OPUS

Tarifsoftware - Verwaltungssoftware - Produktsoftware

LV 1871

Mit neuen Technologien auf dem Weg zu neuen Märkten

Entwicklungsbausteine eines durchgängigen Java-basierten Lebensversicherungssystems

Siegfried Löhr, LV 1871 a. G., München
Leiter Anwendungsentwicklung

Unternehmensdaten 1999

Lebensversicherung von 1871 a. G., München
Maximiliansplatz 5
80333 München

Beitragseinnahmen	**372 Mio**
Bestand	**12.266 Mio**
Kapitalanlagen	**3.300 Mio**
Nettoverzinsung	**8,4%**
Bewertungsreserven	**23,9 %**
Anzahl MA	**320**
Anzahl MA IT (RZ im Outsourcing)	**37**



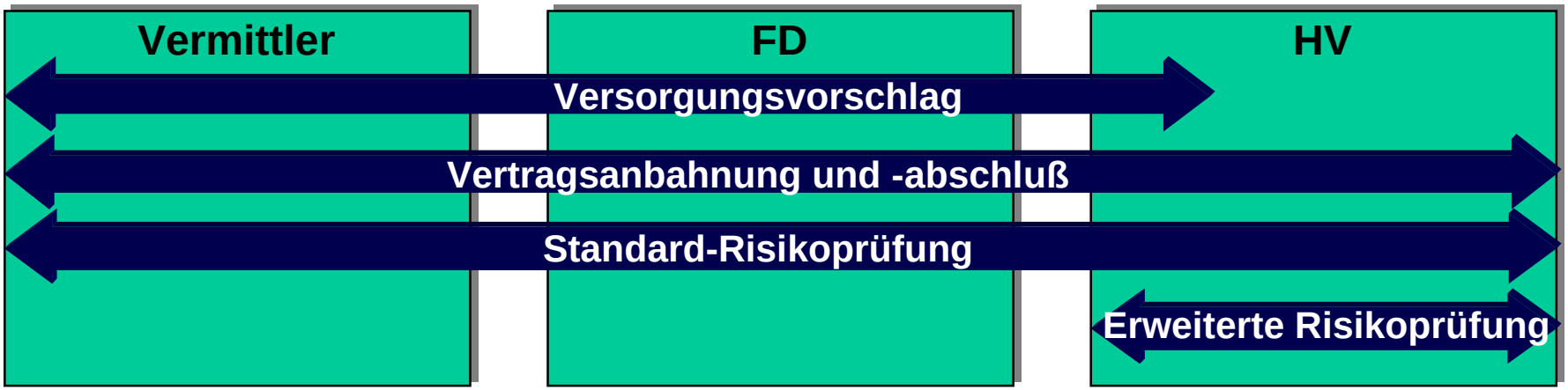
Agenda

- Prozess-Modellierung als Basis
- Objektorientierte Anwendungsentwicklung mit Java/CORBA/Frameworks
- OPUS das System
- Java - Produktionsumgebung
- Werkzeugverbund in der Entwicklung
- Software Konfigurationsmanagement

