



# Entity-Control-Boundary

als Komponentenmodell für  
Geschäftskomponenten

Referent: Gurpreet Dhami



# Agenda

1. Grundlagen
  1. Enge Bindung & Lose Kopplung
  2. Patterns
  3. (Drei-)Schichtenarchitektur
  4. (Drei-)Schichtenarchitektur in Java EE
2. Rollen der einzelnen Komponenten
  1. Boundary
  2. Control
  3. Entity
3. Implementierung mit Java-EE Mitteln
  1. Boundary
  2. Control
  3. Entity
  4. Remoteaufruf aus einem Java Client



# 1. Grundlagen

## Enge Bindung & Lose Kopplung

- Kohäsion = Bindung
- Streben nach Single Responsibility Prinzip
- Wenig bis keine Wiederholung von Code (DRY)
- Bei Änderung einer Codestelle nur diese ändern.
- Wiederverwendbarkeit der Komponenten
  
- Keine feste Verdrahtung der Klassen untereinander
  
- Austauschbarkeit



# 1. Grundlagen Patterns

- wiederkehrende Muster
- Zuordnung von Verantwortlichkeiten
- Kommunikation unter Entwicklern
  
- Arten von Mustern:
  - Erzeugungsmuster
  - Strukturmuster
  - Verhaltensmuster



