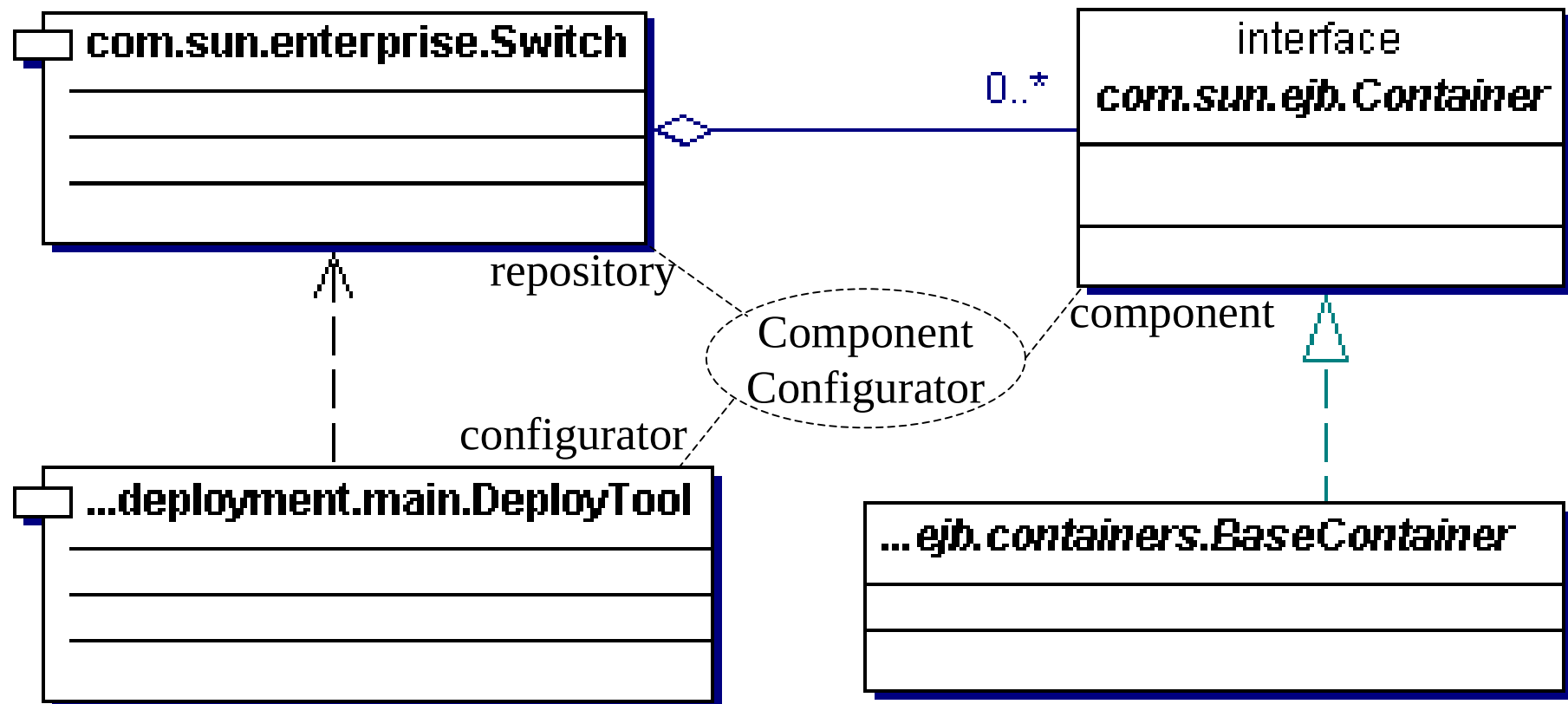
EJB-Container von SUN: Struktur



EJB-Container von SUN: Struktur

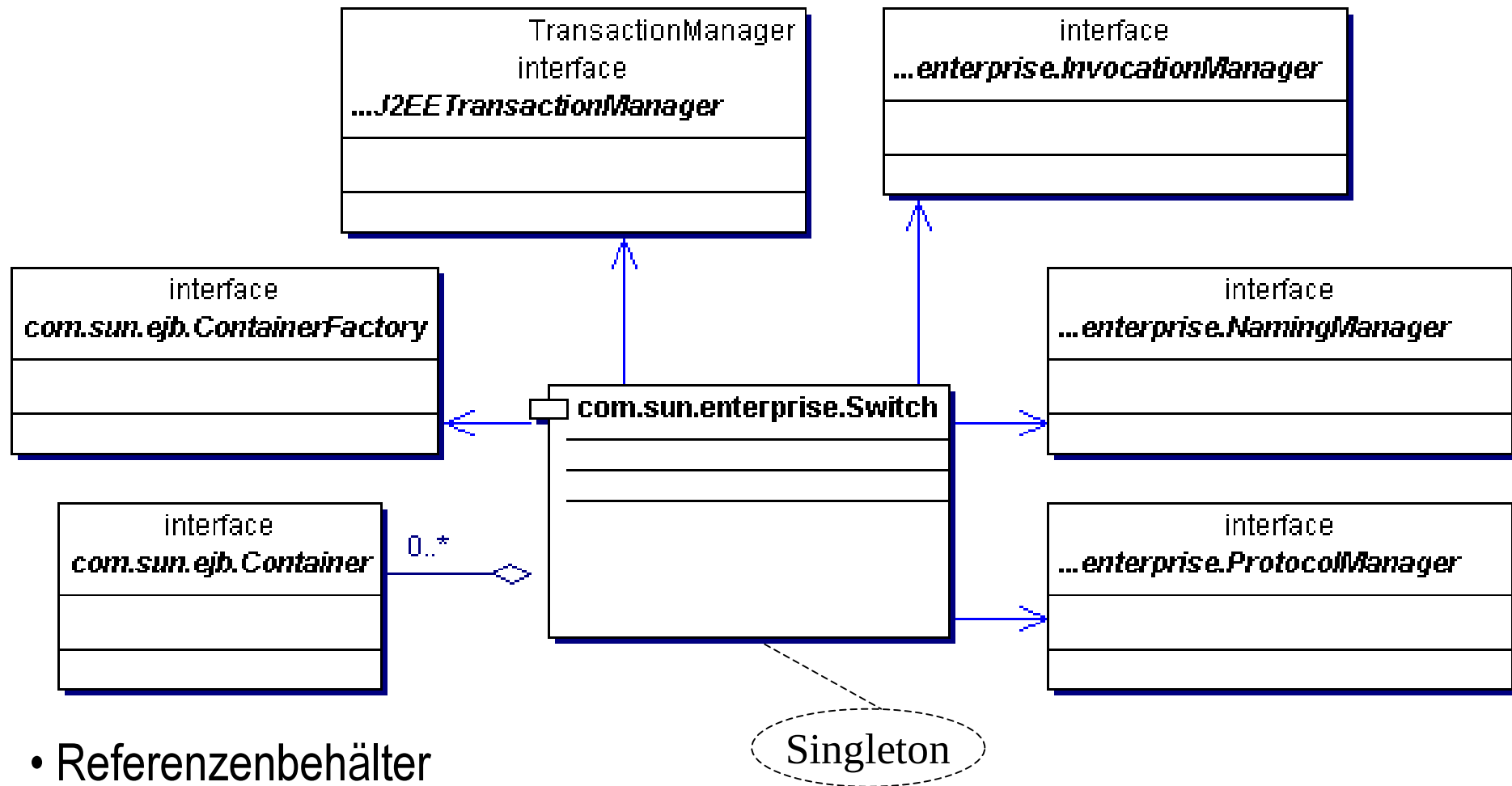
- Deploytool *als Komponenten-Installator*
 - Generierung der Wrapper-Klassen
 - Kreieren der Komponentencontainer
 - Behandlung der Descriptoren
- Komponentencontainer
 - Funktionalität im Rahmen des Komponentenkontrakts
- ServiceAdapter