Prozess-Modellierung als Basis

Übersicht über die OPUS / VIVA Geschäftsprozesse

Kern- Prozesse OPUS



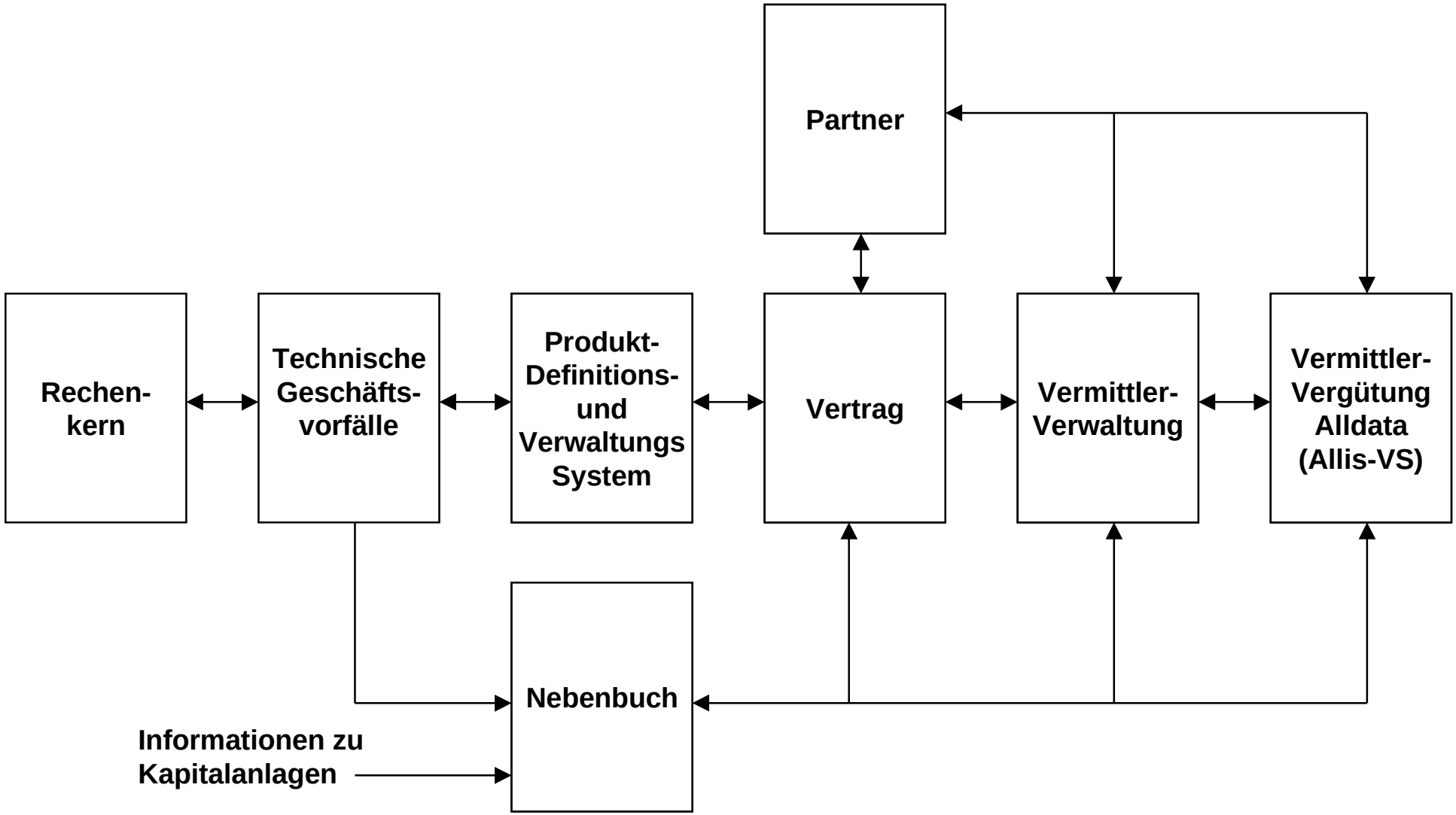
Unterstützende Prozesse



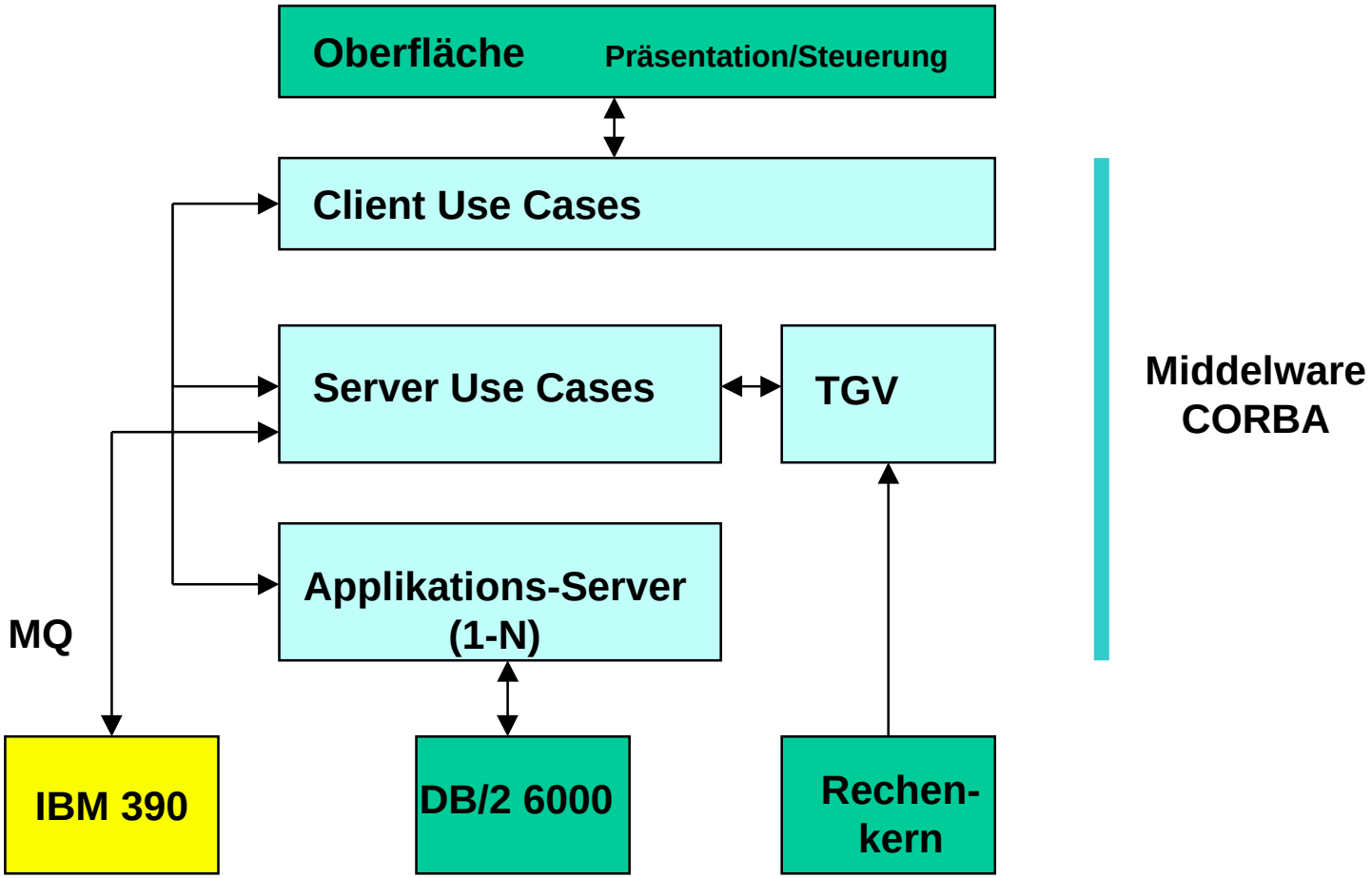
LV 1871

Objektorientierte Software-Entwicklung

Fachliche Komponenten - Objektmodell



Technische Komponenten - Schichtenarchitektur



Implementierung mit integrierten Werkzeugen

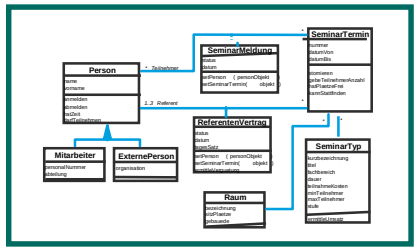
Technisches Modell

- Generische Umsetzung, z.B. mit Java

Avantis Unisuite for Java®

Objekt-Modell

- DatenModell-Schicht
- ParadigmPlus



Java Entwicklungs-werkzeug

Polar Kernel

C++, Java und SmallTalk

- Anwendungs-Klassen, z.B. GUI und Use-Cases

- Fachliche Interfaces
- Technische Hilfs-Klassen

- Polar-Kernel
- CORBA-Klassen
- diverse Klassen-Bibliotheken

Generator

- Fachliche Klassen
- MetaModell der Basis- und der fachlichen Klassen
- idl-Beschreibung

