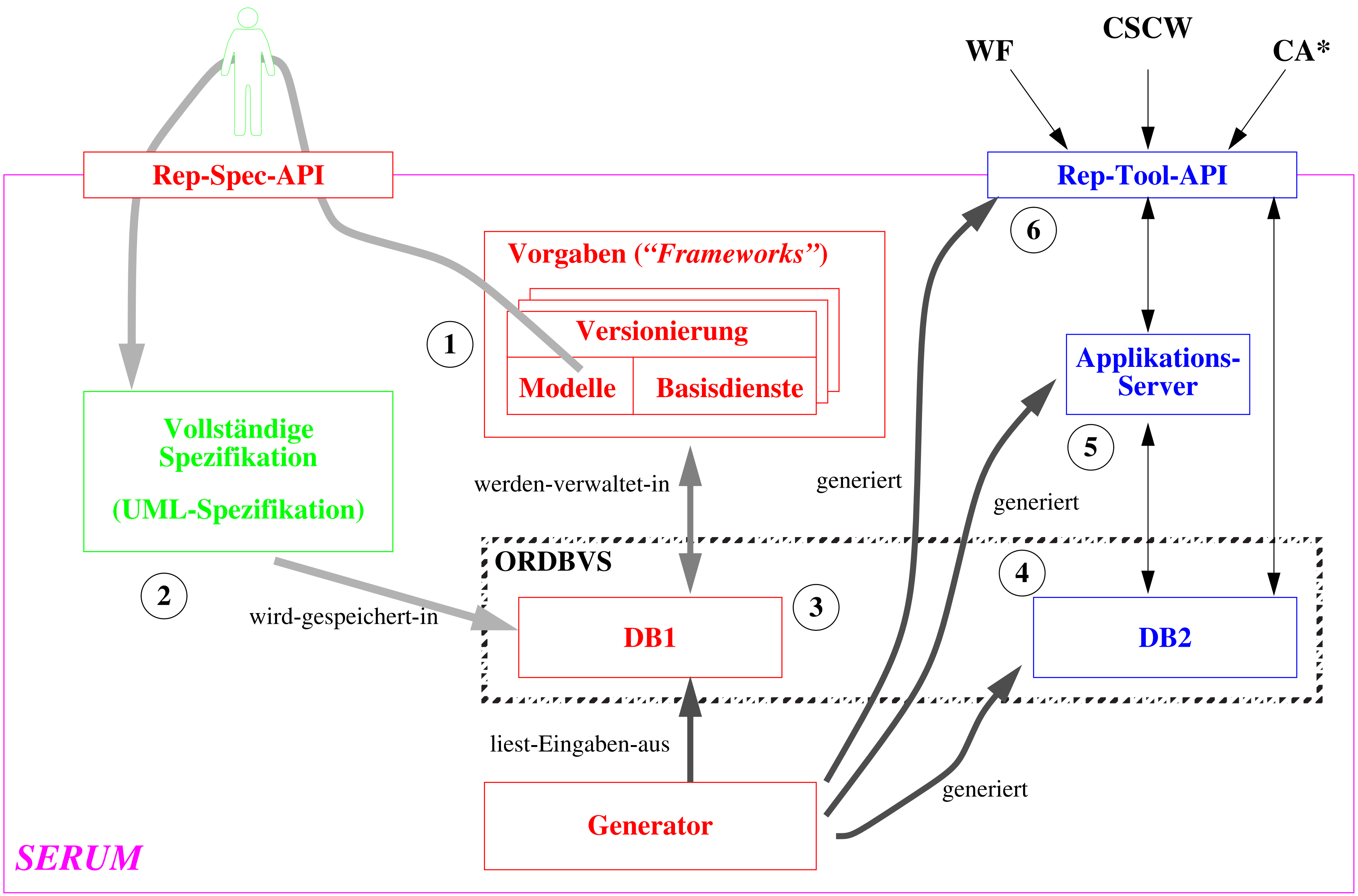
SERUM am Beispiel Versionierung

- Überblick über SERUM
- Grundkonzept des Versionierungs-Frameworks
- Herleiten der Pattern des Versionierungs-Frameworks
- Ausblick

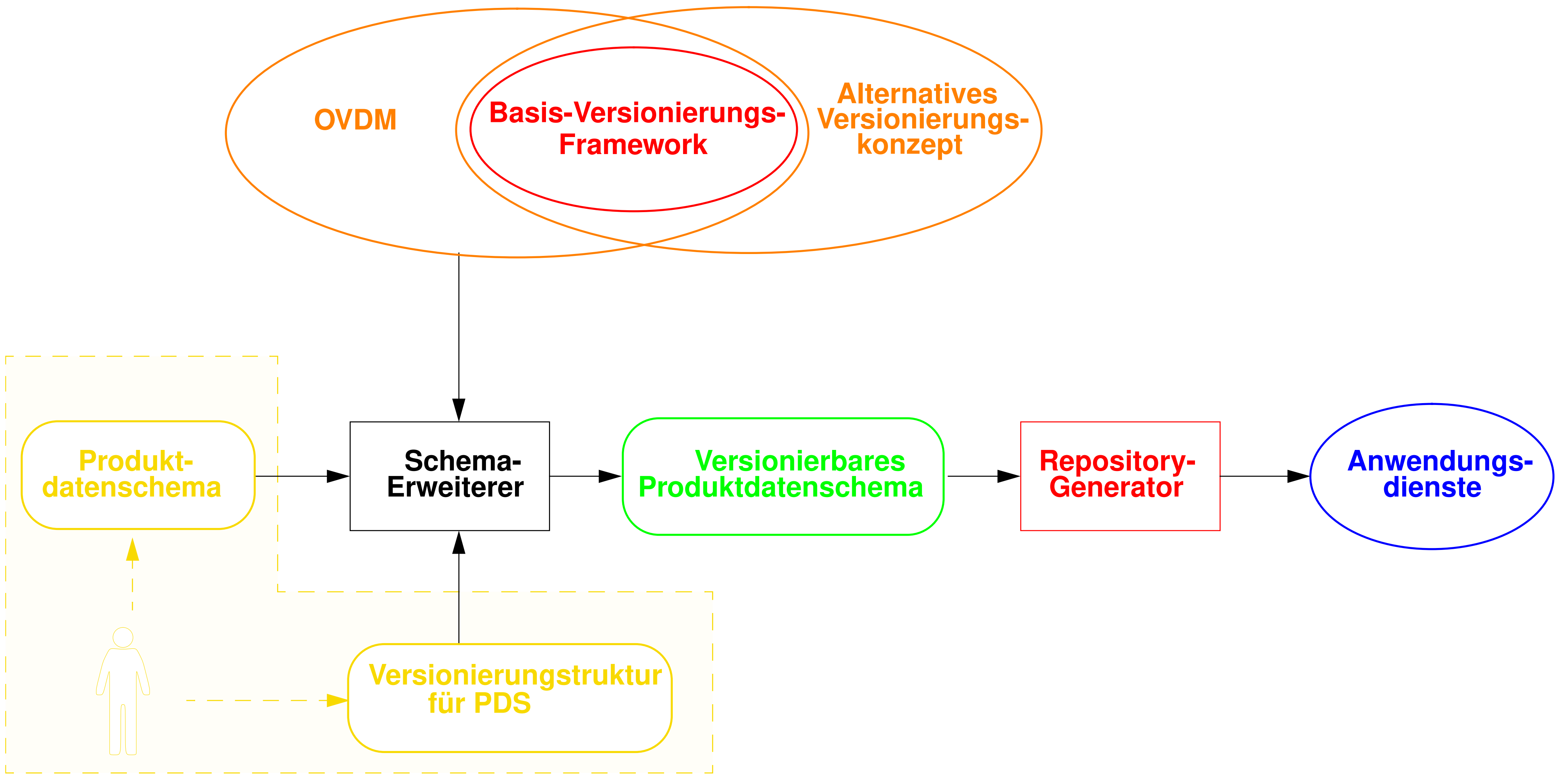
Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Framework-Ansatz für SERUM



