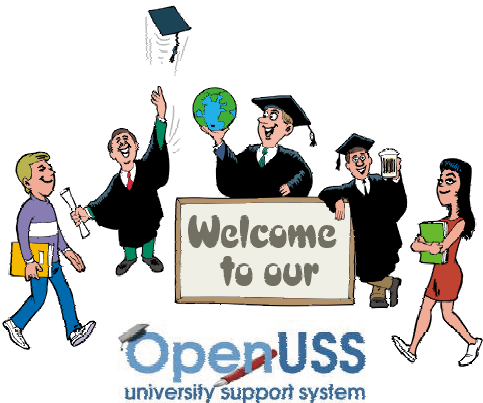


## Integration von CampusSource-Plattformen mittels Model Driven Architecture (MDA)?

Blasius Lofi Dewanto



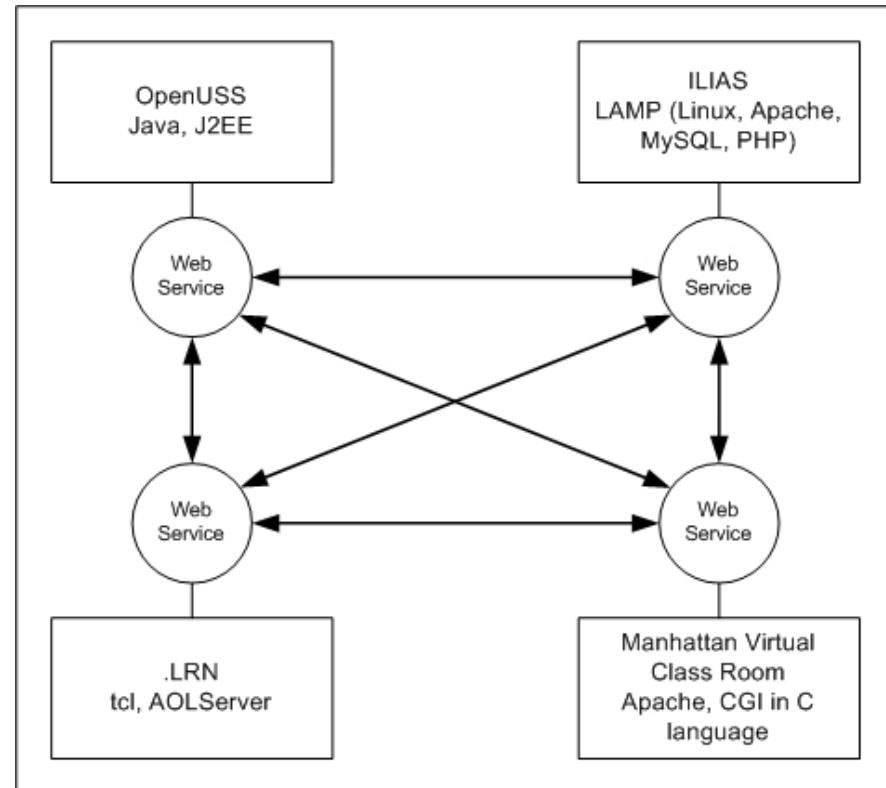
# Situation von CampusSource-Plattformen

- ❑ Ähnliche Funktionen (Verwaltung von Studenten und Dozenten, Diskussionsforen, ...), jedoch
  - Vergleich zwischen Plattformen schwierig, da keine gemeinsame „Sprache“ existiert
- ❑ Verschiedene Technologien (Java, PHP, ...)
  - ⇒ Wiederverwendung von „Komponenten“ diverser Technologien schwierig
- ❑ Geringe technologieübergreifende Zusammenarbeit