DBMS

ControlCenter

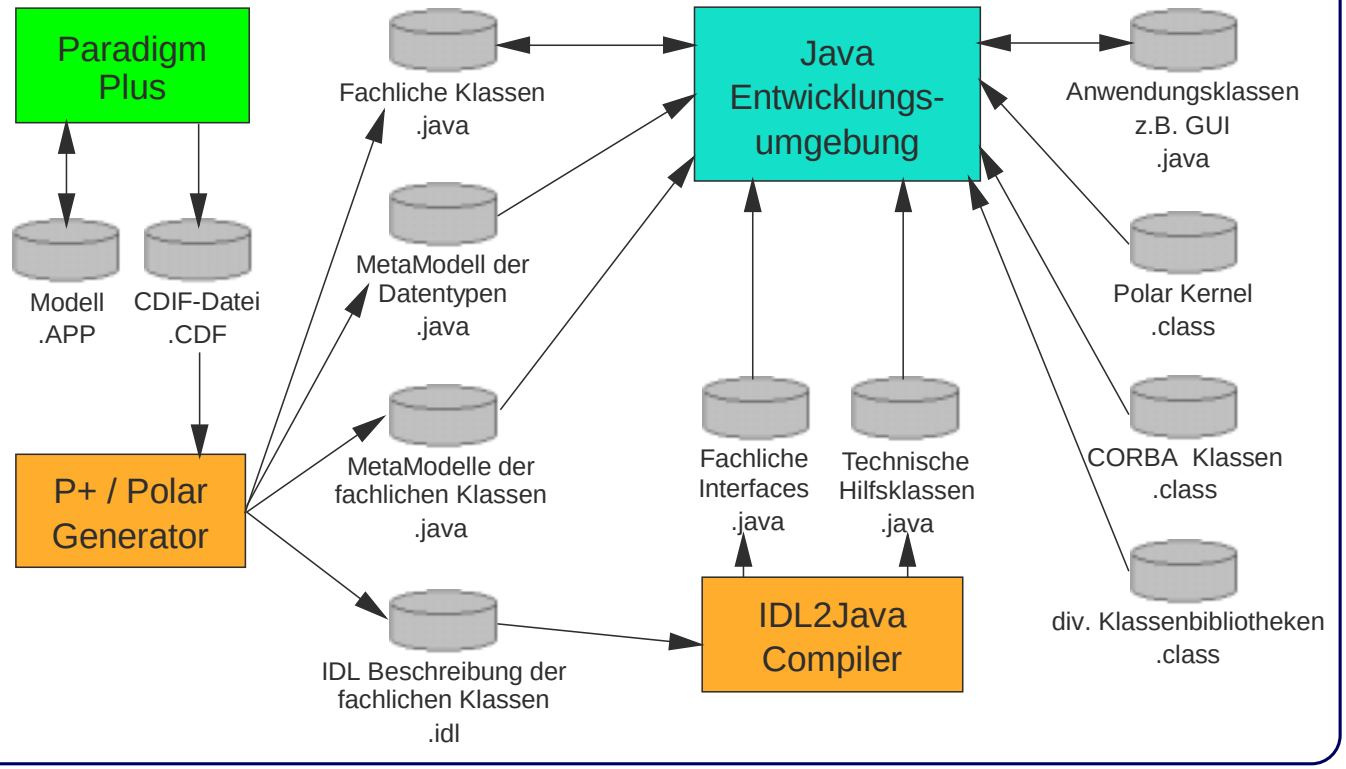
- Erzeugung von Tabelle, Indizes und Views aus den Informationen der MetaModelle

Implementierung Thin Client

Kommt eine Software-Architektur mit CORBA und Avantis Unisuite for Java® zum Einsatz, ergeben sich meist nicht zu unterschätzende Zeitaufwände für die Generierung von Quellcode. Die persistenten, fachlichen Klassen müssen mit IDL-Beschreibung generiert werden. Zusätzlich muß ein IDL Compiler eingesetzt werden, der für jede fachliche Klasse bis zu 7 technische Klassen für die CORBA Funktionalität generiert. Die folgende Abbildung zeigt die Vorgehensweise beim Einsatz von Avantis Unisuite for Java® mit CORBA.

