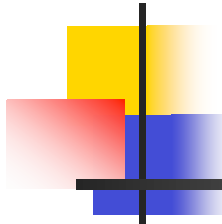


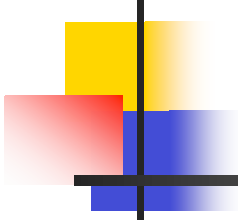
Besprechung Uebung 4

Software Engineering



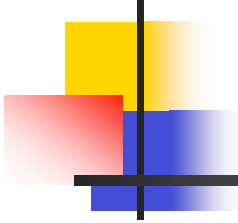
Formelles

- Abgaben:
 - Namen auf Abgaben, keine Umlaute in Dateinamen!
 - **1 Punkt Abzug**
- Verschiedenes:
 - Gruppen für die Übungsbearbeitung nicht Einzelpersonen.
 - Eine Abgabe pro Gruppe genügt.
 - Gut gelöst!
- Rechtschreibung, Grammatik
 - Mühsam zu lesen
 - Keine Abzüge
- Strukturierte Argumentation



Aufgabe 2.1

- Architekturüberblick
- Nicht schlecht gelöst.
- Zweck:
 - Grobüberblick über die Organisation der Software
 - In dieser Aufgabe: Einführungsaufgabe um den Code zu verstehen



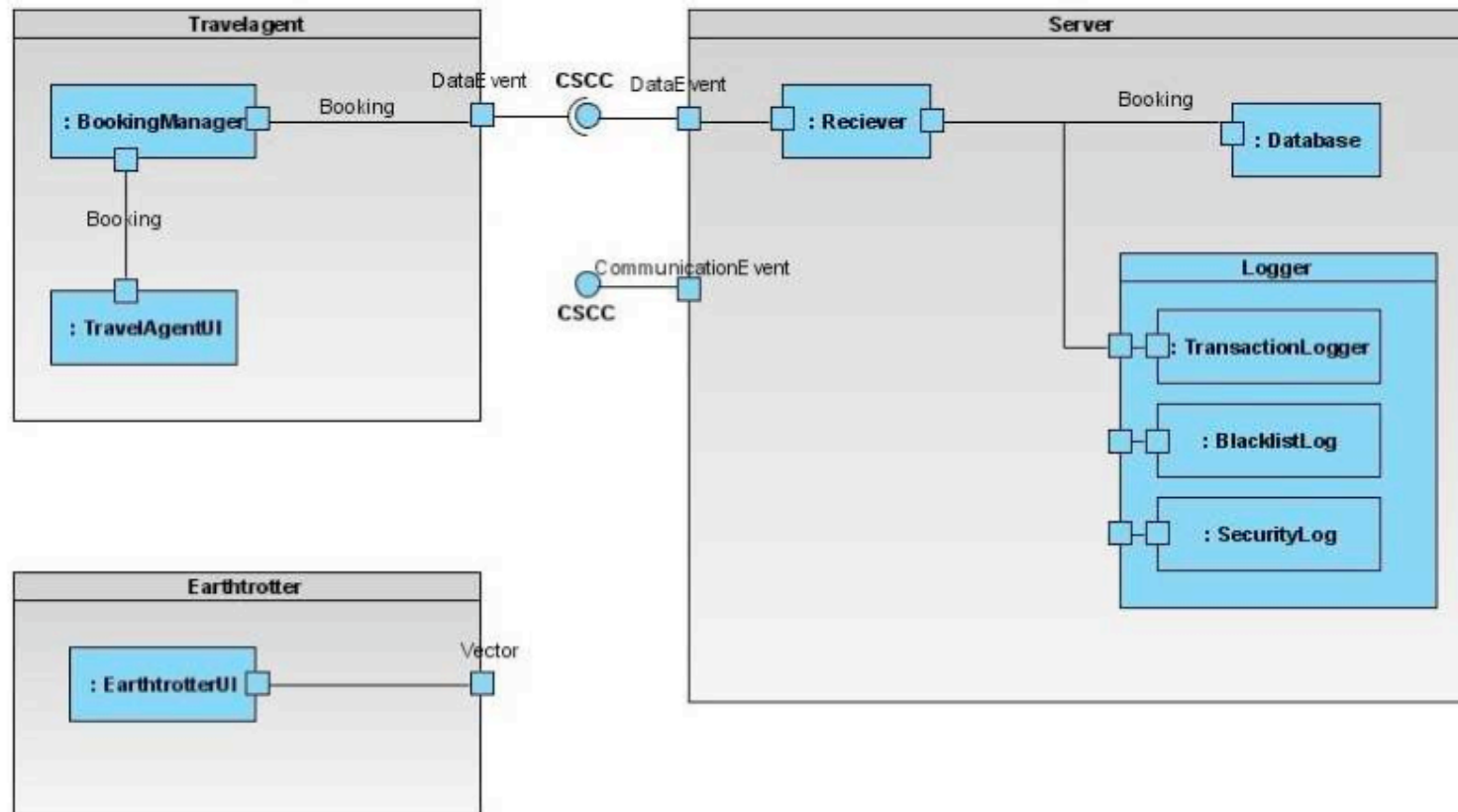
Aufgabe 2.1 (2)