# Mögliche Lösung?

## □ Webservice (k)eine Lösung?

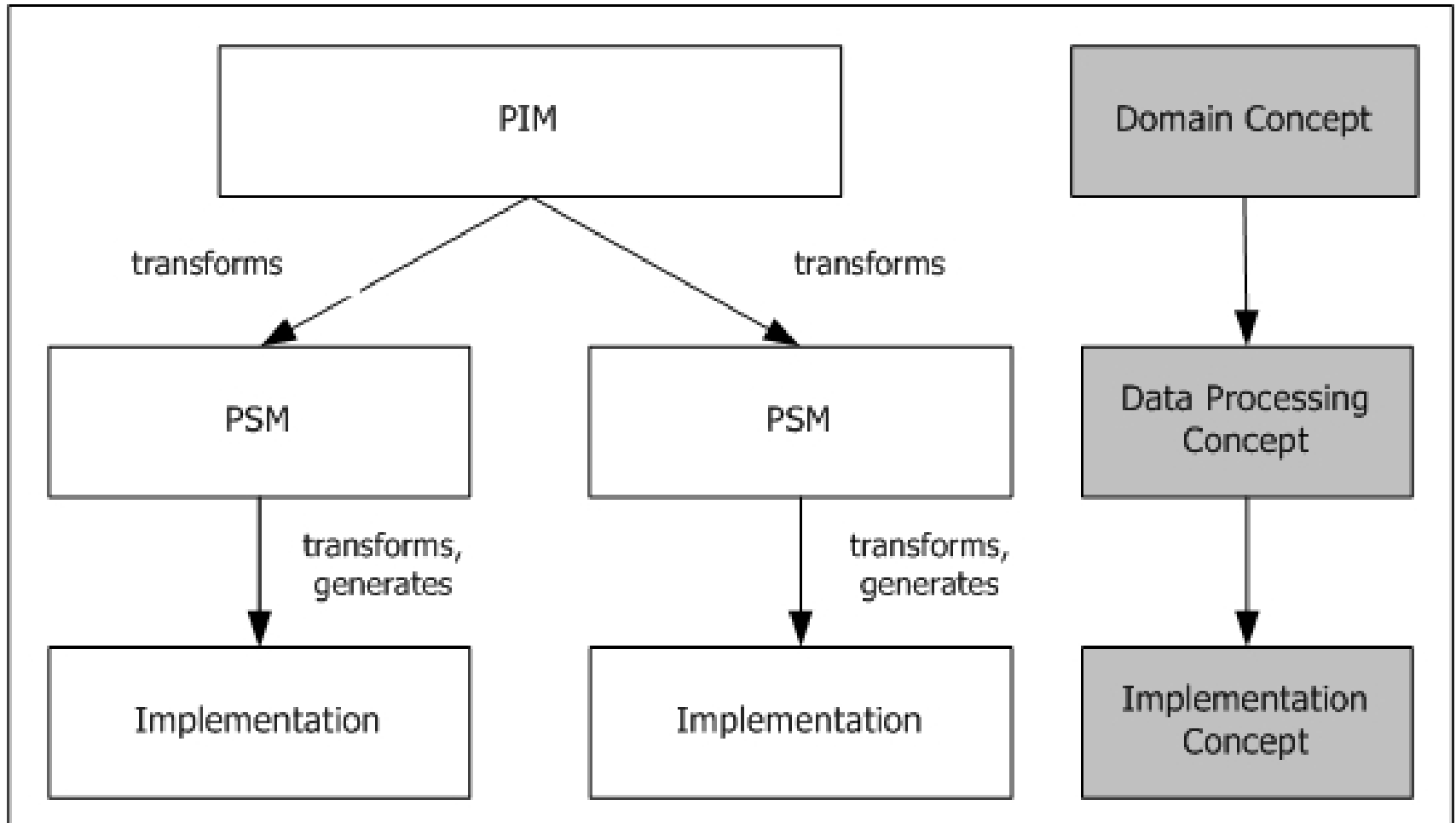
- Webservice ermöglicht die Interoperabilität zwischen Plattformen
- Installation aller Plattformen in einer Institution?
  - Wartung aufwändig
  - Wissen und Erfahrung in allen Technologien vorhanden?



# Model Driven Architecture (MDA)

- Model Driven Architecture mit UML (Unified Modeling Language) als Sprache?
  - Trennung von
    - PIM (Platform Independent Model)
    - PSM (Platform Specific Model)
    - Implementierung
  - Nicht neu ⇒ Bereits bei „Architektur Integrierter Informationssysteme“ (ARIS) bekannt:
    - Fachkonzept
    - DV-Konzept
    - Implementierungskonzept
  - **Neu:**
    - Selbstdefinierte Transformationsregel für die Transformationen zwischen Modellen
    - Standardisiert bei OMG (Object Management Group)
    - Offen durch Open-Source-Werkzeuge