# 1. Grundlagen Patterns Beispiele

## ► Erzeugungsmuster

- Singleton
- Factory

## ► Strukturmuster

- Fassade
- Kompositum
- Dekorierer

## ► Verhaltensmuster

- Beobachter
- Strategie
- Iterator

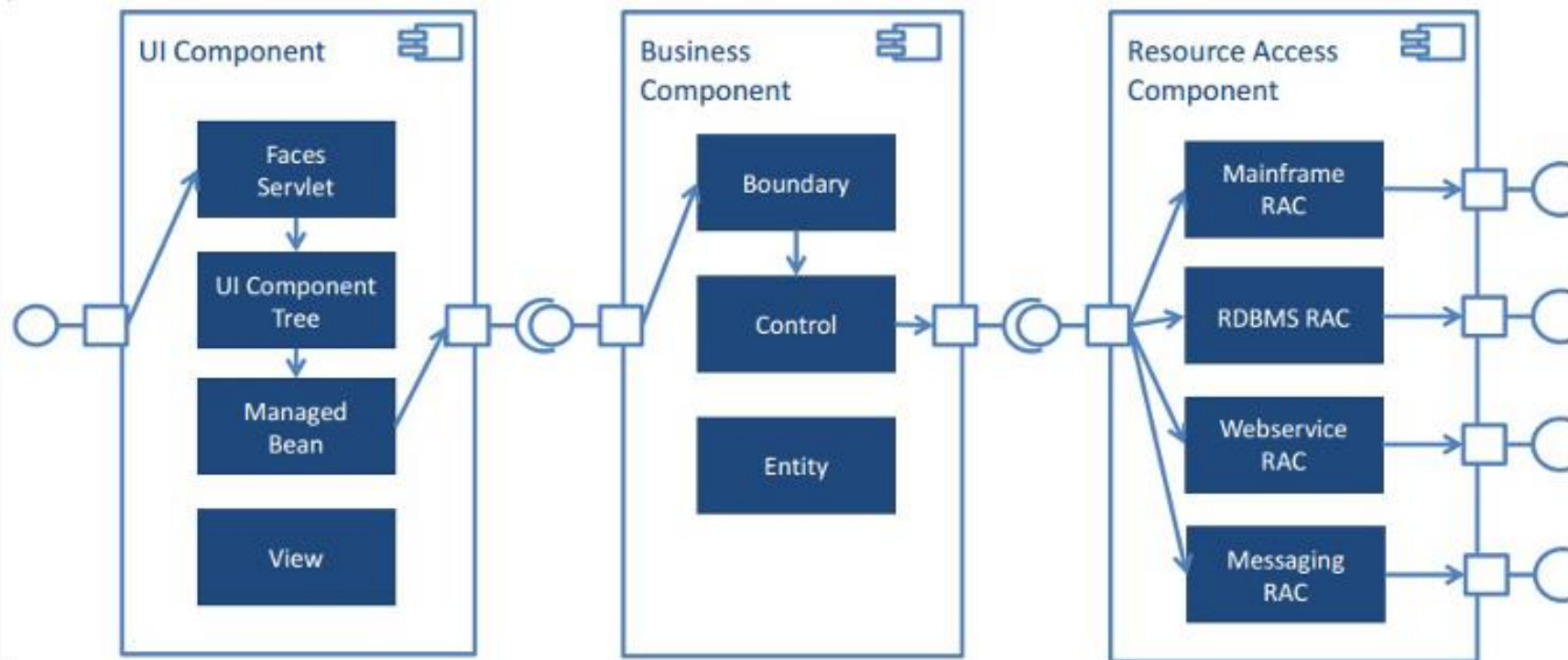