 - Verwaltung der Komponentencontainer
 - Bereitstellung der Schnittstellen, die Zugang zu den Diensten ermöglichen und erleichtern

EJB-Container von SUN: Installation



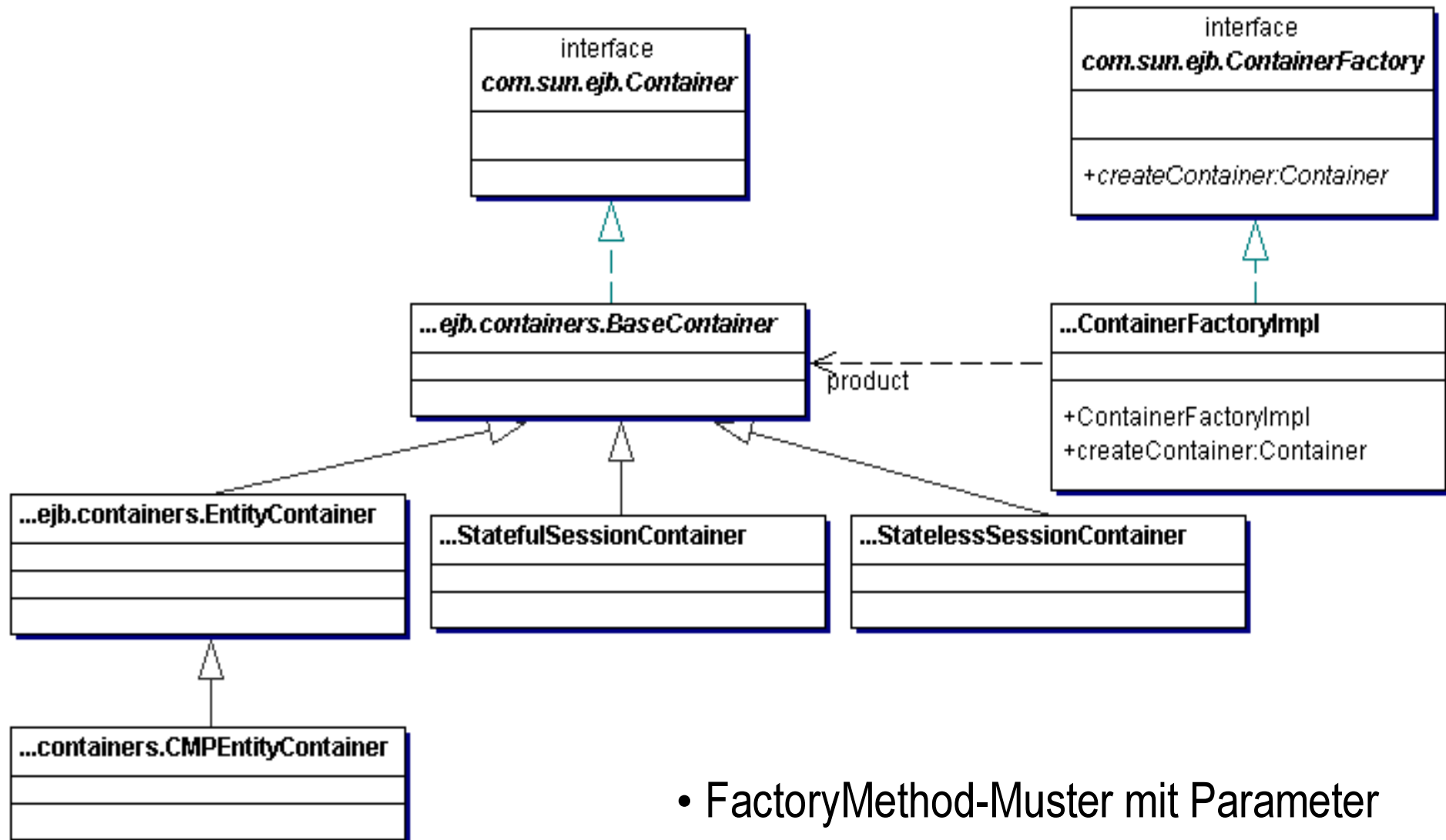
- ContainerConfigurator-Variante zu ComponentConfigurator-Muster
 - Repository verwaltet die Container nicht
 - Descriptoren als Konfigurationsanweisungen zu Laufzeit