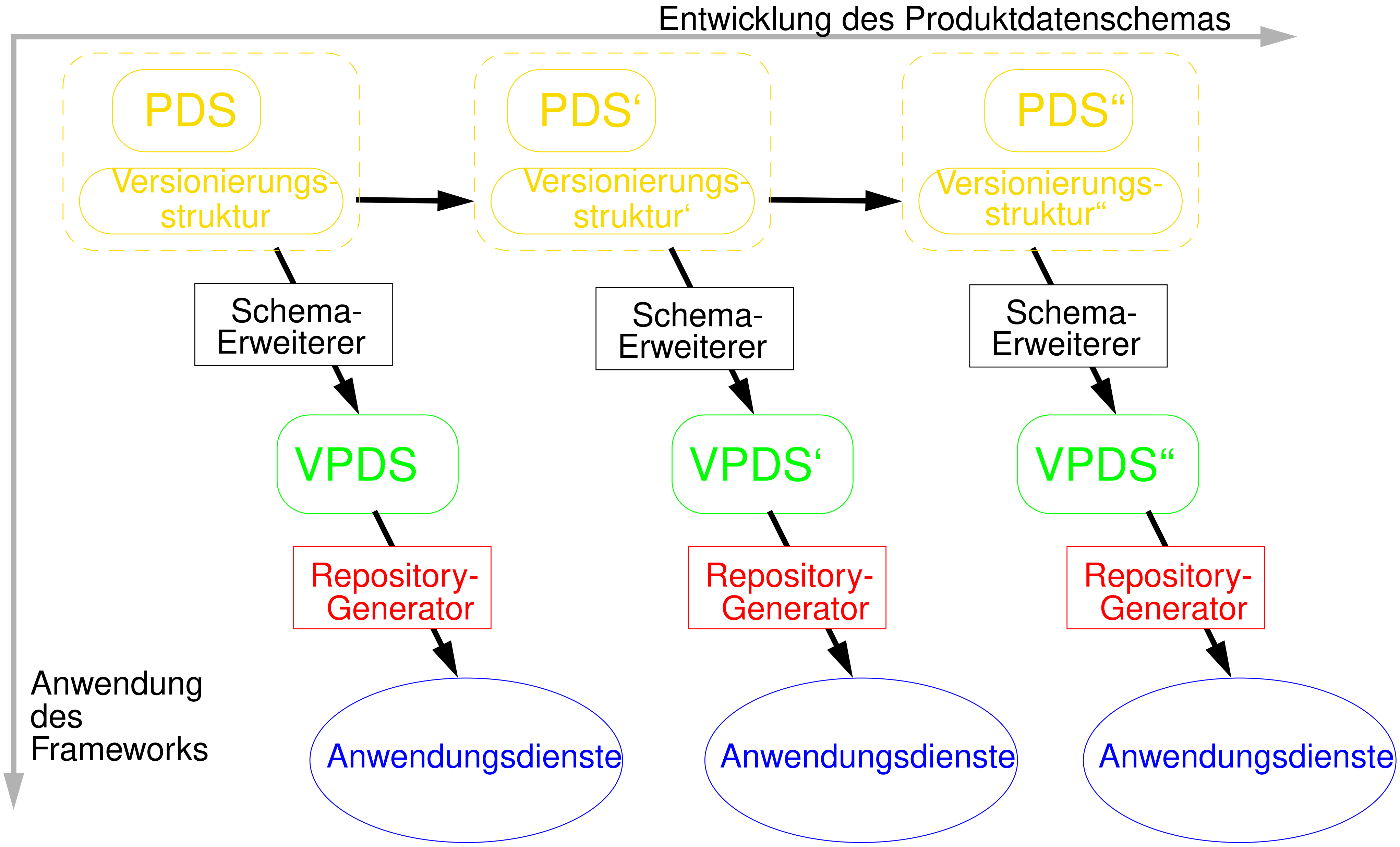
Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Anwendung eines Versionierungs-Frameworks



Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

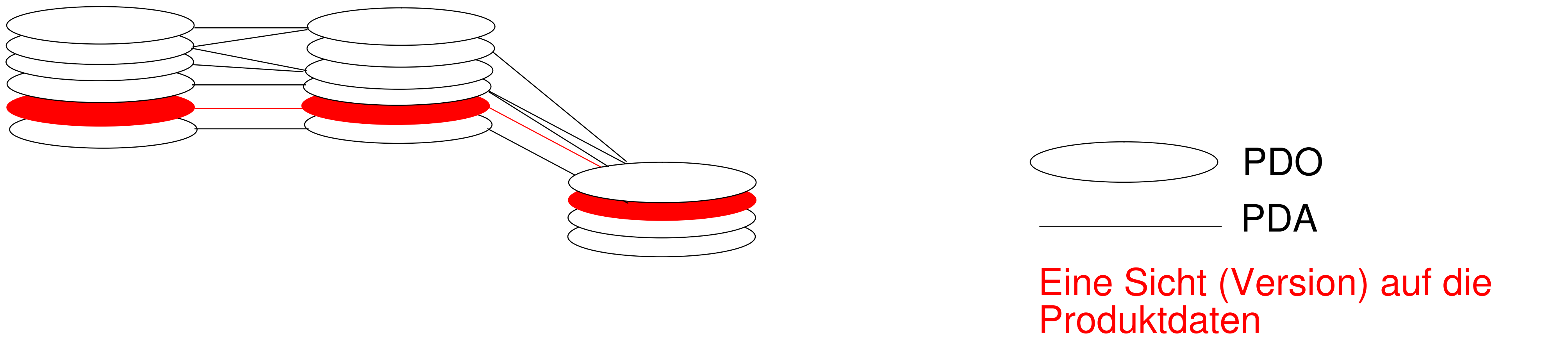
Weiterentwicklung des Produktdatenschemas



Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

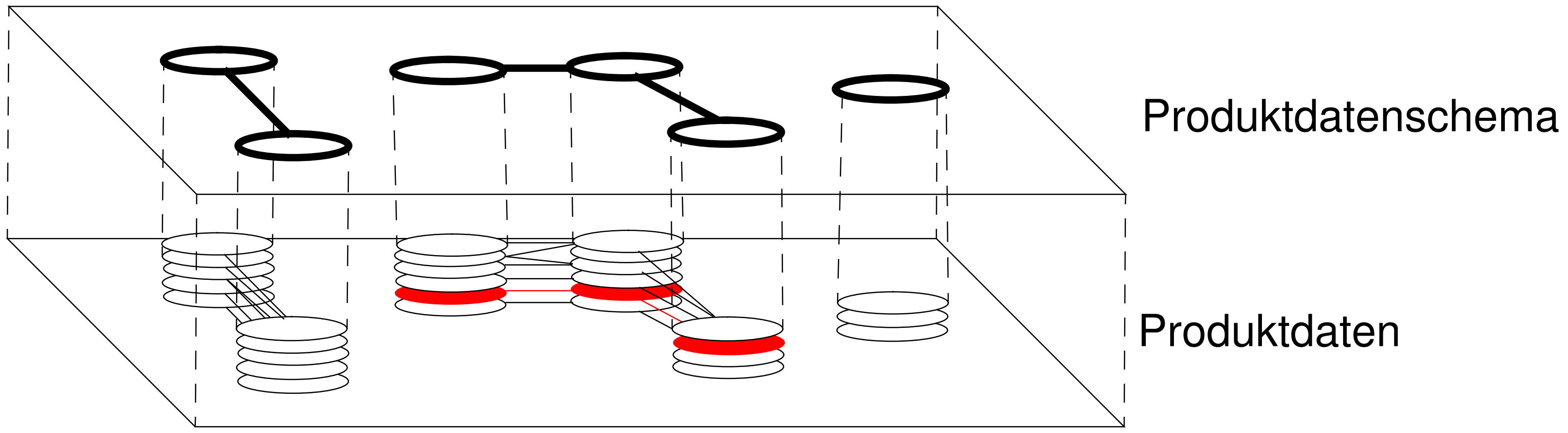
Grundkonzept des Versionierungs-Frameworks

- Versionierung als Sichtenbildung auf Produktdaten
- Produktdaten bestehen aus:
  - ➥ Objekten (Produktdatenobjekte - POD)
  - ➥ Beziehungen zwischen diesen Objekten (Produktdatenassoziationen - PDA)