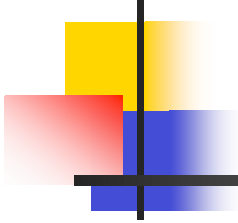
- Korrekturkriterien:
 - Wahl einer sinnvollen Notation
 - Sinnvoll modellierte Architektur
 - Beschreibung der Komponenten

- Probleme:
 - Zu detailliert (Abstraktion):
Reverse Engineering Klassendiagramm $\neq$ guter Architekturüberblick
 - Probleme mit der Syntax:
Ungeeignete Notation, Inkonsistenzen, falsch verwendete Beziehungen

- Probleme von Klassendiagrammen, Composite Structure Diagrams?

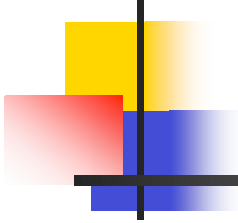
Aufgabe 2.1 (3)





Aufgabe 2.2

- Analyse der verwendeten Architekturstile
- Generell gut gelöst!
- Architektur:
 - Verwendete Module (Komponenten)
 - Kooperation zwischen den Modulen
- Architekturstil:
 - Vokabular
 - Regeln und Einschränkungen
 - Semantische Interpretation
- Kombination verschiedener Stile

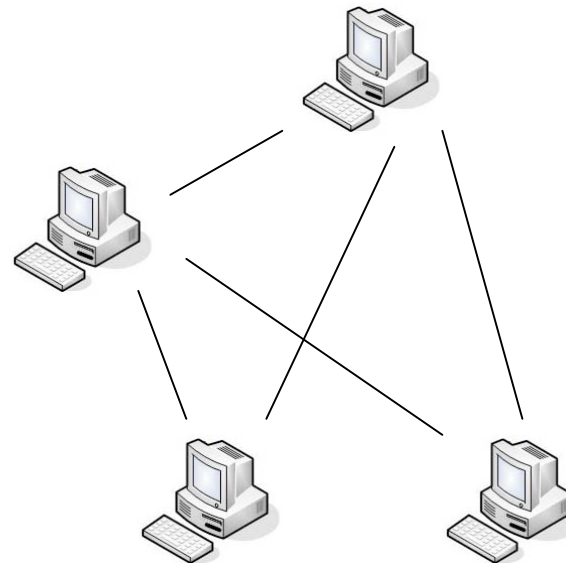
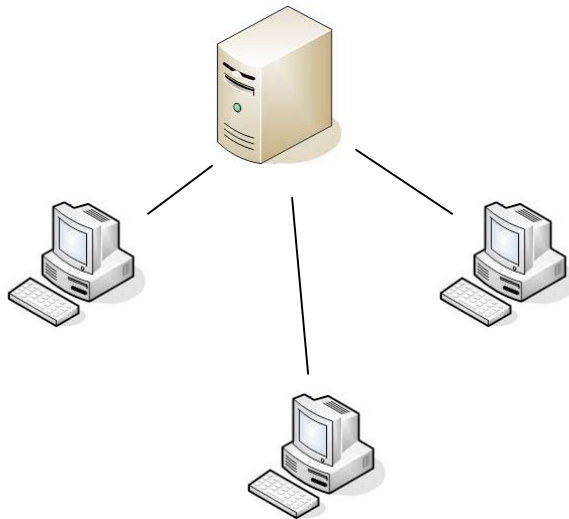