# 1. Grundlagen (Drei-)Schichtenarchitektur



Quelle: Vgl. [http://tschutschu.de/resources/SS2014\\_01\\_Architektur.pdf](http://tschutschu.de/resources/SS2014_01_Architektur.pdf), S. 5

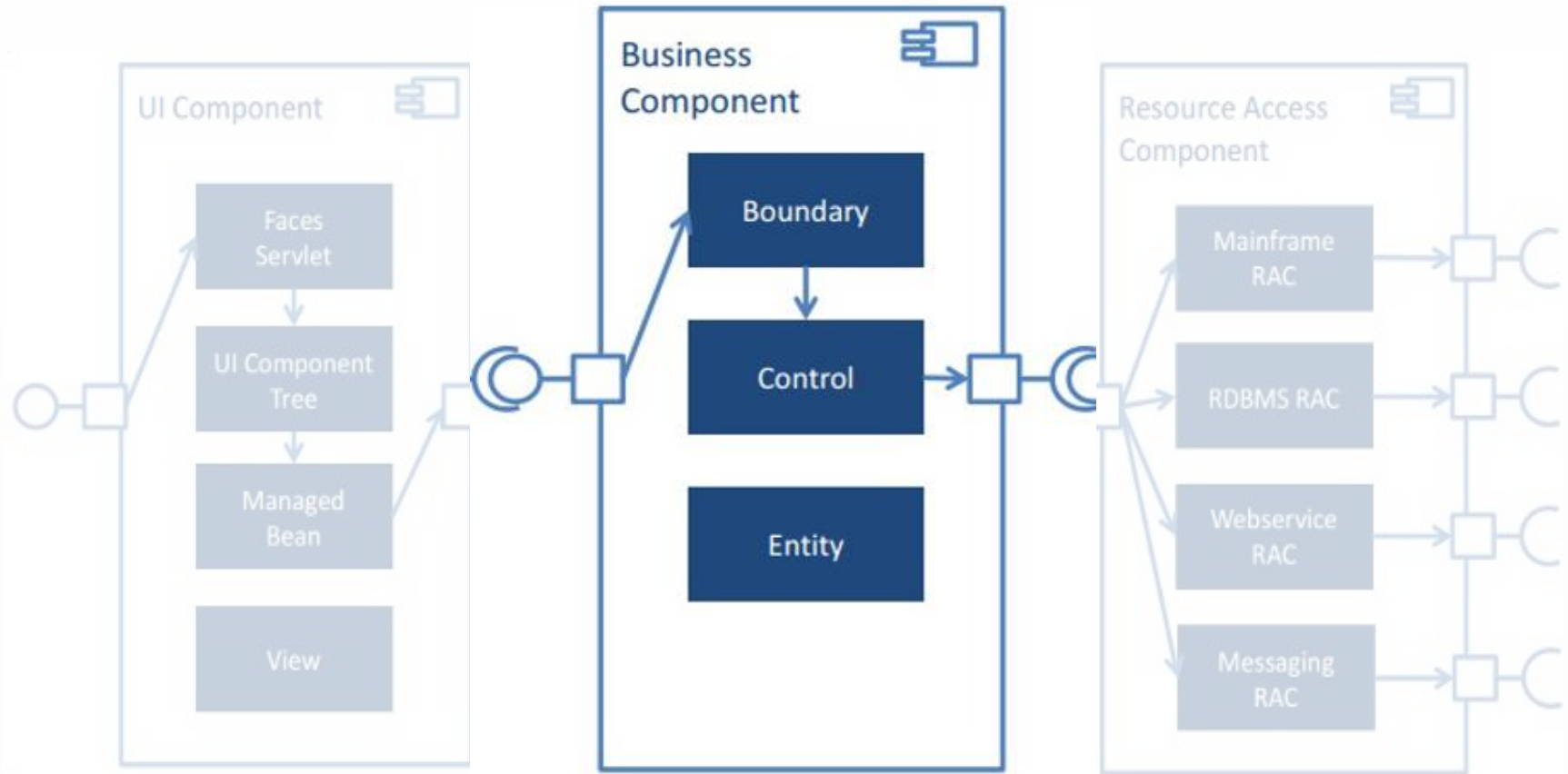
# 1. Grundlagen (Drei-)Schichtenarchitektur in Java EE