Grundkonzept des Versionierungs-Frameworks

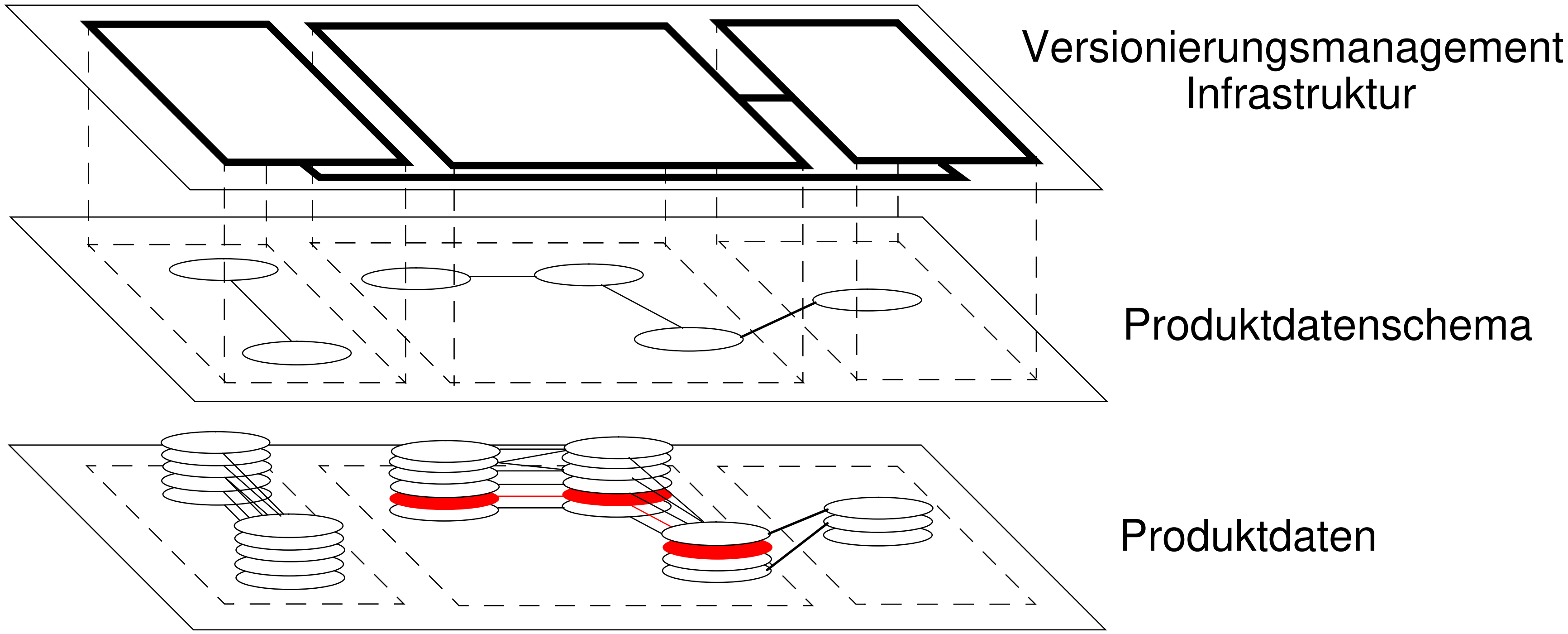
- Struktur der Produktdaten ist in einem Produktdatenschema (PDS) festgelegt



Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Grundkonzept des Versionierungs-Frameworks

- Versioniert werden Ausschnitte (Versionierbare Strukturen - VS) des PDS
  - ➥ Die VS werden in der Versionierungsmanagement Infrastruktur (VMI) festgelegt
  - ➥ Zwischen den VS können Beziehungen bestehen

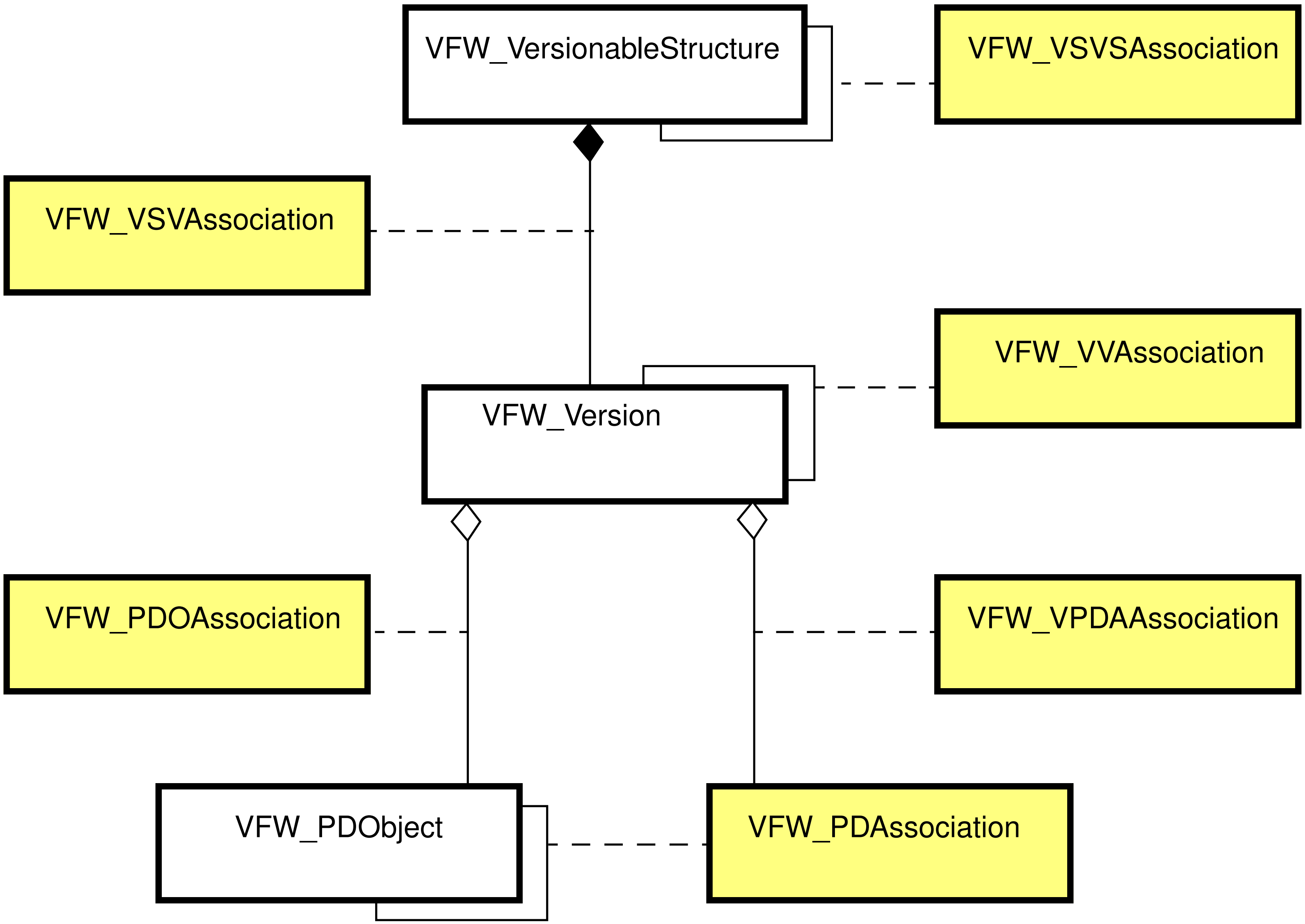


Elemente des Versionierungs-Frameworks

- Produktdaten
  - ➥ Produktdatenobjekte (PDO)
  - ➥ Produktdatenassoziationen als eigene Objekte, die versioniert werden können (PDA)
- Versionierungsmanagement Infrastruktur (VMI)
  - ➥ Versionierbare Strukturen (VS)
  - ➥ Beziehungen zwischen diesen Strukturen (VSVS_Association)
- Zur Verwaltung der Versionen
  - ➥ Versionen (V)
  - ➥ Beziehungen zwischen Versionen (VV_Association)
  - ➥ Zuordnung der Versionen zu den VS (VVS_Assciation)
  - ➥ Zuordnung von PDOs zu Versionen (VPO_Association)
  - ➥ Zuordnung von PDAs zu Versionen (VPDA_Association)

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Elemente des Versionierungs-Frameworks