Vorgehensweise

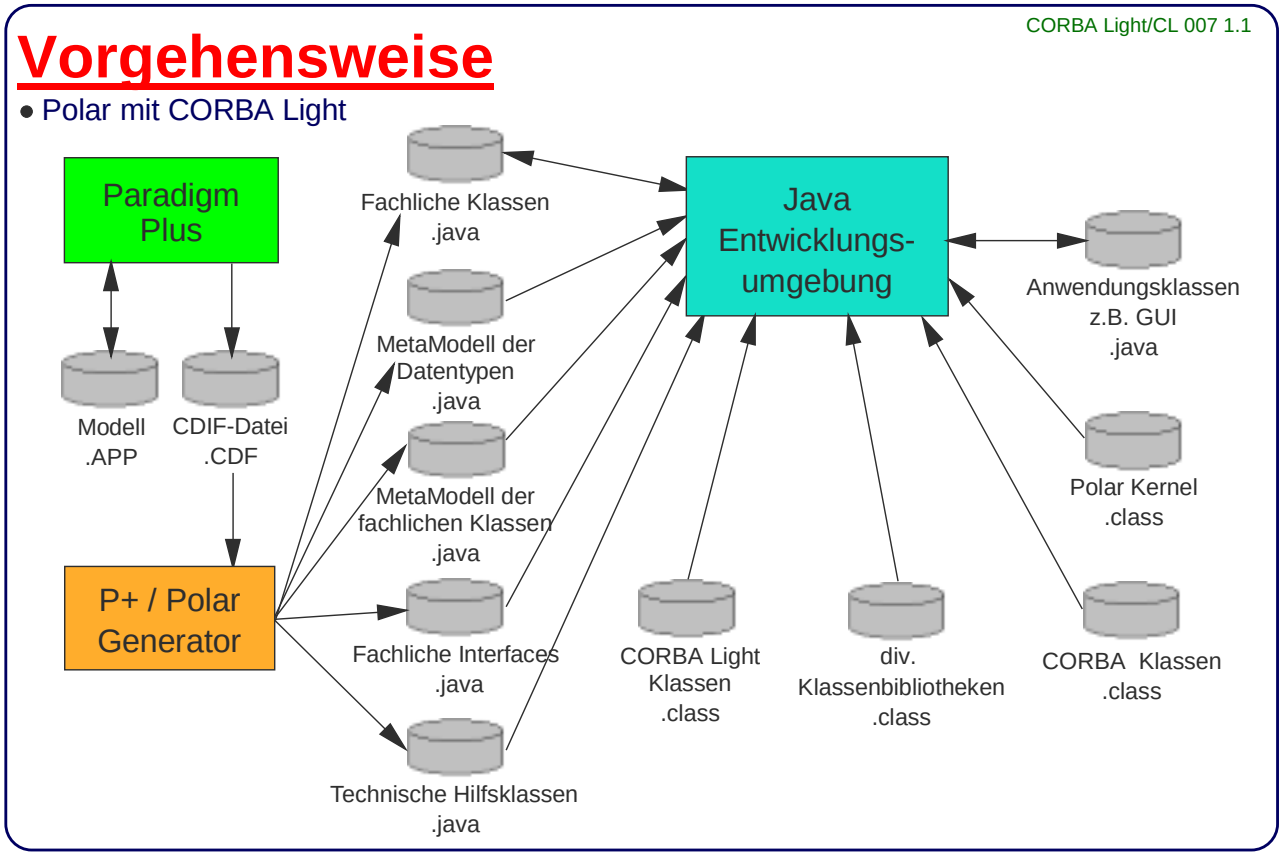
- Polar mit CORBA



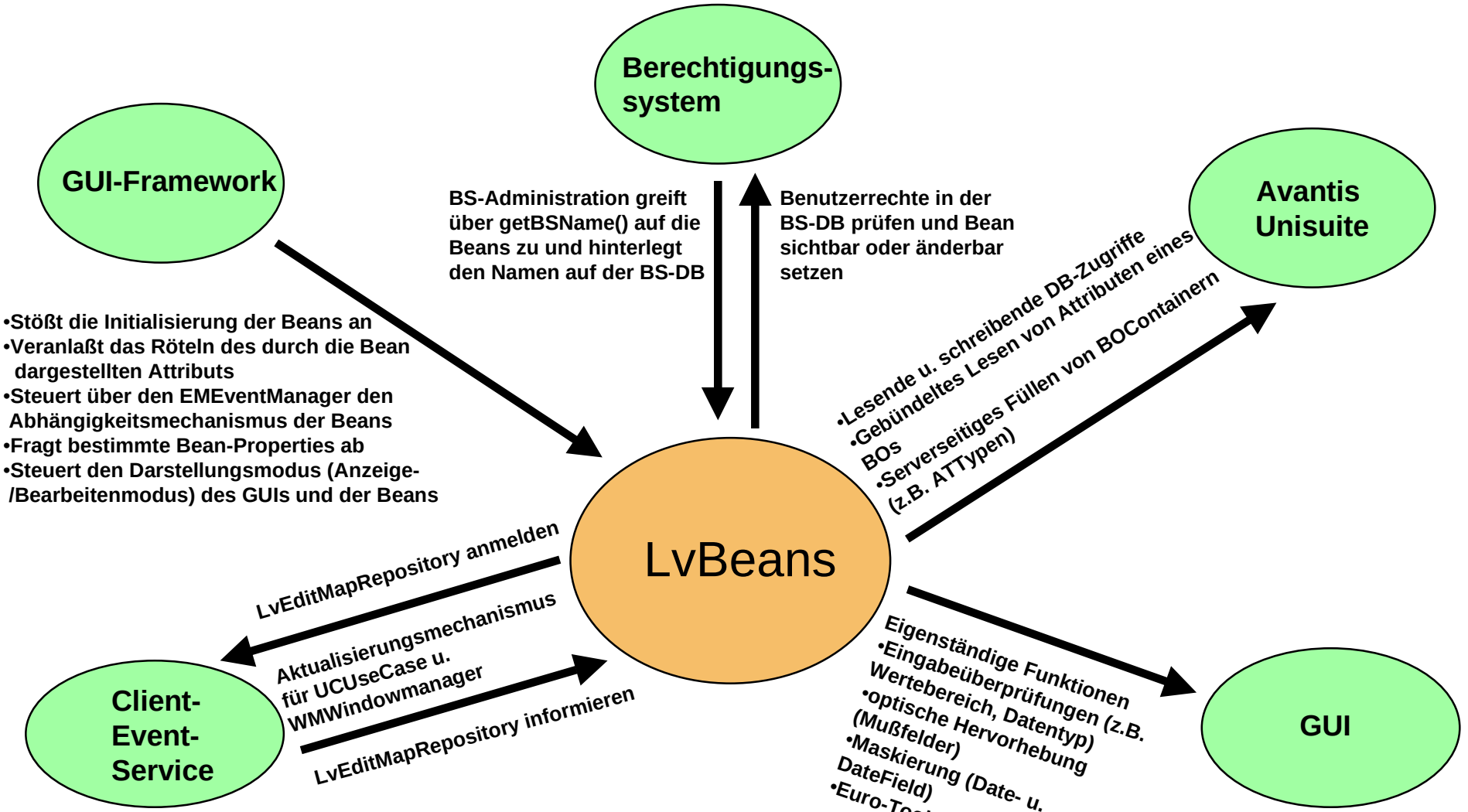
Implementierung mit Fat Client

Diese Folie beschreibt eine Software-Architektur, die es zuläßt, die Software als Fat-Client zu entwickeln, und auf einfache Weise auf eine Thin-Client Architektur mit CORBA zu wechseln - fast ohne Anpassungen am entwickelten Quellcode vornehmen zu müssen. Diese Architektur wird im folgenden als *CORBA Light* bezeichnet.