Quelle: Vgl. [http://tschutschu.de/resources/SS2014\\_01\\_Architektur.pdf](http://tschutschu.de/resources/SS2014_01_Architektur.pdf), S. 12

# 1. Grundlagen

## ECB in der Geschäftslogikschicht

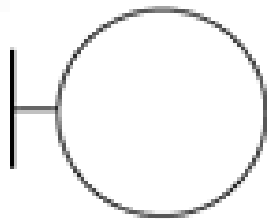


Quelle: Vgl. [http://tschutschu.de/resources/SS2014\\_01\\_Architektur.pdf](http://tschutschu.de/resources/SS2014_01_Architektur.pdf), S. 12





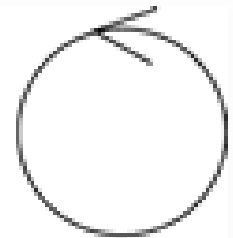
## 2. Rollen der einzelnen Komponenten UML Stereotypen



**Boundary  
object**



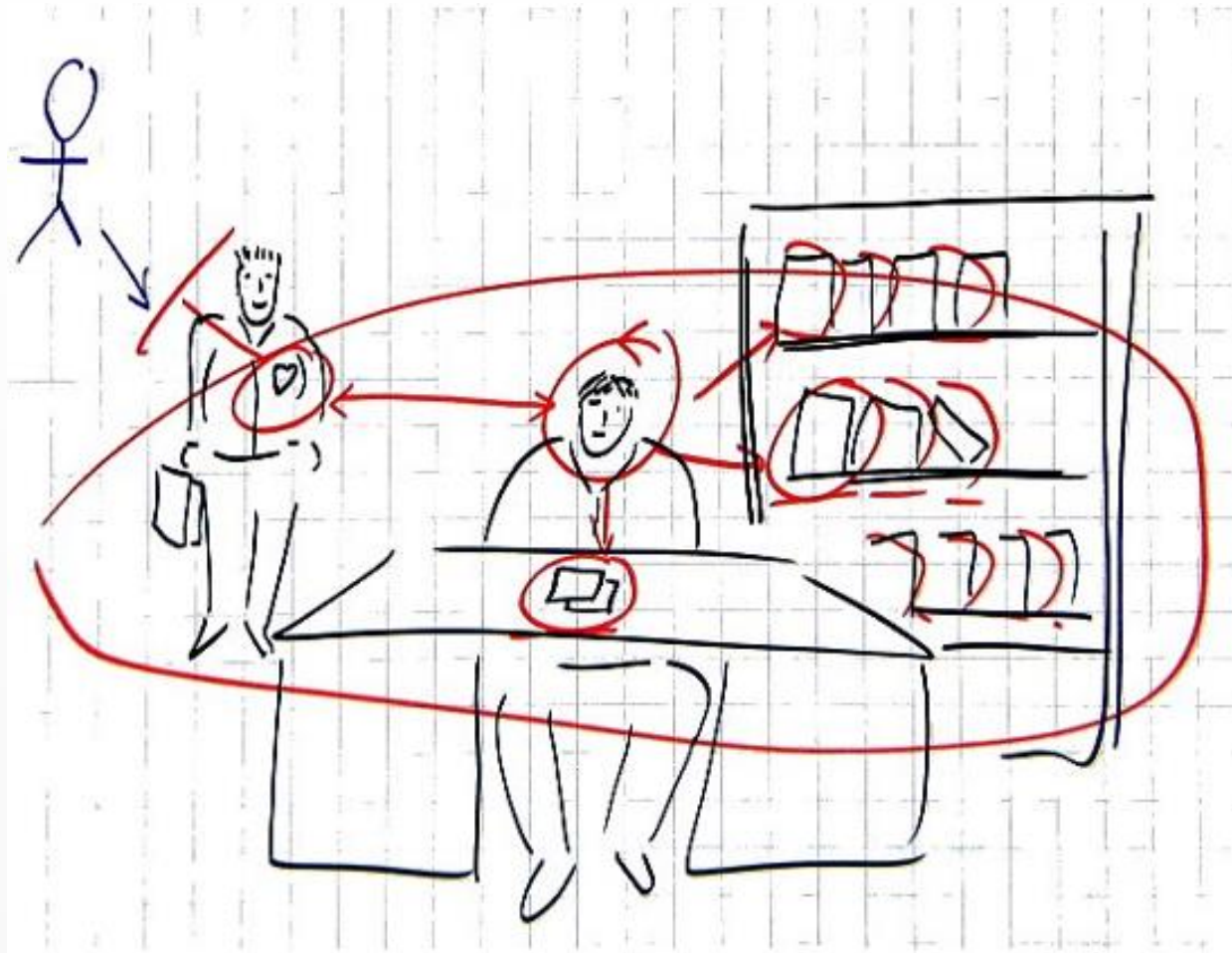
**Entity  
object**



**Control  
object**

Quelle: <http://i.stack.imgur.com/rQPvd.gif>

## 2. Rollen der einzelnen Komponenten Analogbeispiel



Quelle: <http://agilewiki.ipponsoft.de/doku.php?id=entity-control-boundary>

## 2. Rollen der einzelnen Komponenten Marketing Beispiel

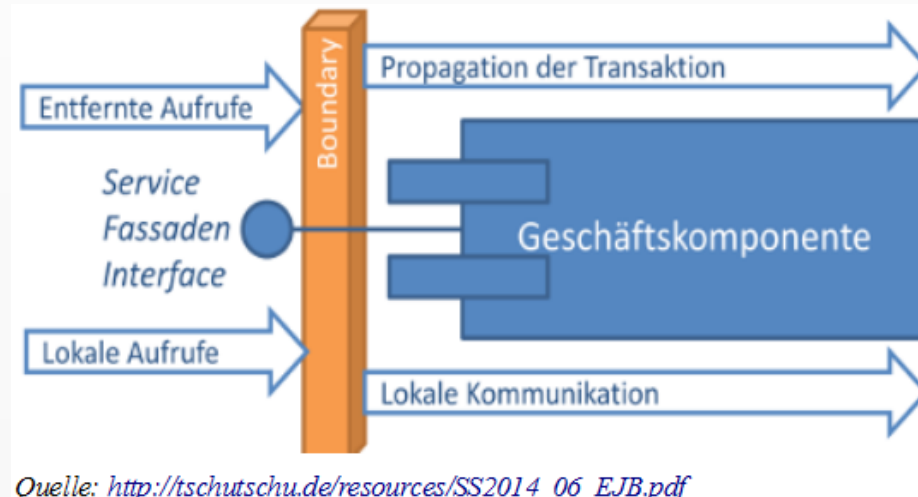


Quelle: [http://epf.eclipse.org/wikis/openup/core.tech.common.extend\\_supp/guidances/guidelines/entity\\_control\\_boundary\\_pattern\\_C4047897.html](http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/guidelines/entity_control_boundary_pattern_C4047897.html)