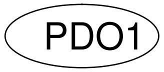


Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Mögliche Anwendungsfälle im VFW

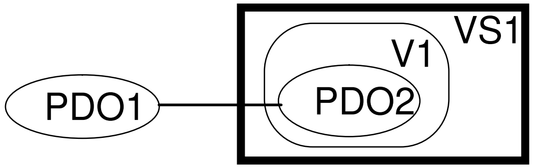
Produktdatenobjekte

- Nichtversionierte PDOs
  - ❍ kein Bezug zu versionierten PDOs



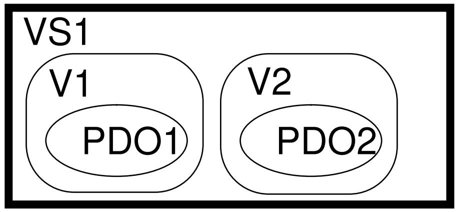
=> Object_NonVersioned

  - ❍ Bezug zu versionierten PDOs



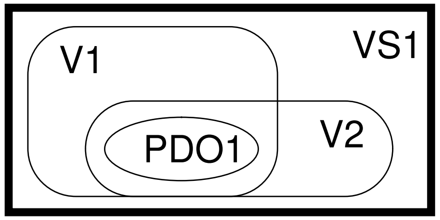
=> PDO_NonVersioned

- PDO in max. einer Version
  - ➥ kann nicht in zwei VS liegen

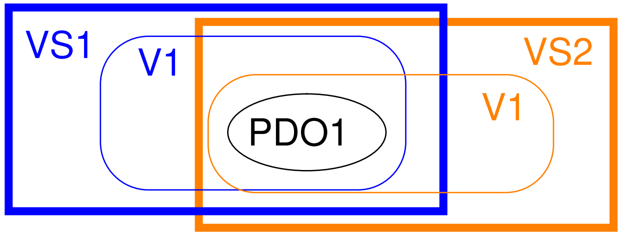


=> PDO_Exclusive

- PDO in mehreren Versionen
  - ❍ der gleichen VS



  - ❍ verschiedener VS



=> PDO_Shared

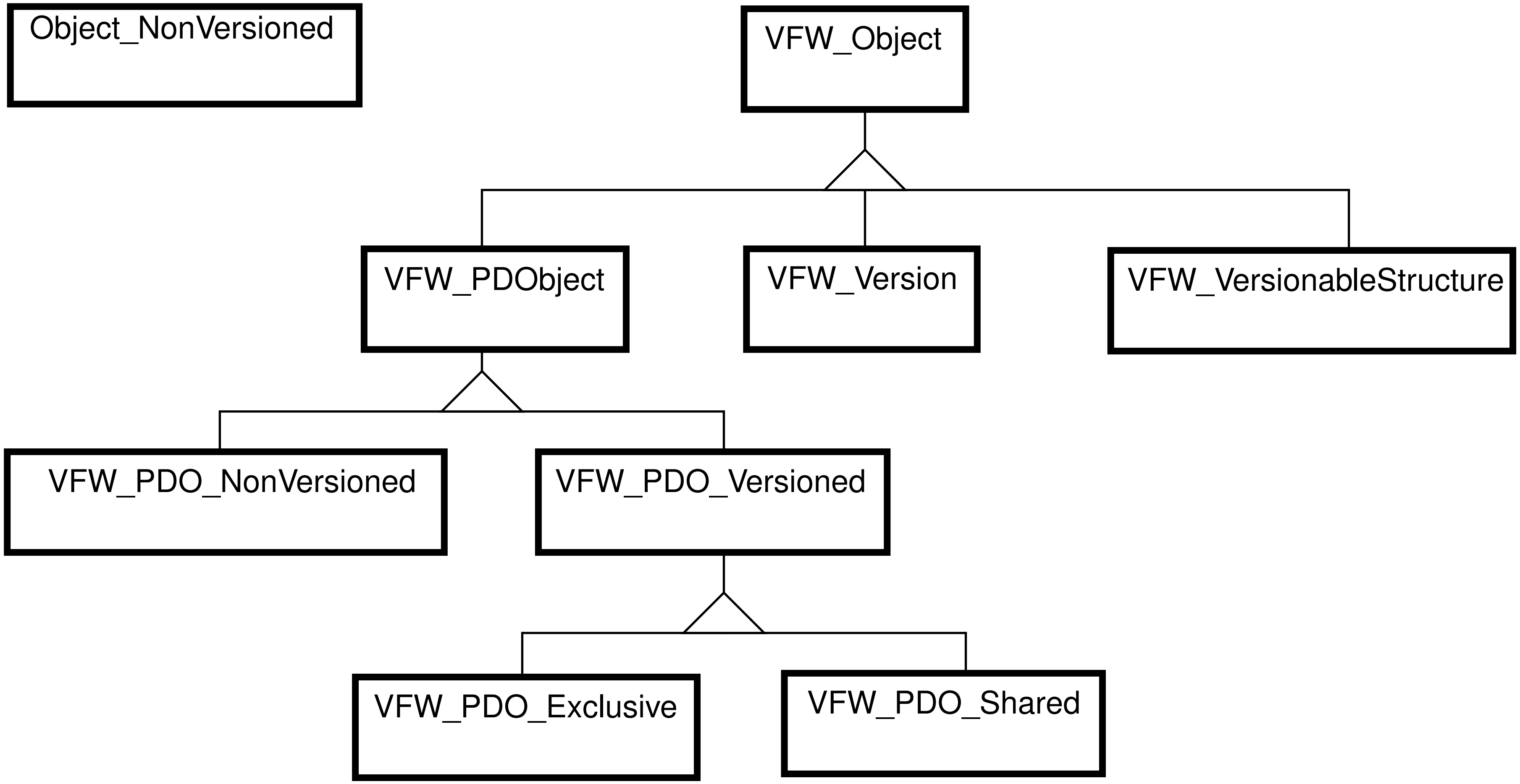
UNIVERSITÄT KAISERSLAUTERN

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

SERUM

DBIS Datenbanken und Informationssysteme

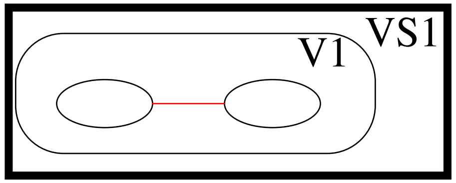
Objekt-Pattern mit Vererbungshierarchie



Mögliche Anwendungsfälle im VFW

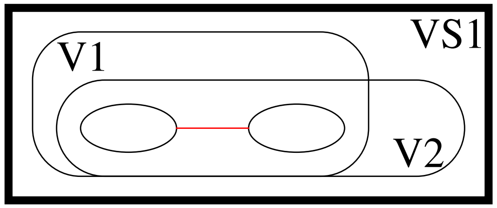
Produktdatenbeziehungen

- PDA in genau einer Version
  - ➥ PDOs in genau einer Version

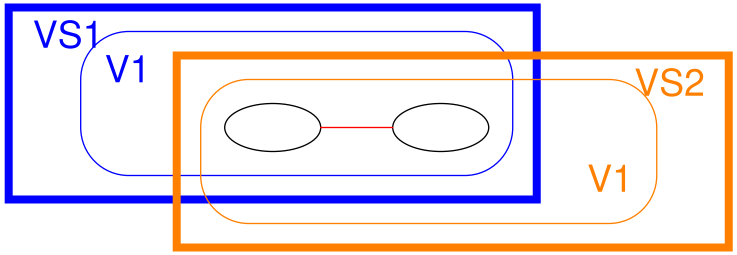


=> PDA_Exclusive

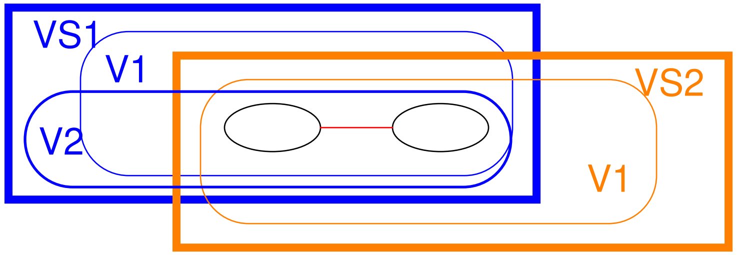
- PDA in verschiedenen Versionen
  - ❍ einer VS
    - ➥ PDOs in verschiedenen Versionen einer VS