EJB-Container von SUN: Laufzeitumgebung



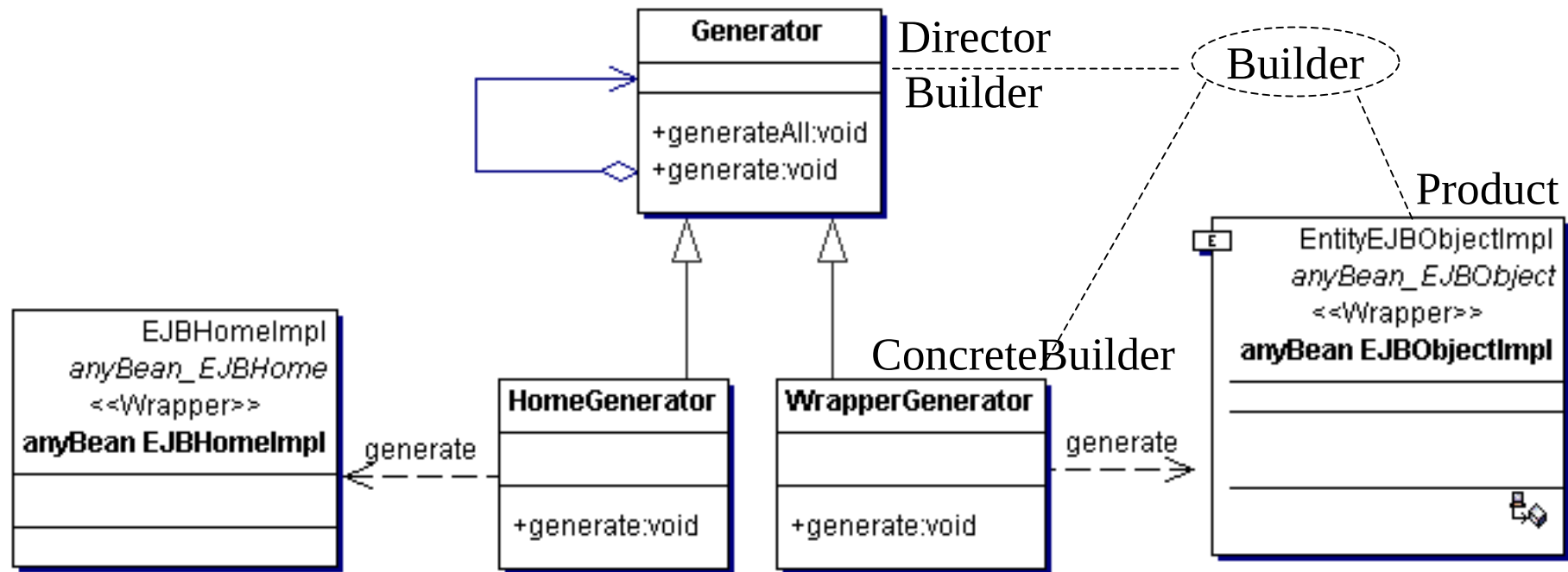
- Referenzenbehälter
- keine Verwaltungsfunktionen
- Manager sind quasi Fassaden