## 2. Rollen der einzelnen Komponenten

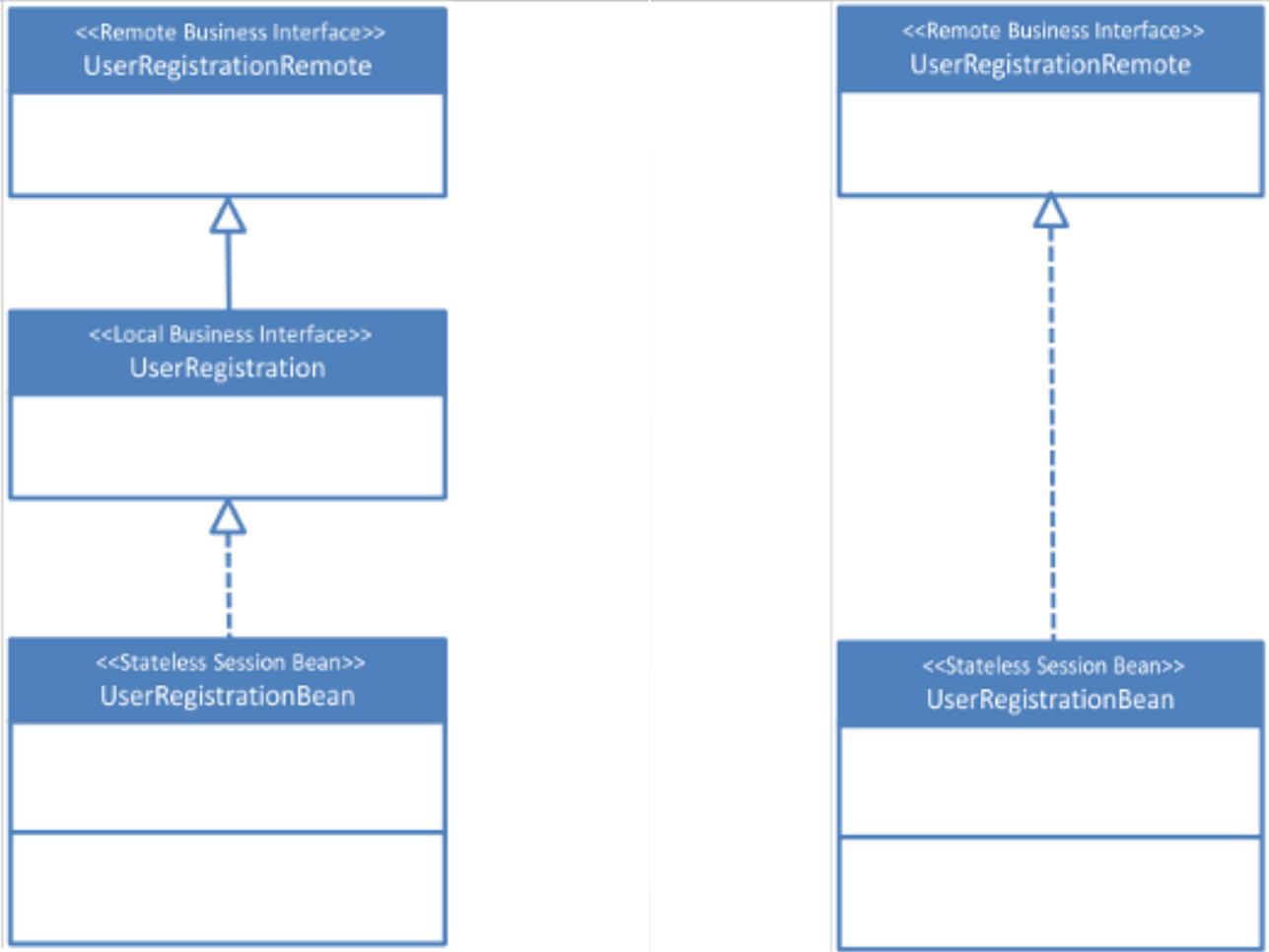
### 2.1 Boundary

- Entwurfsmuster Fassade
- grob-granularer Zugriff auf die BLL von der UI
- Transaktionssteuerung
- Session Beans
- Schnittstellen/ Adapter für alle Zugriffsarten
- Zugriff auf Conrols, Entities und Integration Layer





## 2. Rollen der einzelnen Komponenten Dual-View Interfaces



Quelle: [http://tshutschu.de/resources/SS2014\\_06\\_EJB.pdf](http://tshutschu.de/resources/SS2014_06_EJB.pdf) Quelle: [http://tshutschu.de/resources/SS2014\\_06\\_EJB.pdf](http://tshutschu.de/resources/SS2014_06_EJB.pdf)



## 2. Rollen der einzelnen Komponenten

### 2.2 Control

- Kernfunktionalität der Geschäftslogikschicht
- fein-granulare Geschäftslogik
- wiederverwendbar
- POJOs oder Stateless Session Beans
- Zugriff auf:
  - Integrationschicht
  - andere Conrols
  - Entities
  - aber nicht auf die Boundary