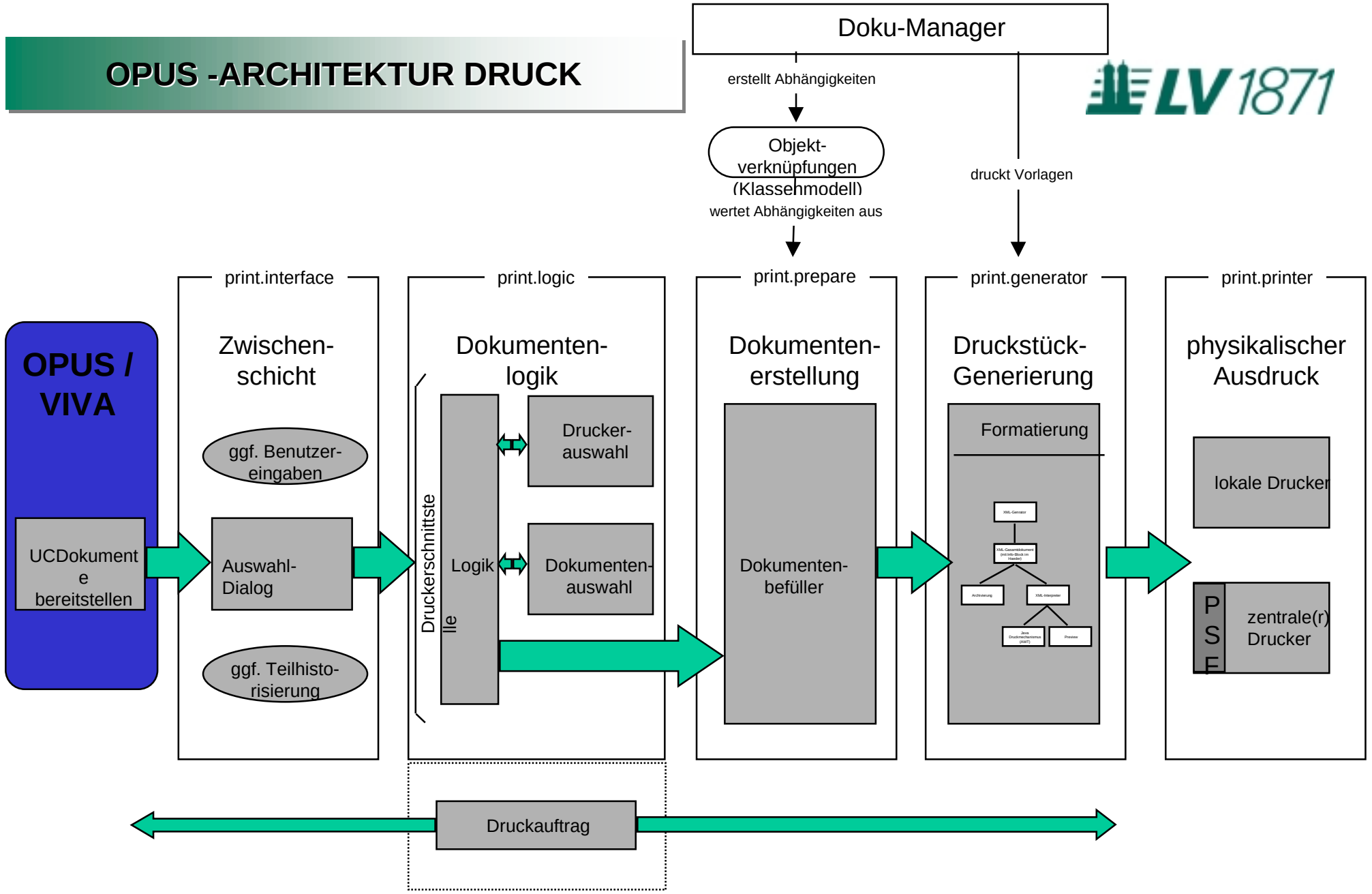
Die Vorgehensweise unterscheidet sich von Avantis Unisuite for Java® mit CORBA darin, daß die IDL Beschreibung der fachlichen Klassen und die Codegenerierung durch den IDL Compiler entfällt. Das Paradigm Plus Modell sowie die Anwendungsklassen bleiben unverändert. Lediglich die durch den P+/Polar Generator erzeugten Programmdateien sind unterschiedlich.