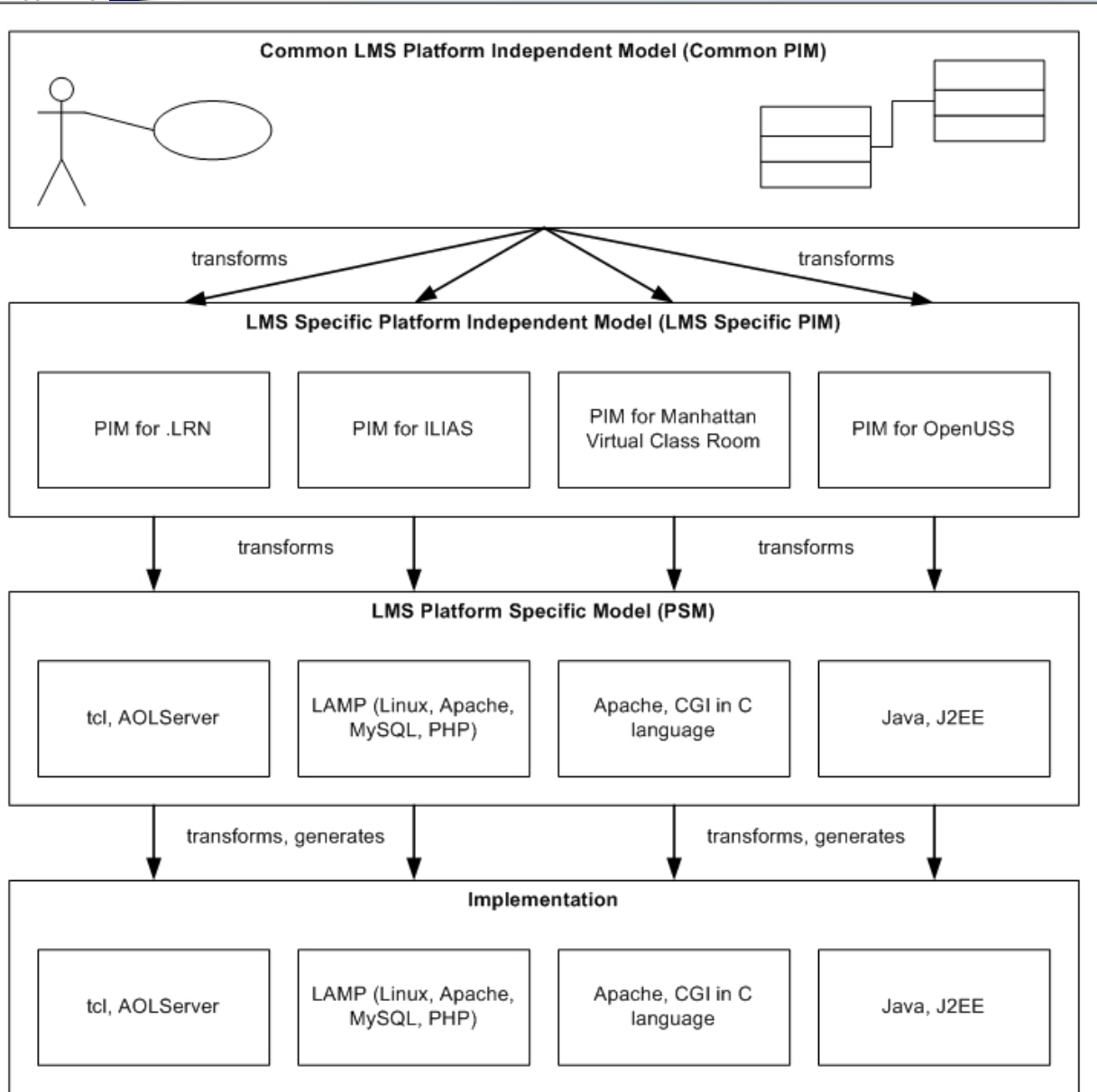
# MDA und ARIS



# Lösung mit MDA (1)

- Definition von allgemeinen PIM für alle Plattformen  
⇒ *Common PIM* (PIM für CampusSource)
- Verfeinerung des CampusSource-PIM zu einzelnen PIM-Plattform, wie beispielsweise
  - *PIM für OpenUSS,*
  - *PIM für ILIAS,*
  - *PIM für ...*
- Verfeinerung der einzelnen PIM-Plattform zum PSM
- Verfeinerung des PSM zur Implementierung

# Lösung mit MDA (2)



Gemeinsame Aktivitäten

Wie bisher...

- Vergleich zwischen Plattformen möglich, dank gemeinsamer Sprache UML
  - Bessere und stets aktuelle Projektdokumentation!
- Wiederverwendung von „Modellen“ möglich
- Technologieübergreifende Zusammenarbeit möglich
- Keine große Veränderung im Entwicklungsprozess notwendig
  - Modell == Quellcode
  - Forward-Engineering: Modellieren, Kompilieren, Implementieren, Kompilieren, Ausführen, Testen