  - ❍ verschiedener VS
    - ➥ PDOs in verschiedenen VS



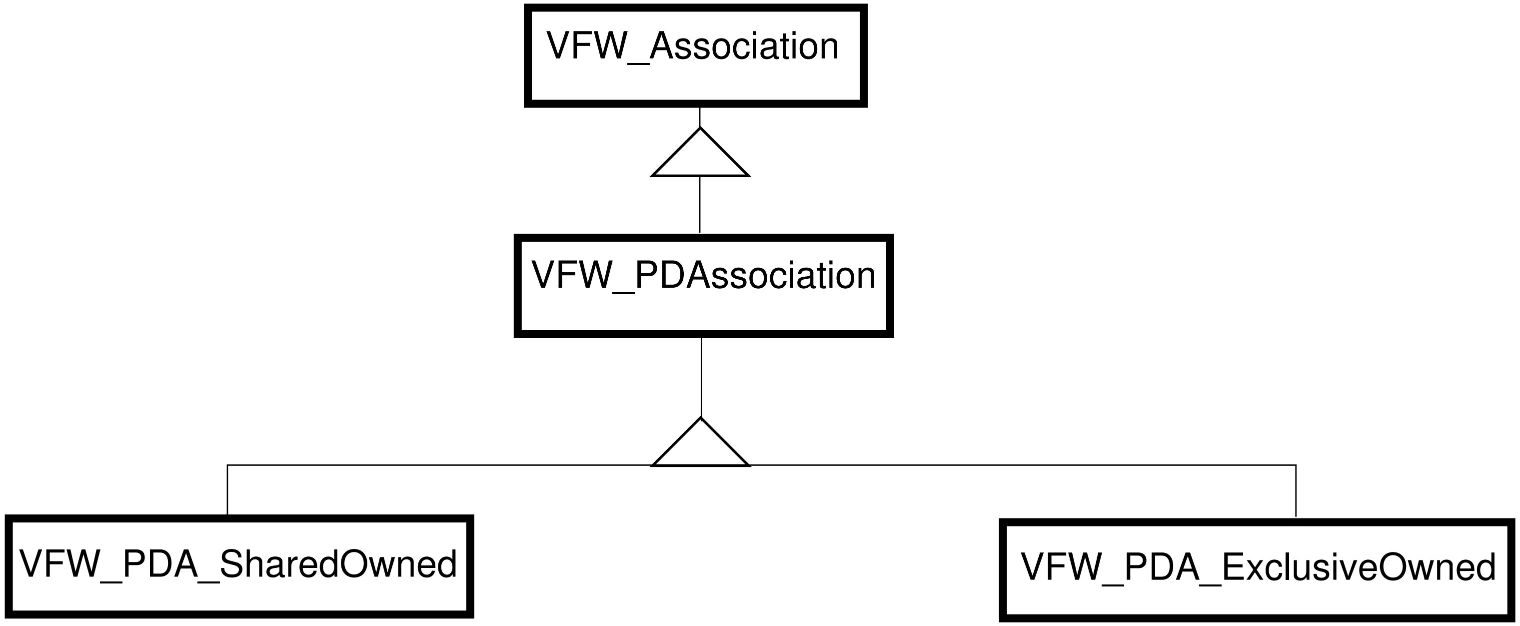
  - ❍ verschiedener VS und innerhalb einer VS
    - ➥ PDOs in verschiedenen VS
    - ➥ PDOs in verschiedenen Versionen einer VS



=> PDA_Shared

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

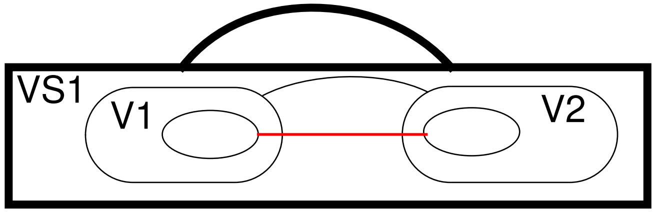
Assocation-Pattern mit Vererbungshierarchie



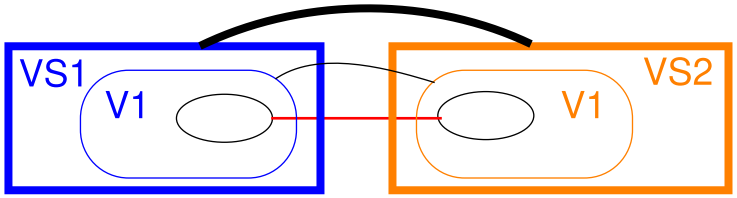
Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Mögliche Anwendungsfälle im VFW

- PDA zwischen Versionen (PDOs in verschiedenen Versionen)
  - ❍ innerhalb einer VS
    - ➥ Beziehung zwischen den Versionen (keine Versionsfortführung)
    - ➥ Beziehung der VS mit sich selbst

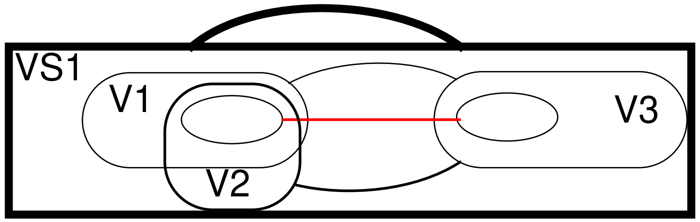


  - ❍ verschiedener VS
    - ➥ Beziehungen zwischen den Versionen
    - ➥ Beziehungen zwischen den VS

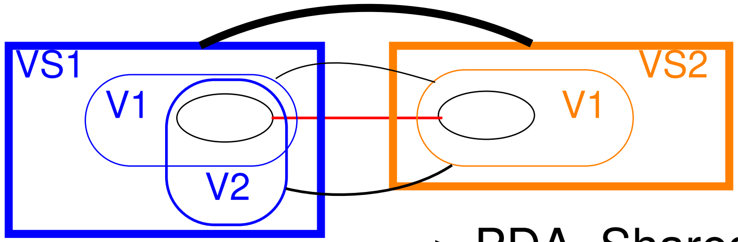


=> PDA_ExclusiveRefinement

- PDA zwischen Versionen (PDOs können in mehreren Versionen liegen)
  - ❍ innerhalb einer VS