Die LvBeans im OPUS-Projektumfeld

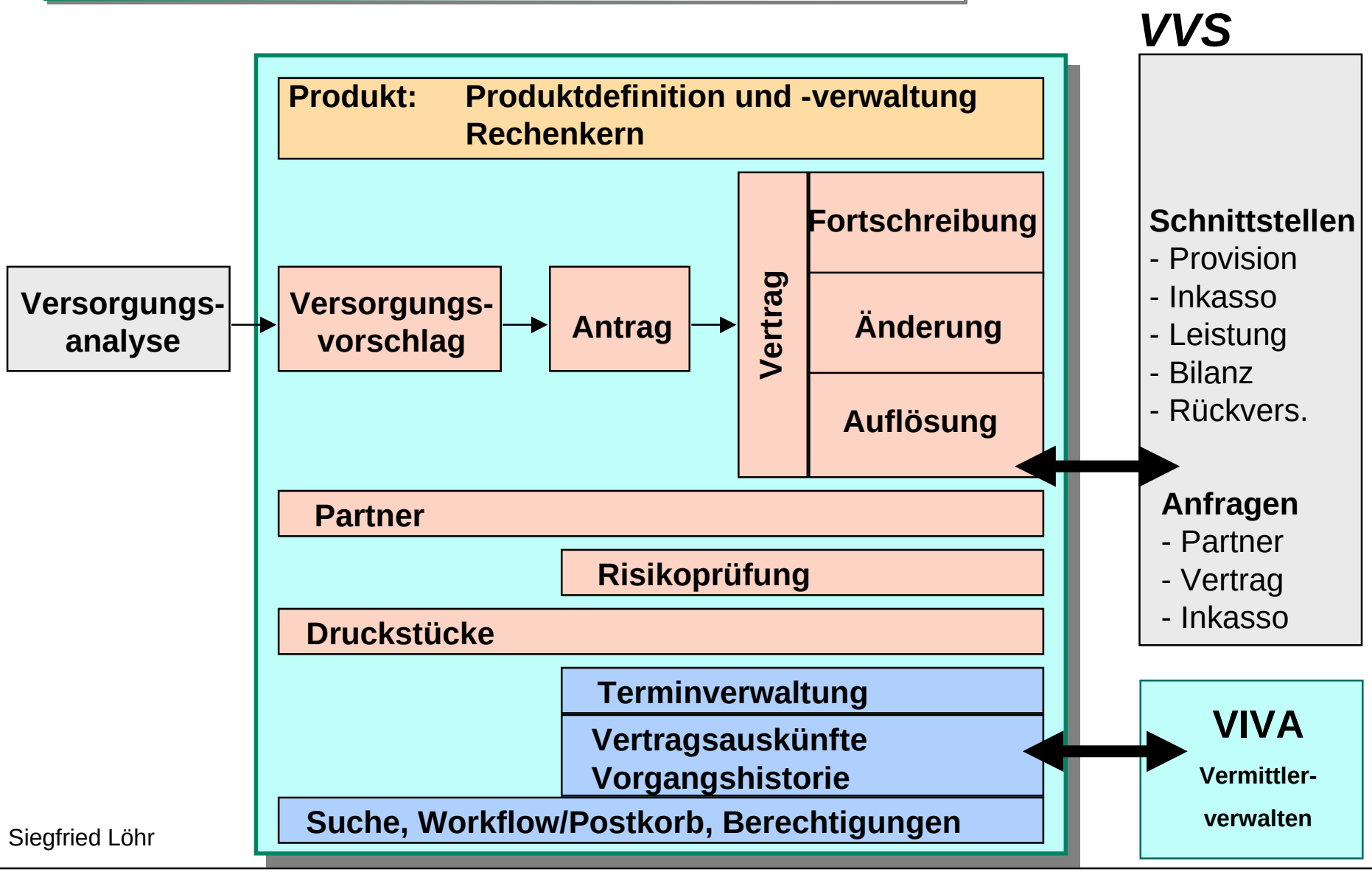


OPUS -ARCHITEKTUR DRUCK

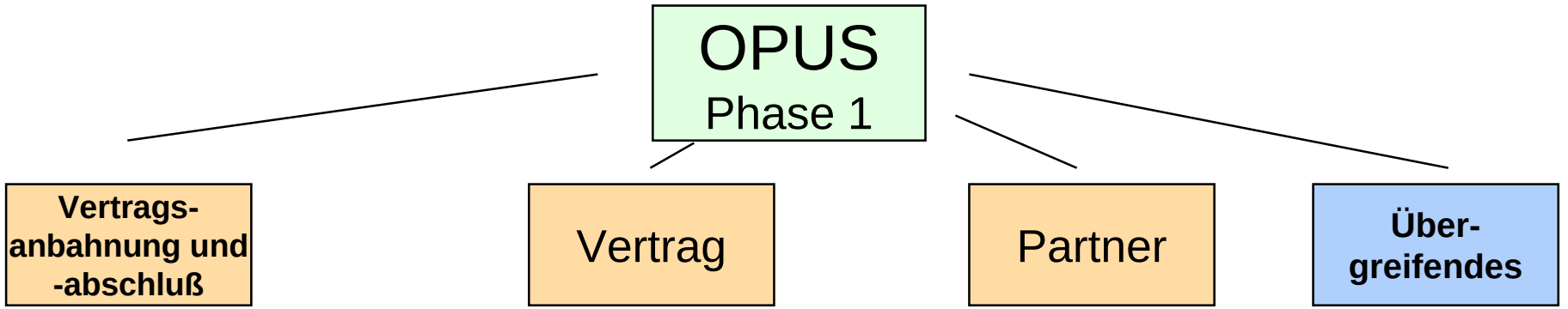


OPUS/VIVA das System

OPUS/VIVA- Überblick



OPUS- Funktionalität



Use-Cases

Vertrags-anbahnung und -abschluß

- UCVersorgungsvorschlagEinzel
- UCVersorgungsvorschlagRückdeckung
- UCNeuantragErfassenEinzel
- UCQualifiziertesRisiko

Vertrag

Vertragsänderung

- UCVertragsänderungBearbeiten (Einzel)
  - Daueränderung
  - Summenänderung
  - Dynamikaus- und -einschluß
  -

Vertragsfortschreibung

- UCFortschreibung
- UCVertragsaenderungDurchfuehren
- UCVertragszustandAktivieren
- UCAngebotLoeschen