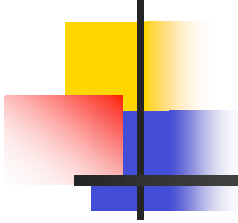
Aufgabe 2.2 (2)

- Pipe-and-Filter
- Layered System
- Blackboard
- **Event-based** (Kommunikation mit csc)
- Distributed Peer-to-Peer
- **Client / Server** (physische Verteilung)

Aufgabe 2.2 (3)

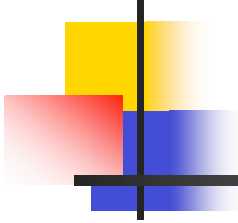
- Client/Server vs. Peer-to-Peer
 - Peers sind gleichberechtigt
 - Keine zentrale Instanz (Mischformen: z.B. Napster)





Aufgabe 2.3

- Entwurfsmuster in der Software
- Gut gelöst!
- Zweck:
 - Bewährte, vorgefertigte Lösungsschablonen für wiederkehrende Entwurfsprobleme
 - Wiederverwendung von Entwurfswissen
 - Begriffliche Basis und Terminologie



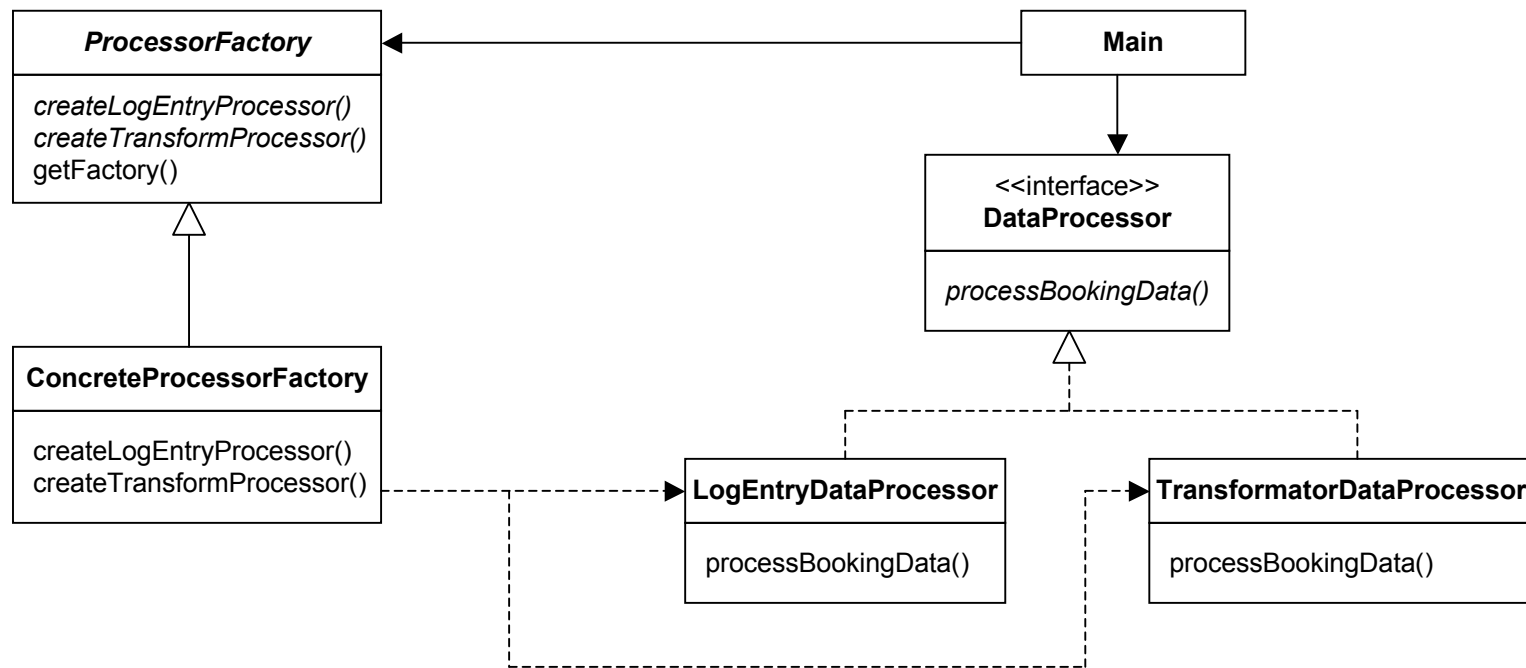
Aufgabe 2.3 (2)