EJB-Container von SUN: Kreieren des Containers



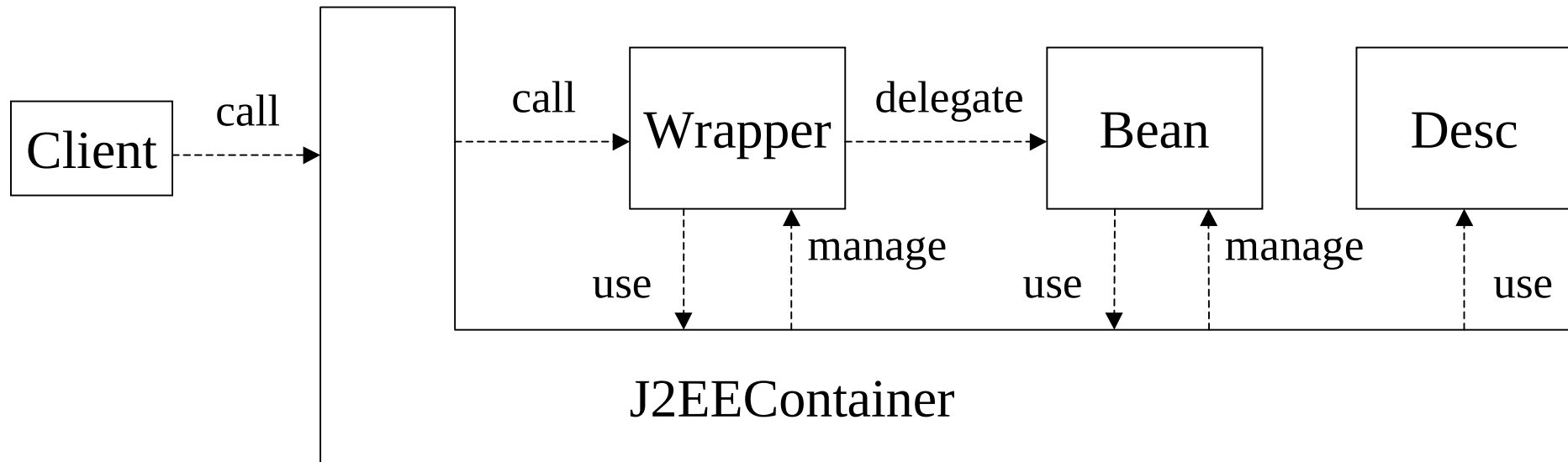
- FactoryMethod-Muster mit Parameter

EJB-Container von SUN: Generierung



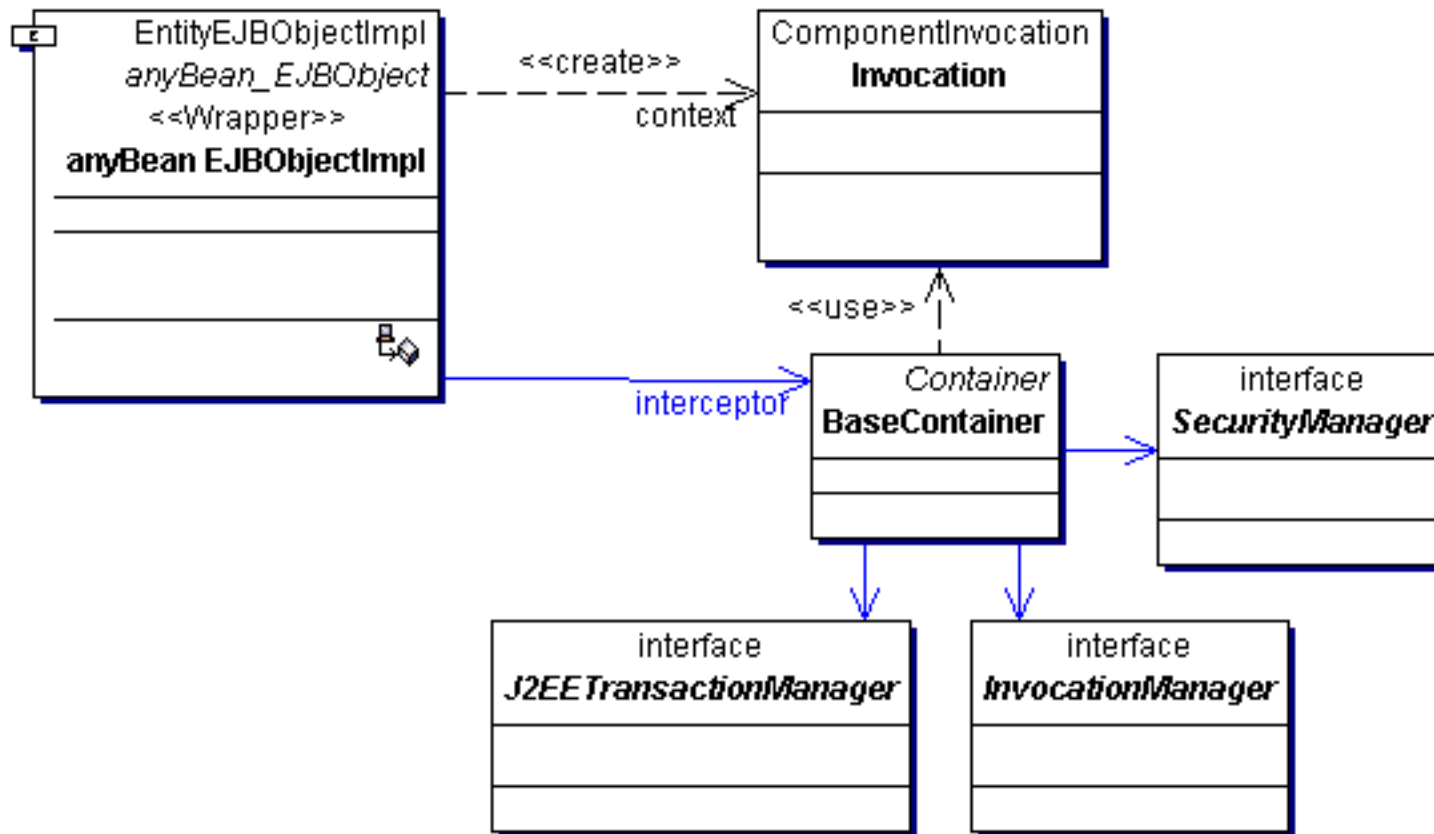
- Generierung der Wrapper-Klassen nach Builder-Muster
 - Vereinigung der Director- und Builder- Funktionen