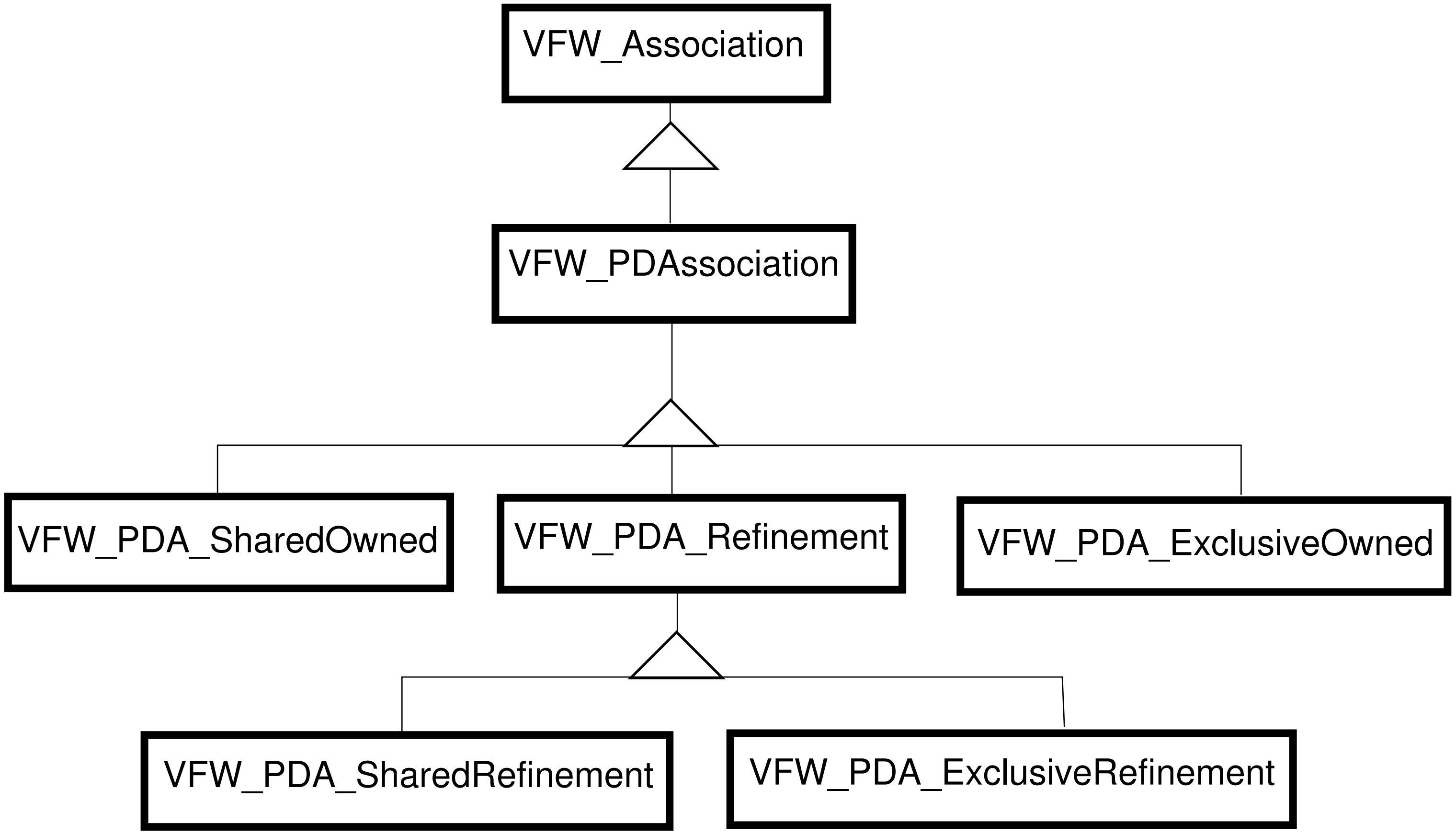
  - ❍ verschiedener VS



=> PDA_SharedRefinement

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

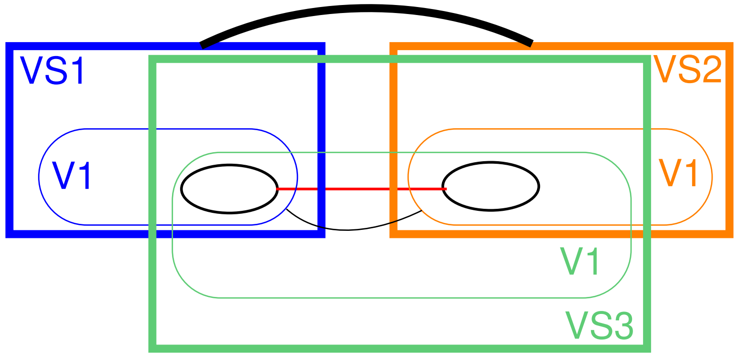
Assocation-Pattern mit Vererbungshierarchie



Mögliche Anwendungsfälle im VFW

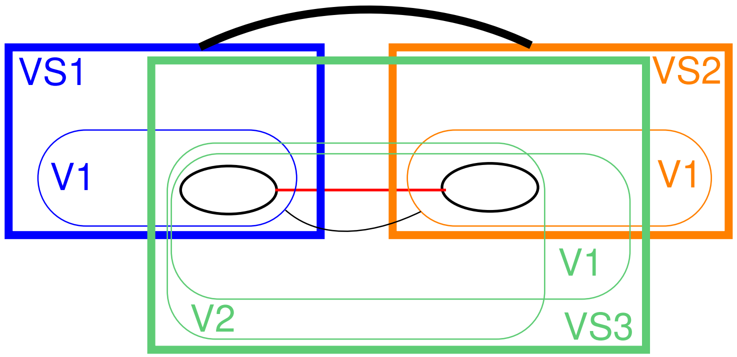
Produktdatenbeziehungen

- PDAs die sowohl zwischen Versionen als auch in Versionen liegen können
  - ❍ In max. einer Version und max. einmal zwischen zwei Versionen



=> PDA_ERSO

  - ❍ In mehreren Versionen und max. einmal zwischen zwei Versionen



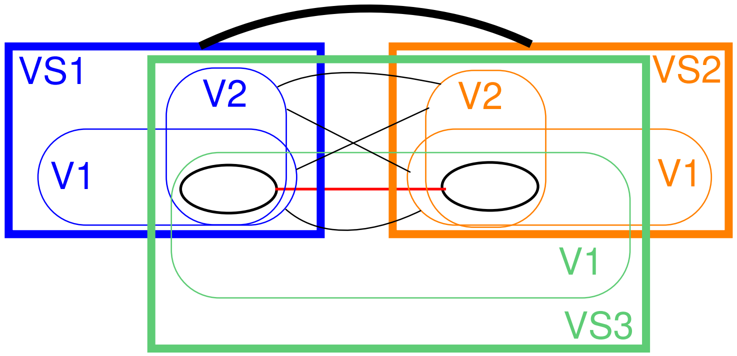
=> PDA_ERSO

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Mögliche Anwendungsfälle im VFW

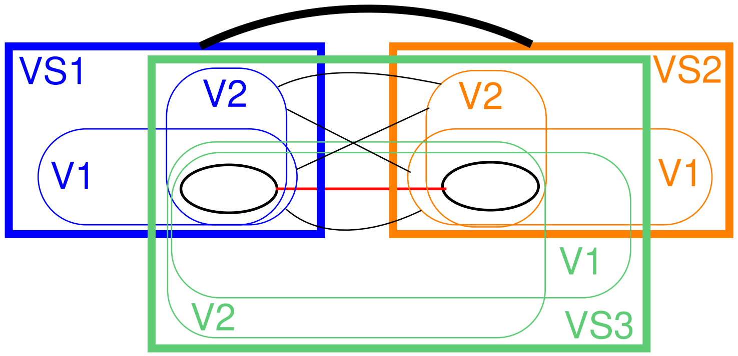
Produktdatenbeziehungen

- In max. einer Version und zwischen mehreren Versionen



=> PDA_SREO

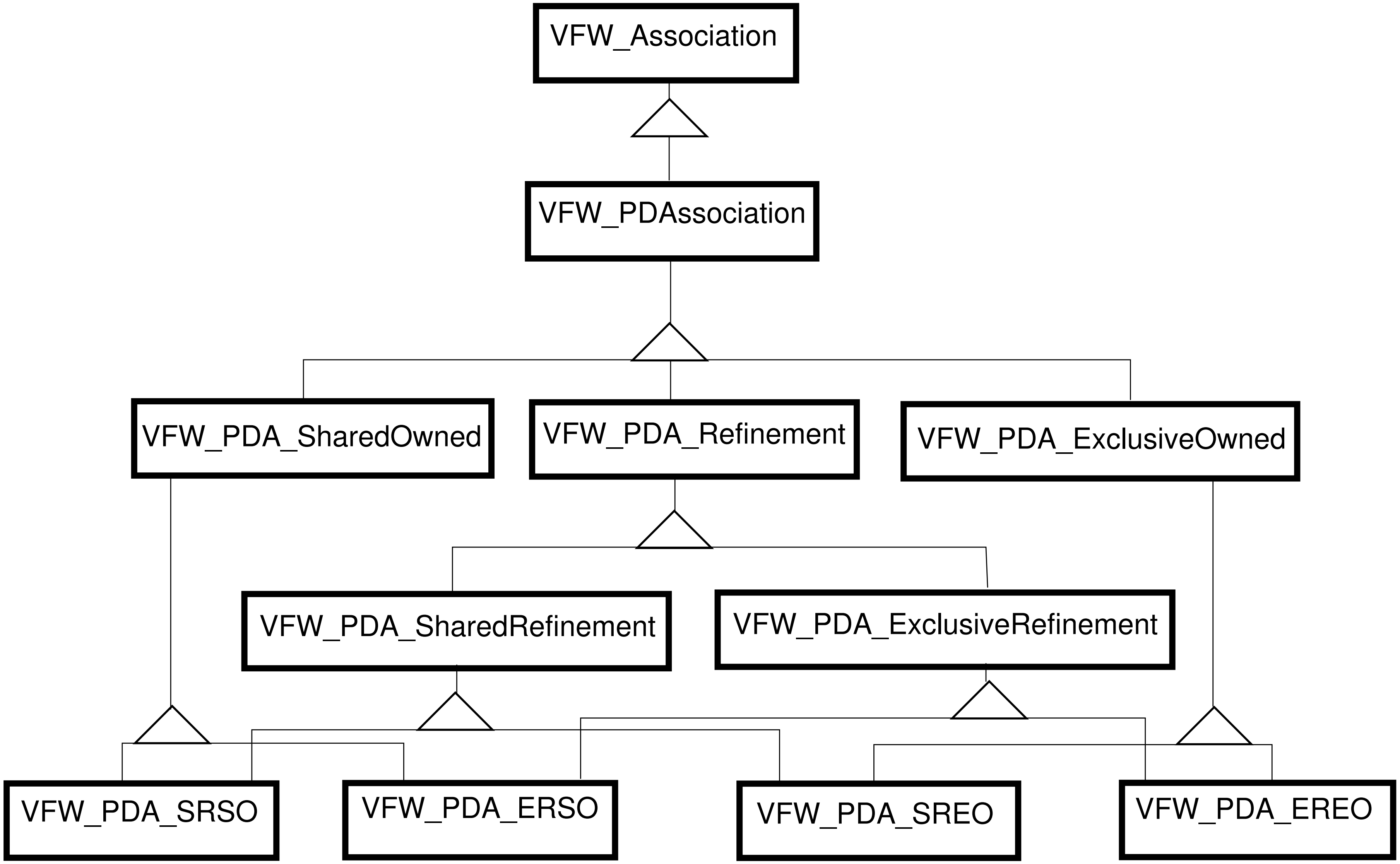
- In mehreren Versionen und zwischen mehreren Versionen