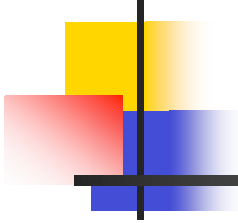
- **Abstract Factory** (pinkair.processing)
- Adapter
- **Observer** (Benachrichtigung durch csc)
- State
- Factory Method

- Singleton

Aufgabe 2.3 (3)

Abstract Factory + Factory Method



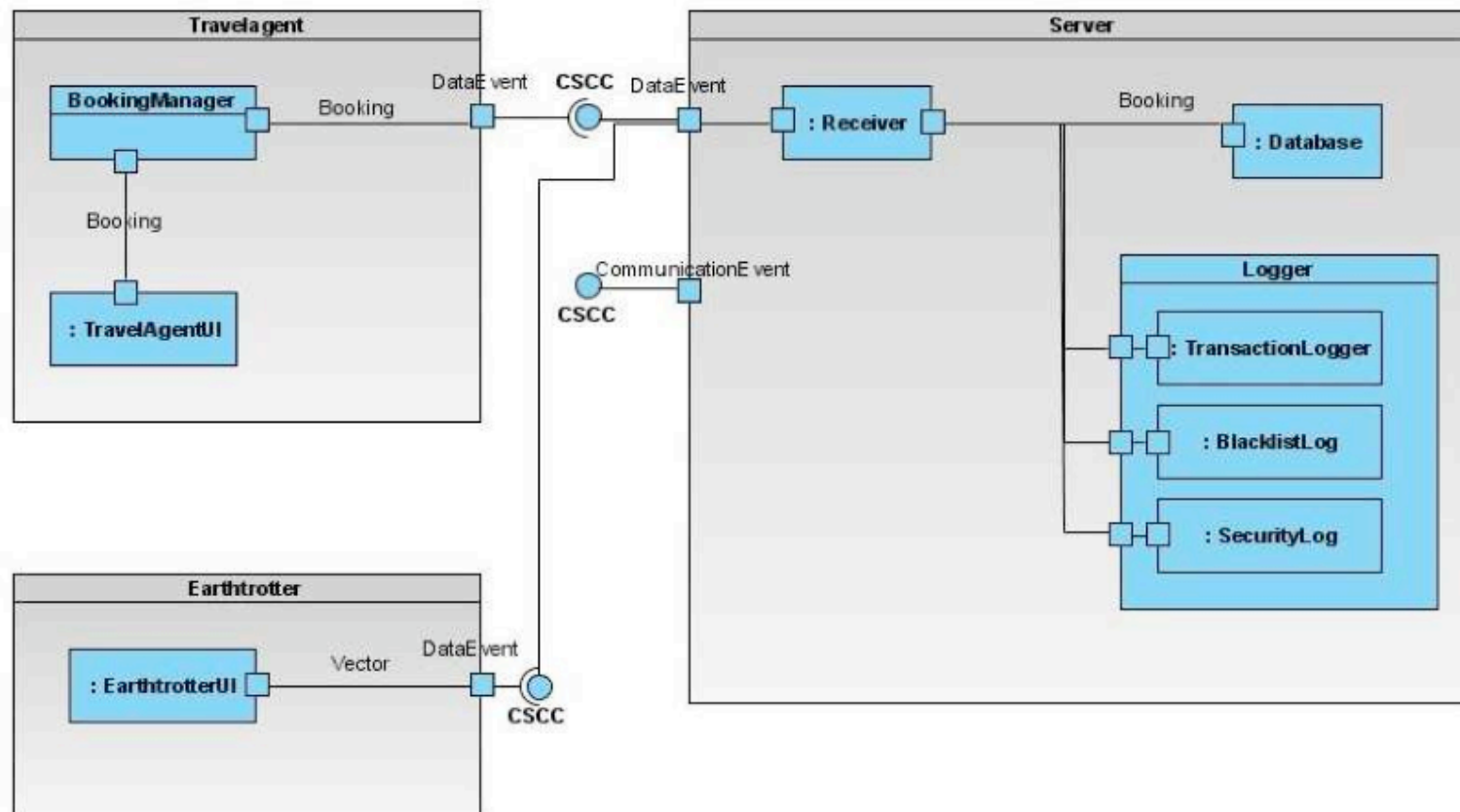


Aufgabe 2.4