EJB-Container von SUN: Interzeption



- Proxy-Variante des Interception-Musters

EJB-Container von SUN: Interzeption (2)



- Referenz zum Bean
- Anschaltung der Sicherheits-, Transaktions-,... Mechanismen
- Invocation-Objekt als Kontext
- keine Interceptor-Schnittstelle für Container

Ausblick

- Weitere Implementierung(en) untersuchen
 - EJB oder CCM konforme
 - nur die Problembereiche oder Architektur
- Iteration
 - Vergleich auf der gröberen Abstraktionsebene
- Katalog
 - Sammlung von Mustern, wobei jeder einzelnen Muster die Lösung eines containerspezifischen Problems darstellt
 - Muster: neue Muster, bekannter Muster, Lösungsansätze

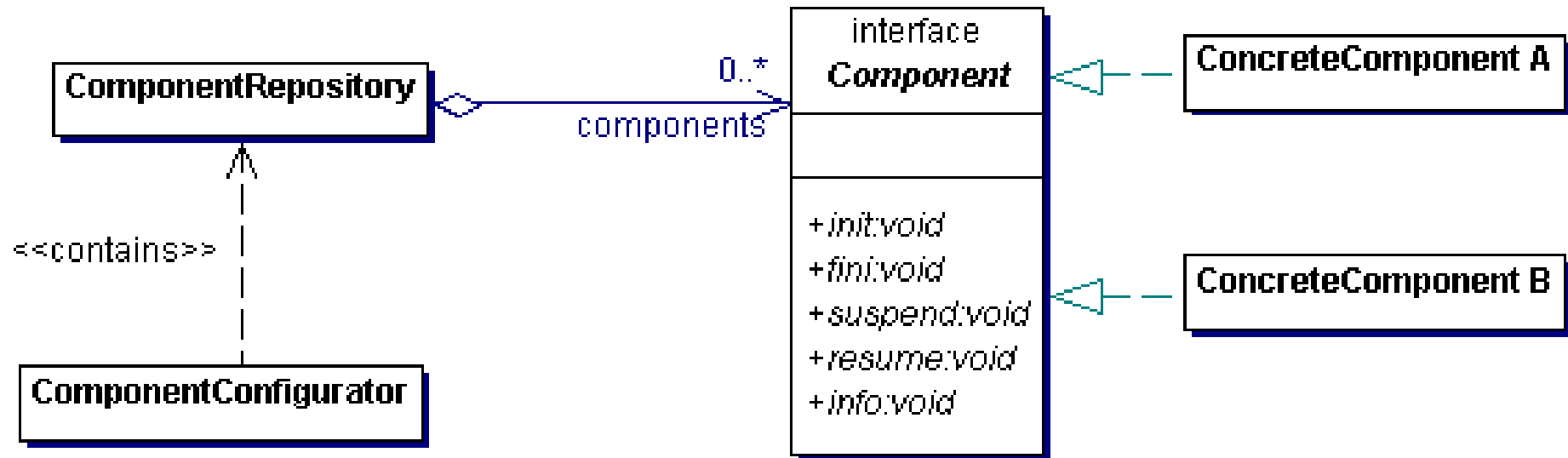
Quellen

- Enterprise JavaBeans Specification, Version 2.0, Proposed Final draft 2, April 2001
- „CORBA 3.0 New Component Chapters“, OMG Dokument ptc/99-10-04, OMG, Oktober 1999
- E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns, Addison Wesley 1994
- F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: Pattern-Oriented Software Architecture, Volume 1, Wiley 1996
- D.Schmidt, M. Stal, H. Rohnert, F. Buschmann: Pattern-Oriented Software Architecture, Volume 2, Wiley 2000
- Sun Java Center J2EE Patterns: J2EE Patterns Catalog
- T. Mowbray, R. Malveau: CORBA Design Patterns, Wiley 1997

Quellen (2)