Vertragsauflösung

- UCVertragsauflösung
- UCAbgangDurchführen

Partner

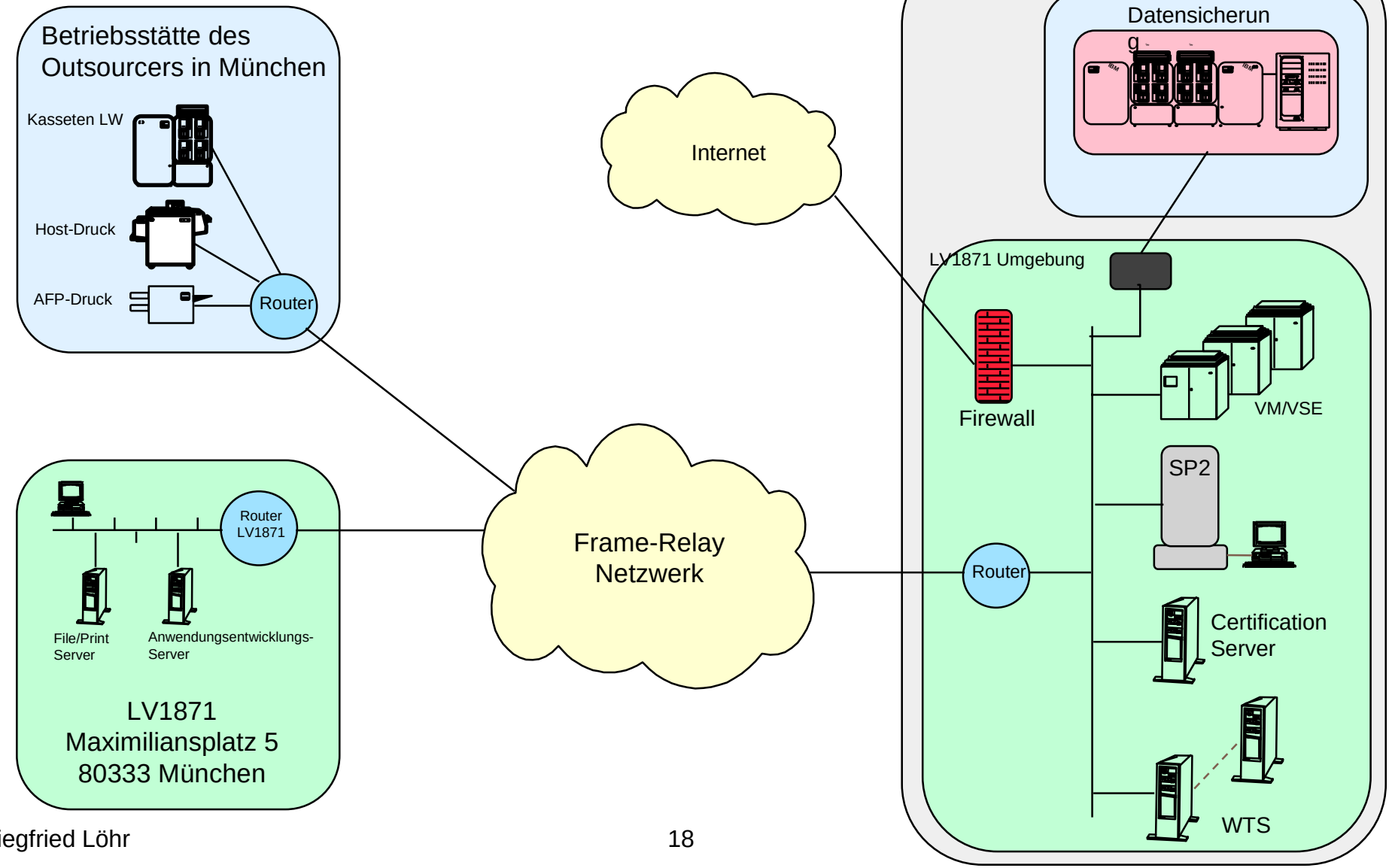
- UCPartnerSuche
- UCPartnerVerwalten
- UCTPartner/Rollen