=> PDA_SRSO

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Assocation-Pattern mit Vererbungshierarchie



Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Mögliche Anwendungsfälle im VFW

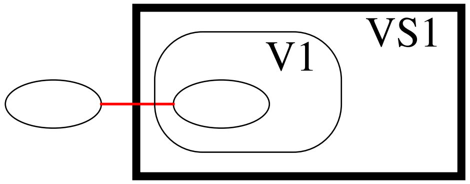
Produktdatenbeziehungen

- PDA zwischen nichtversionierten PDOs

  ➥ PDA nicht versioniert

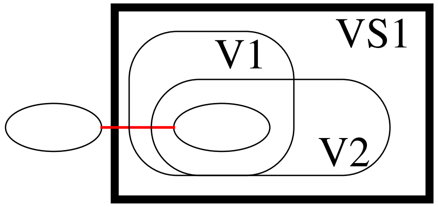
=> Association_NonVersioned

- PDA zwischen versionierten und nicht-versionierten PDOs
  - ❍ versionierte PDO ist PDO_Exclusive



=> PDA_NV_to_EO

  - ❍ versionierte PDO ist PDO_Shared



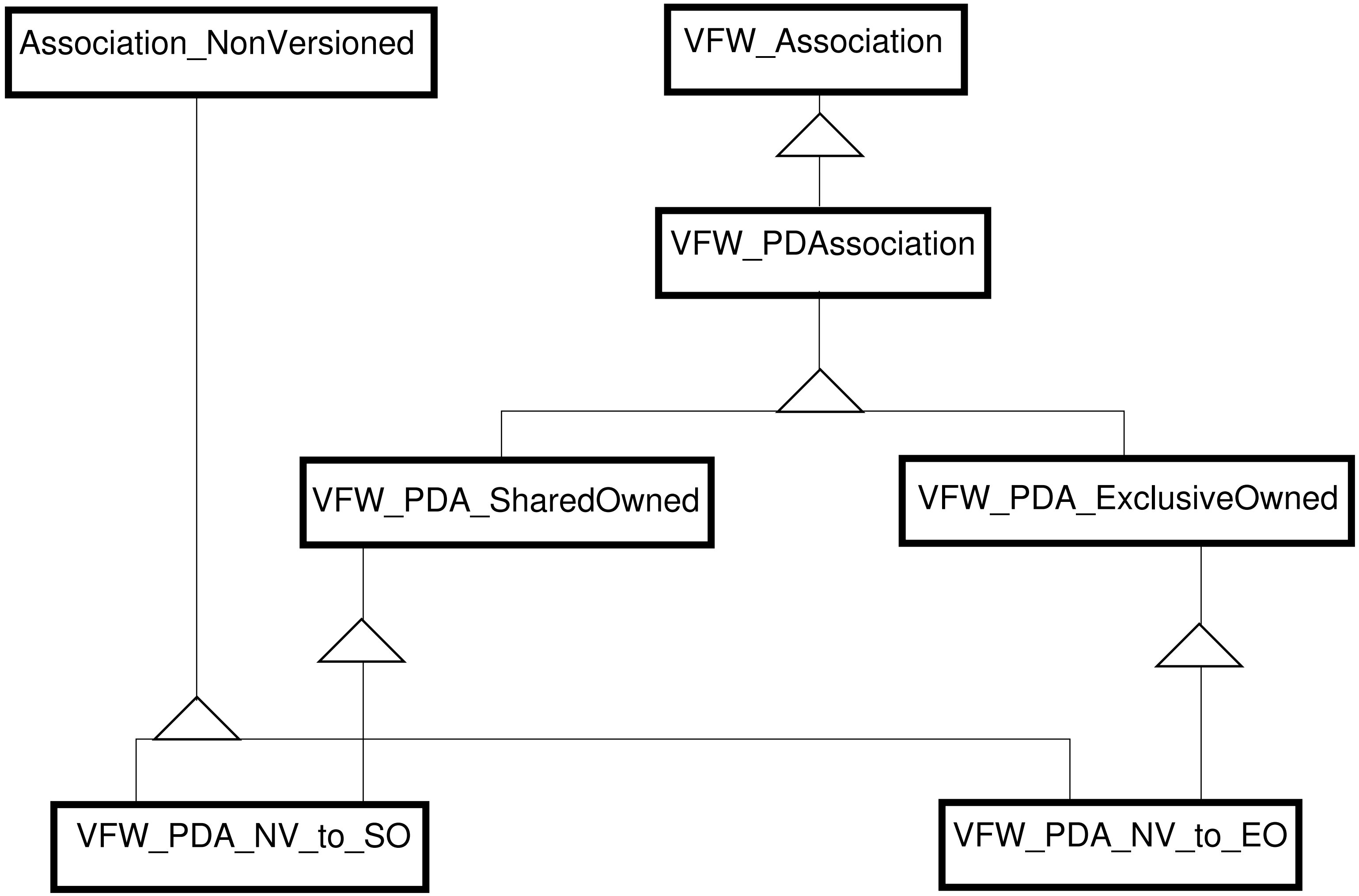
=> PDA_NV_to_SO

  ➥ keine Beziehung zwischen PDO_NonVersioned und VS

  ➥ PDA gilt von Versionierungssemantik als in der Version liegend

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Assocation-Pattern mit Vererbungshierarchie



Rückschlüsse auf PDAs aufgrund der PDOs

- PDAs zwischen versionierten PDOs

		in Version		zwischen Versionen		in und zwischen Versionen			
PDO1	PDO2	Exlusive	Shared	ER	SR	EREO	ERSO	SREO	SRSO
Exclusive	Exclusive	❍	✗	❍	✗	❍	✗	✗	✗
Exclusive	Shared	❍	✗	✔	✔	✔	✗	✔	✗
Shared	Shared	✔	✔	✔	✔	✔	✔	✔	✔