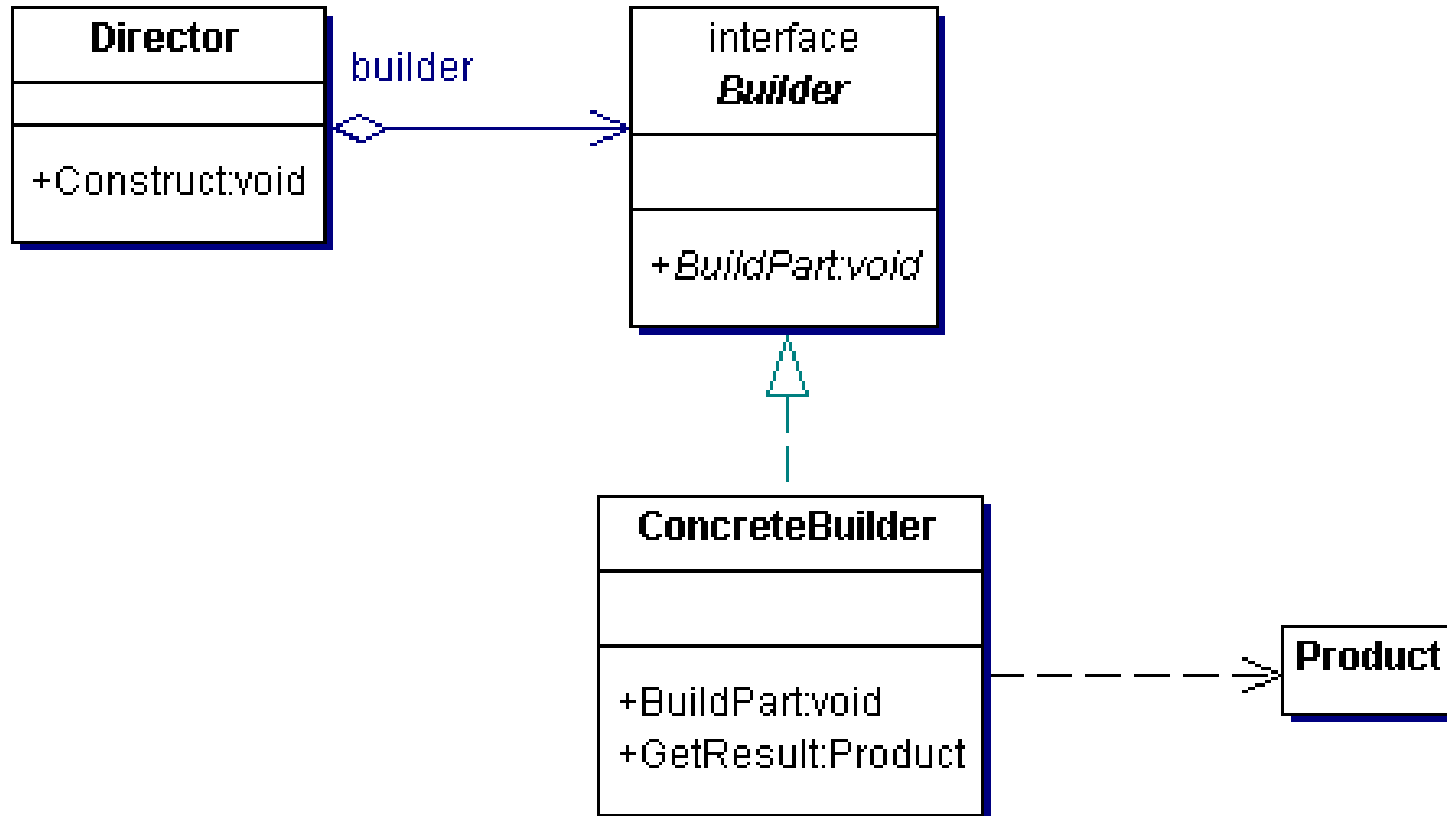
- V.Gruhn, A. Thiel: Komponentenmodelle, Addison-Wesley 2000
- Technical Report: An Inductive Method For Discovering Design Patterns From Object-Oriented Software Systems, University of Maryland, Computer Science Department
<http://hem.passagen.se/gumby/cs/patterns.html>