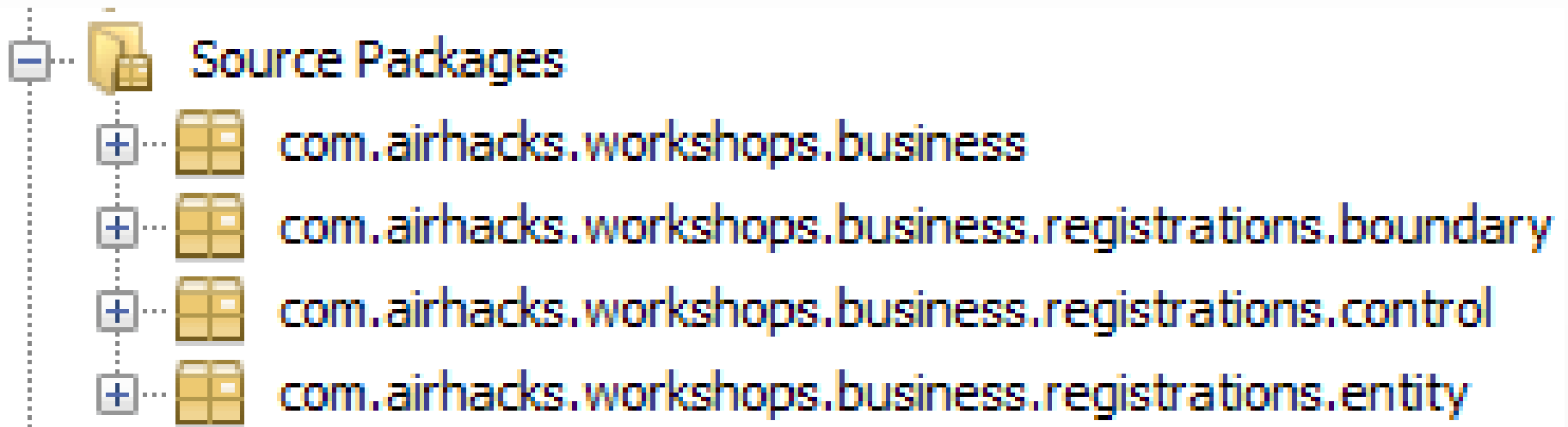


## 2. Rollen der einzelnen Komponenten

### 2.3 Entity

- bildet das Domänenmodell ab
- Simple Datenträger (SOA) ↔ „fette“ Objekte (DDD)
- Zugriffe nur auf andere Entities

## 3 Implementierung des ECB mit Java EE Packaging Struktur





## 3 Implementierung des ECB mit Java EE Packaging Struktur





## 3.1 Implementierung der Boundary

```
package RemoteClient;

import java.util.Date;

public interface IRegistrations{

    Registration find(int registrationId);
    Registration register(String Name, Date termin);
}
```

## 3.1 Implementierung der Boundary



```
package com.airhacks.workshops.business.registrations.boundary;

import javax.ejb.*;
import javax.inject.Inject;
import javax.persistence.*;
import com.airhacks.workshops.business.registrations.control.*;

@Stateless
@Remote(IRegistrations.class)
@Transactional(TransactionalType.REQUIRES_NEW)
public class Registrations implements IRegistrations
{
    @PersistenceContext
    EntityManager em;

    @Inject
    VatCalculator priceCalculator;

    @Override
    public Registrations register(Registrations request) {
```

## 3.2 Implementierung des Controls



```
package com.airhacks.workshops.business.registrations.control;

import java.math.BigDecimal;
import static java.math.BigDecimal.valueOf;
import javax.ejb.TransactionAttribute;
import javax.ejb.TransactionAttributeType;

@TransactionAttribute(TransactionAttributeType.MANDATORY)
public class VatCalculator {

    public int calculateTotal(boolean vatIdAvailable, int price) {
        BigDecimal net = valueOf(price);
        if (vatIdAvailable) {
            return net.intValue();
        } else {
            return net.add(net.multiply(valueOf(0.19))).intValue();
        }
    }
}
```

## 3.3 Implementierung der Entity



```
package com.airhacks.workshops.business.registrations.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

@Entity
public class Registration {

    @Id
    @GeneratedValue
    private long id;

    private int numberOfDays;
    private int numberOfAttendees;
    private boolean vatIdAvailable;

    private final static int DAILY_PRICE = 300;

    private int totalPrice;

    public Registration(boolean vatIdAvailable, int numberOfDays, int numberOfAttendees) {
        this.numberOfDays = numberOfDays;
        this.numberOfAttendees = numberOfAttendees;
        this.vatIdAvailable = vatIdAvailable;
    }
}
```

# Remoteaufruf der Geschäftslogikschicht



```
package RemoteClient;

import javax.naming.*;

public class Presentation {

    public static void main(String[] args) {

        try {
            Context jndiCtx = new InitialContext();

            String lookupname = "java:/global/ee-demos-ejb/RegistrationService!"
                + "com.airhacks.workshops.business.registrations.RegistrationService";
            IRegistrations regService = (IRegistrations) jndiCtx.lookup(lookupname);
            IRegistrations reg = regService.find(1);

        } catch (Exception ex) {
        }

    }
}
```



Vielen Dank für Ihre Aufmerksamkeit!

Fragen ?

Referent: Gurpreet Dhani