- PDAs zwischen versionierten und nichtversionierten PDOs

PDO1	PDO2	NV_to_EO	NV_to_SO
NV	Exclusive	❍	✗
NV	Shared	✔	✔

❍ = einzige Möglichkeit

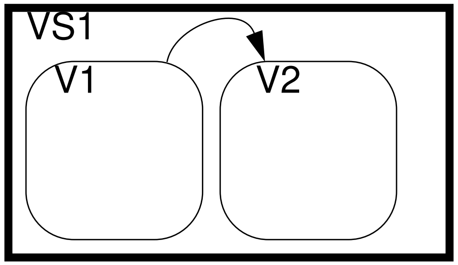
✔ = eine Möglichkeit

✗ = nicht möglich

Mögliche Anwendungsfälle im VFW

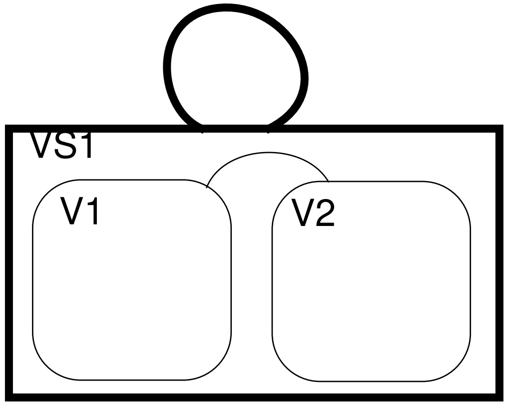
Versionsbeziehungen

- Versionsfortführung

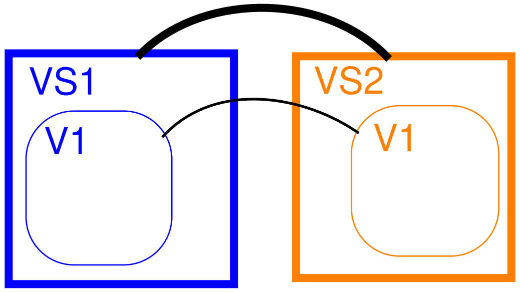


=> VVA_Evolution

- Verfeinerung von VSVS_Assoziationen
  - der gleichen VS



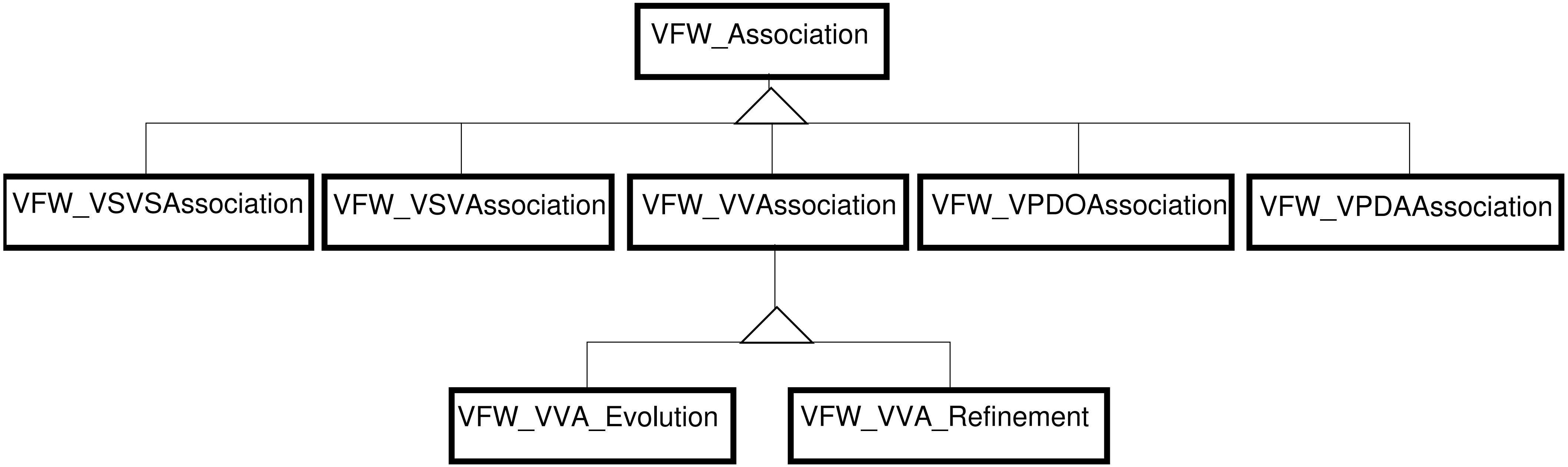
  - verschiedener VS



=> VVA_Refinement

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Assocation-Pattern mit Vererbungshierarchie

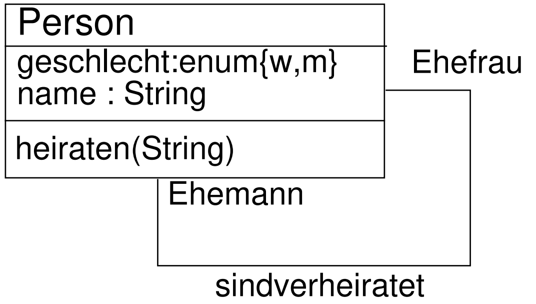


Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Ausblick

Erweiterung der OCL-Constraints des PDO zur Einhaltung der Versionierungssemantik

Produktdatenschema (Ausschnitt)



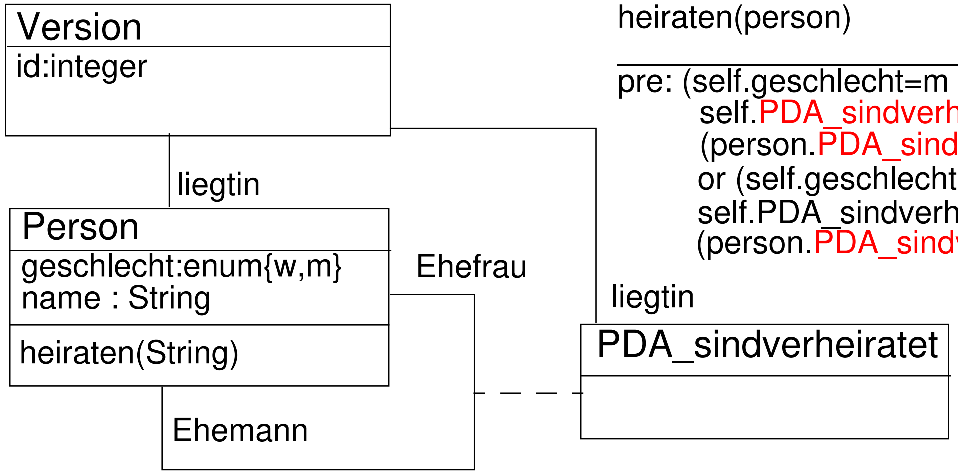
heiraten(person)

pre: (self.geschlecht=m and person.geschlecht=w and
self.Ehemann->isEmpty and person.Ehefrau->isEmpty)
or (self.geschlecht=w and person.geschlecht=m and
self.Ehefrau->isEmpty and person.Ehemann->isEmpty)

Schemaerweiterung

Erweiterung der OCL-Constraints

VersioniertesProduktdatenschema (Ausschnitt)



heiraten(person)

pre: (self.geschlecht=m and person.geschlecht=w and
self.PDA_sindverheiratet.liegtin->forAll(id<>self.id) and
(person.PDA_sindverheiratet.liegtin->forAll(id<>self.id)))
or (self.geschlecht=w and person.geschlecht=m and
self.PDA_sindverheitatet.liegtin->forAll(id<>self.id) and
(person.PDA_sindverheiratet.liegtin->forAll(id<>self.id)))

(PDA_Shared)

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder

Ausblick II

- Definieren der Pattern in geeigneter Sprache (UML-Template)
- Implementierung von technologieabhängigen Templates (Java, C++, ORDBMS)
- Erweiterung der OCL-Constraints
  - ➥ Wie müssen PDS-Constraints erweitert werden, damit sie auch im VPDS gelten?
  - ➥ Kann mit OCL-Constraints die Versionierung kontrolliert werden?
  - ➥ Was sind gültige PDS?
    - ➥ Modellierungsregeln
    - ➥ Welche OCL-Constraints auf Schemaebene?
- Wie komme ich vom erweiterten UML-Schema zur ORDB?
- Weiteres Vorgehen
  - ➥ Entwicklung eines UML-Repository „von Hand“
  - ➥ Entwicklung der Werkzeuge, mit denen dieses erstellt wird
  - ➥ Entwicklung weiterer Repository
  - ➥ Erweiterung der Werkzeuge

Wolfgang Mahnke
e-mail: mahnke@informatik.uni-kl.de
http://www.uni-kl.de/AG-Haerder