ComponentConfigurator



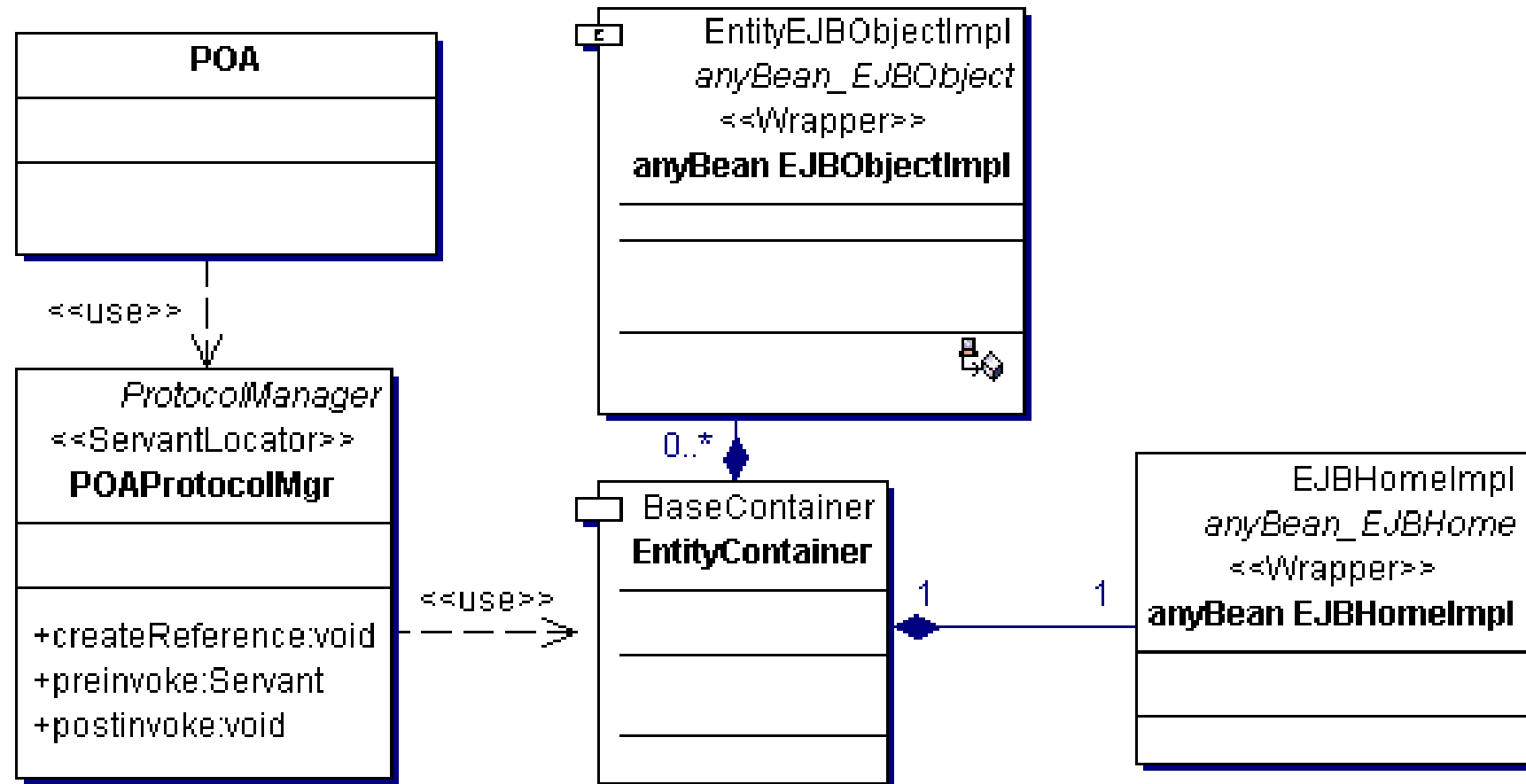
- Configurator konfiguriert die Komponenten
- Repository verwaltet die Komponenten zu Laufzeit
- Component definiert die Schnittstelle zu Verwaltung und Kontrolle
- ConcreteComponent kann dynamisch an Anwendung angebunden werden

Builder-Muster



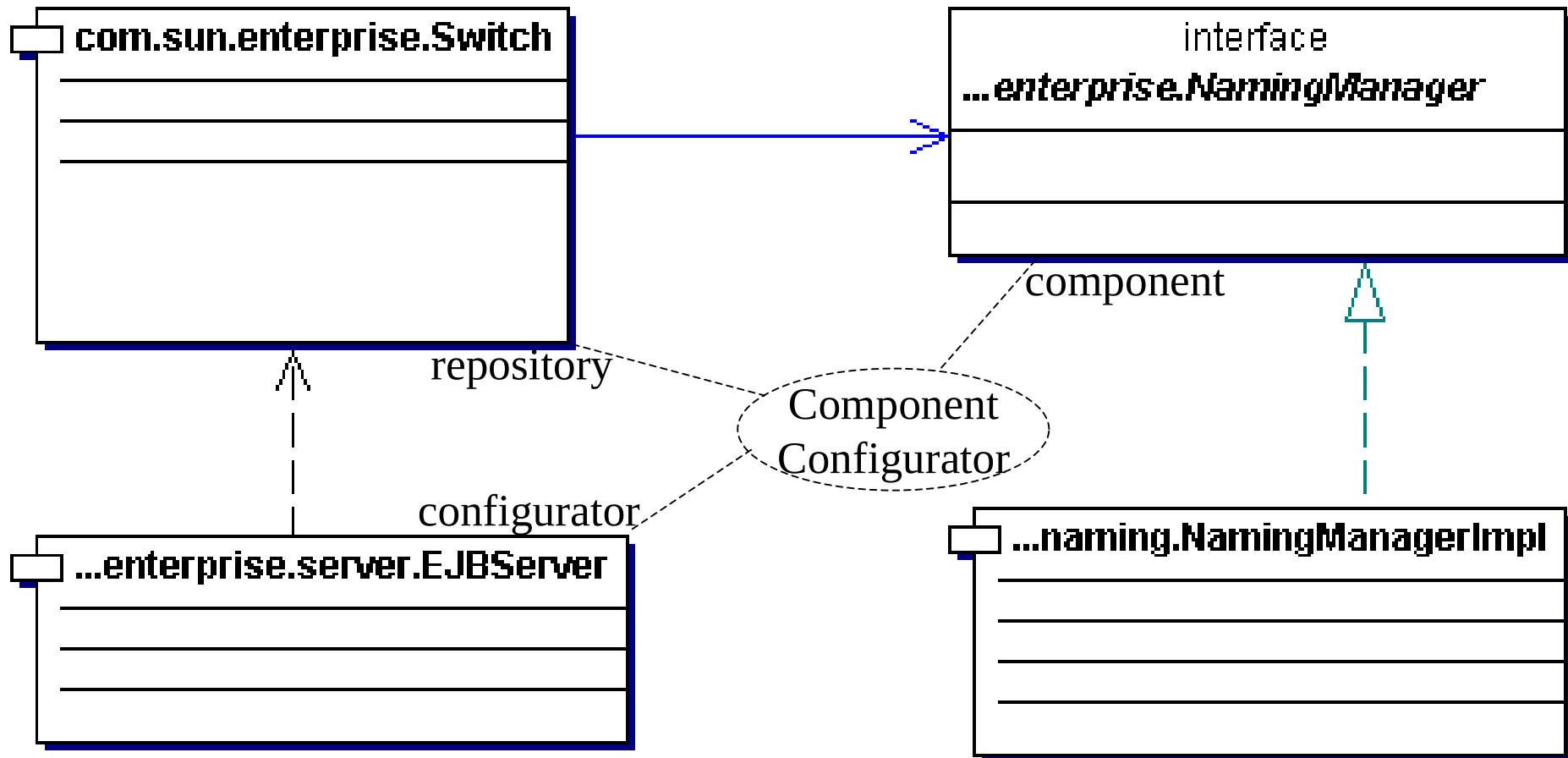
- Builder definiert eine Methode zum Konstruieren des Produkts
- Director konstruiert das Produkt