- Architekturanpassungen
 - Architekturstil um verschiedene Daten zusammenzuführen
 - Architektur für Umsetzung der zusätzlichen Anforderungen
 - Erweiterung der Architekturbeschreibung
- Unterschiedlich gelöst!

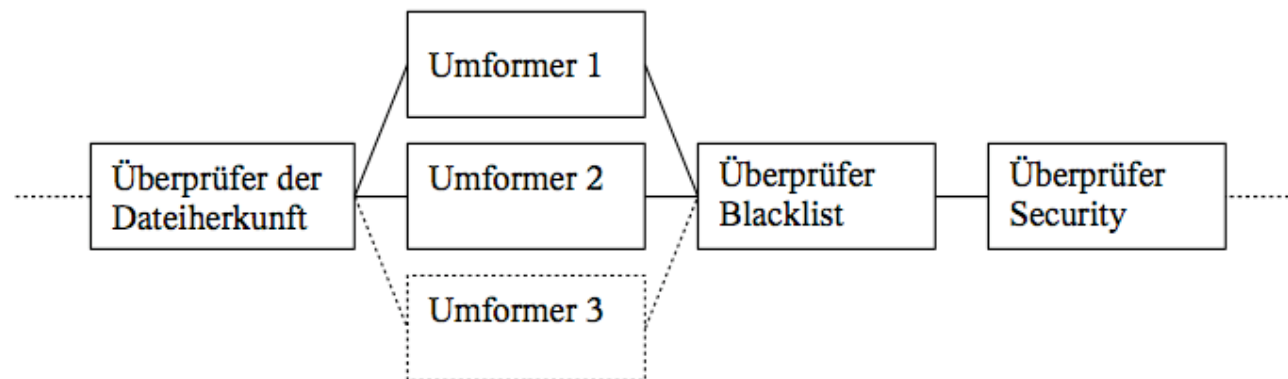
Aufgabe 2.4 (2)

- Client / Server:



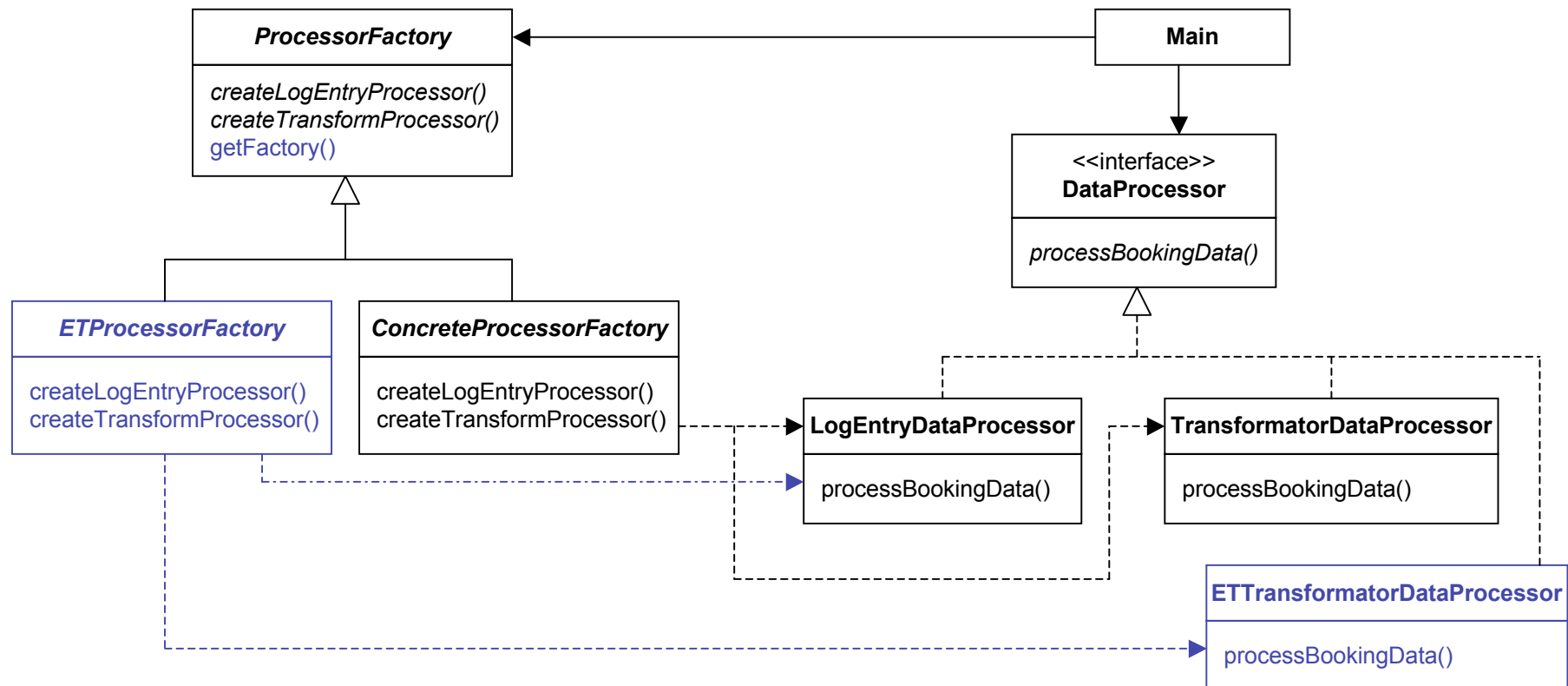
Aufgabe 2.4 (3)

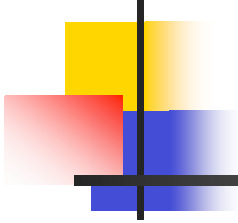
- Zusammenführen, Verarbeiten der Daten: Pipe-and-Filter



Aufgabe 2.4 / 2.5

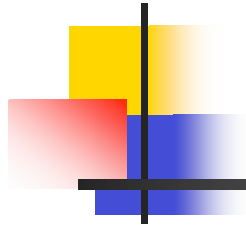
■ Umsetzung





Aufgabe 2.5

- Implementierung
 - Kommunikationsschnittstelle in EarthTrotter
 - Neue Geschäftsregeln, regulatorische Bestimmungen:
 1. Umwandlung der Daten
 2. Transaktions-Log
 3. Blacklist
 4. Besondere Sicherheitsbestimmungen
 5. Buchung in DB ablegen (Stub)
- Laufversuch und Abnahmetest



**Erholungsamen Ferien,
schöne Weihnachten und
ein gutes neues Jahr!**