EJB-Container von SUN: Interzeption (2)



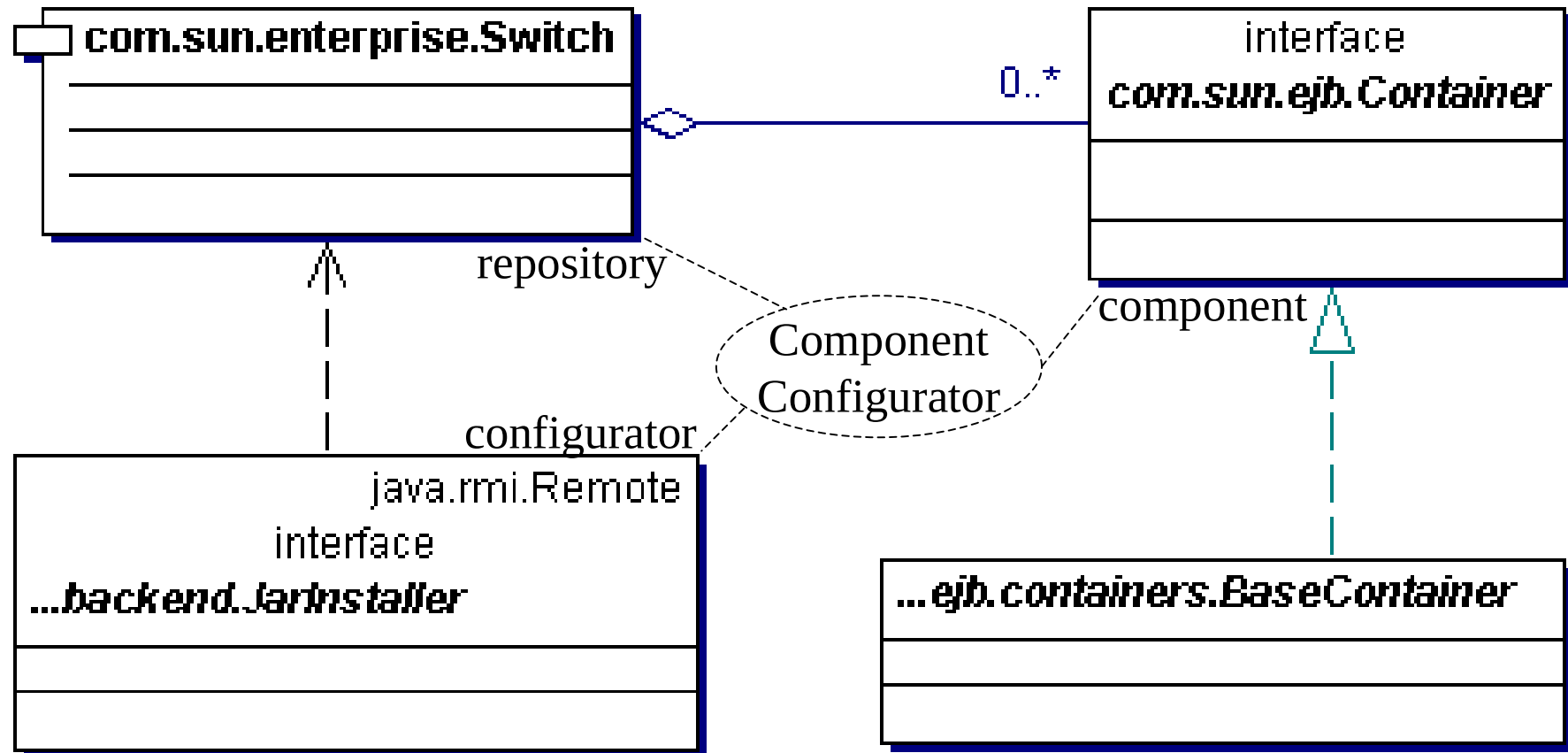
- ServantLocator-Design
- Referenz zum Wrapper-Objekt

EJB-Container von SUN: Laufzeitumgebung



- ComponentConfigurator für Komponenten verschiedener Type
 - Komponenten haben eine Referenz zu Repository - quasi Kontext

EJB-Container von SUN: Installation



- ContainerConfigurator-Variante zu ComponentConfigurator-Muster
 - Configurator verwaltet die Container nicht
 - Descriptoren als Konfigurationsanweisungen zu Laufzeit