Über-greifendes

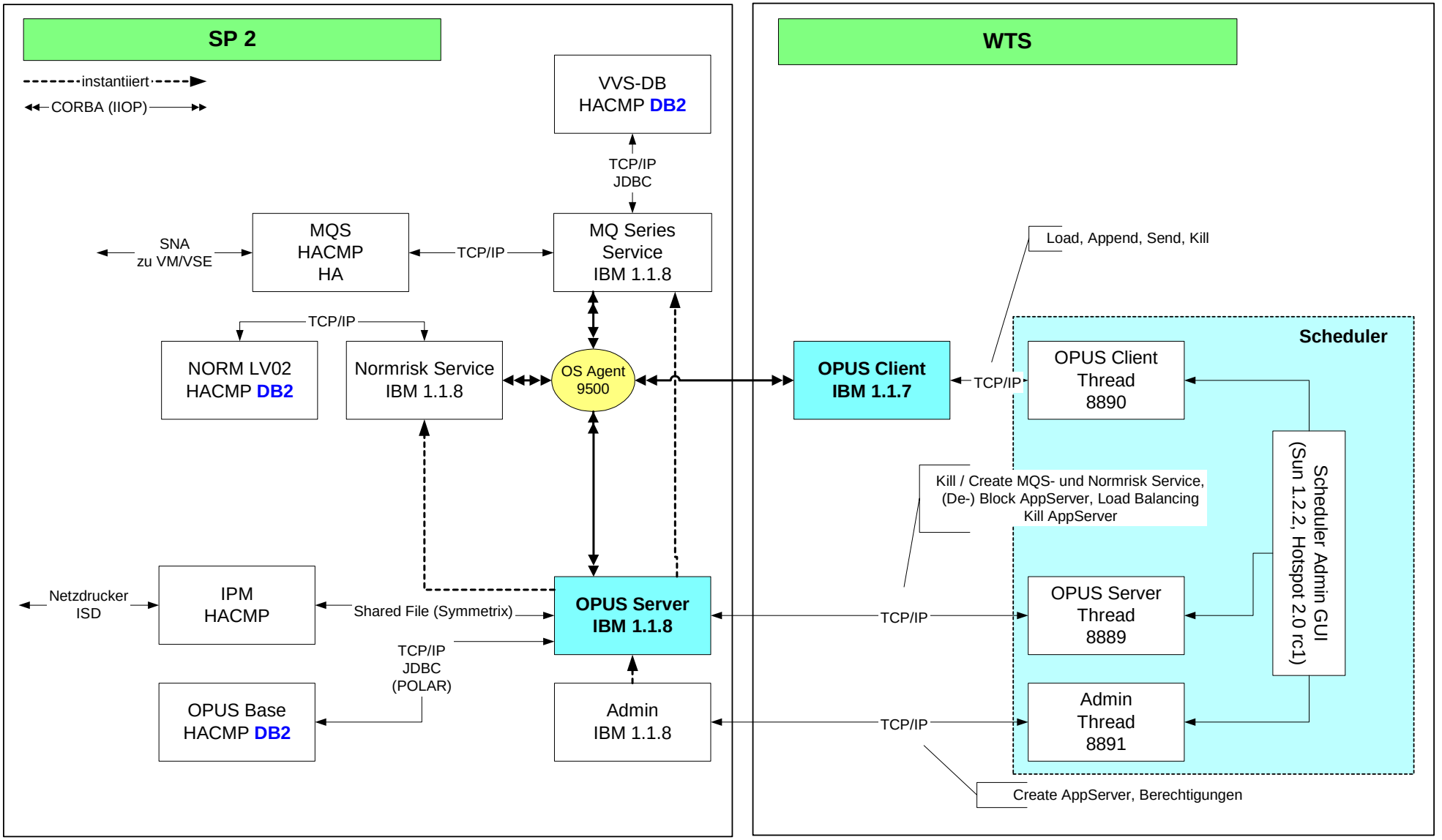
- UCSuchen
- UCÜbersicht
- UCVertragsauskunft
- UCVorgangshistorie
- UCFormulare
- UCAllgemeineBriefe
- Workflow

Java - Produktionsumgebung

Verteilung und Kommunikation

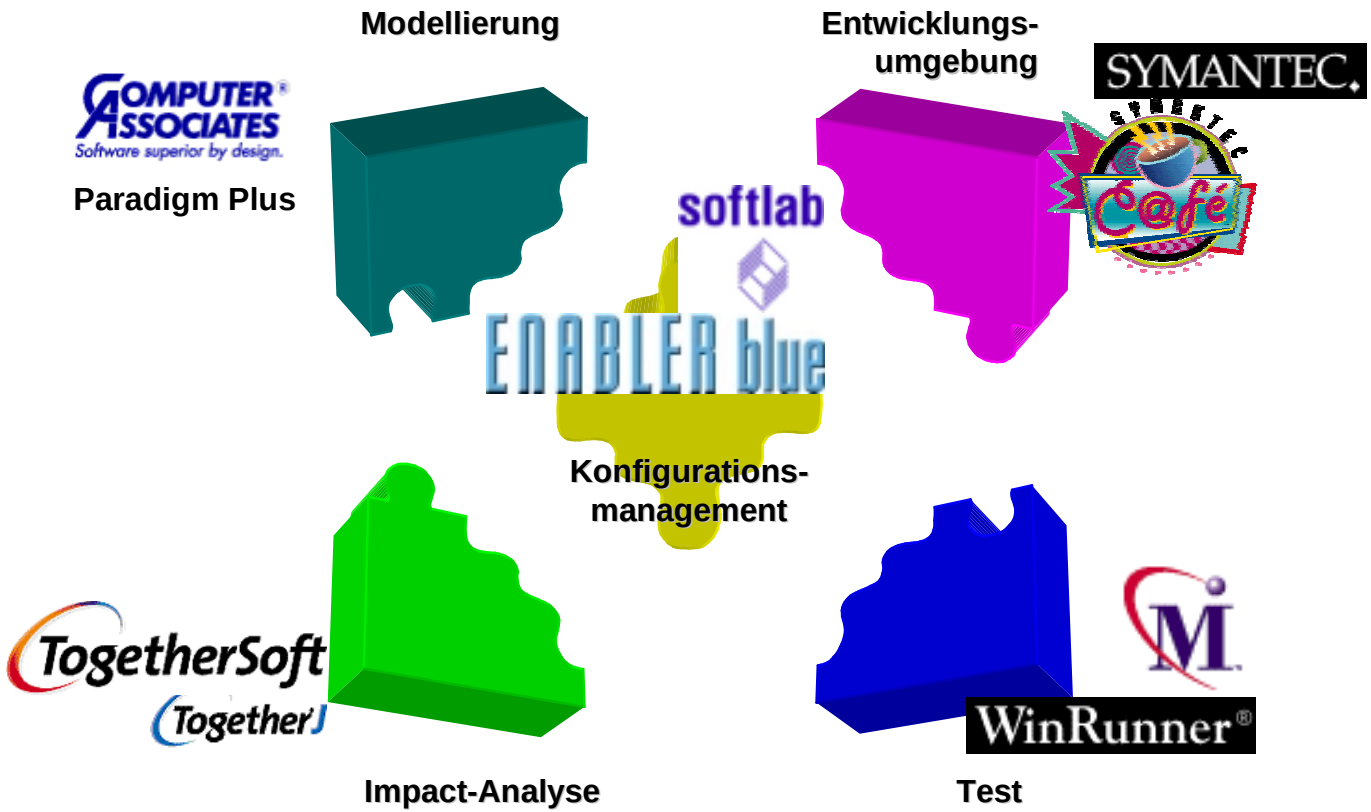


Softwarearchitektur



Werkzeugverbund in der Entwicklung

Werkzeugverbund in der Entwicklung



Software Konfigurationsmanagement

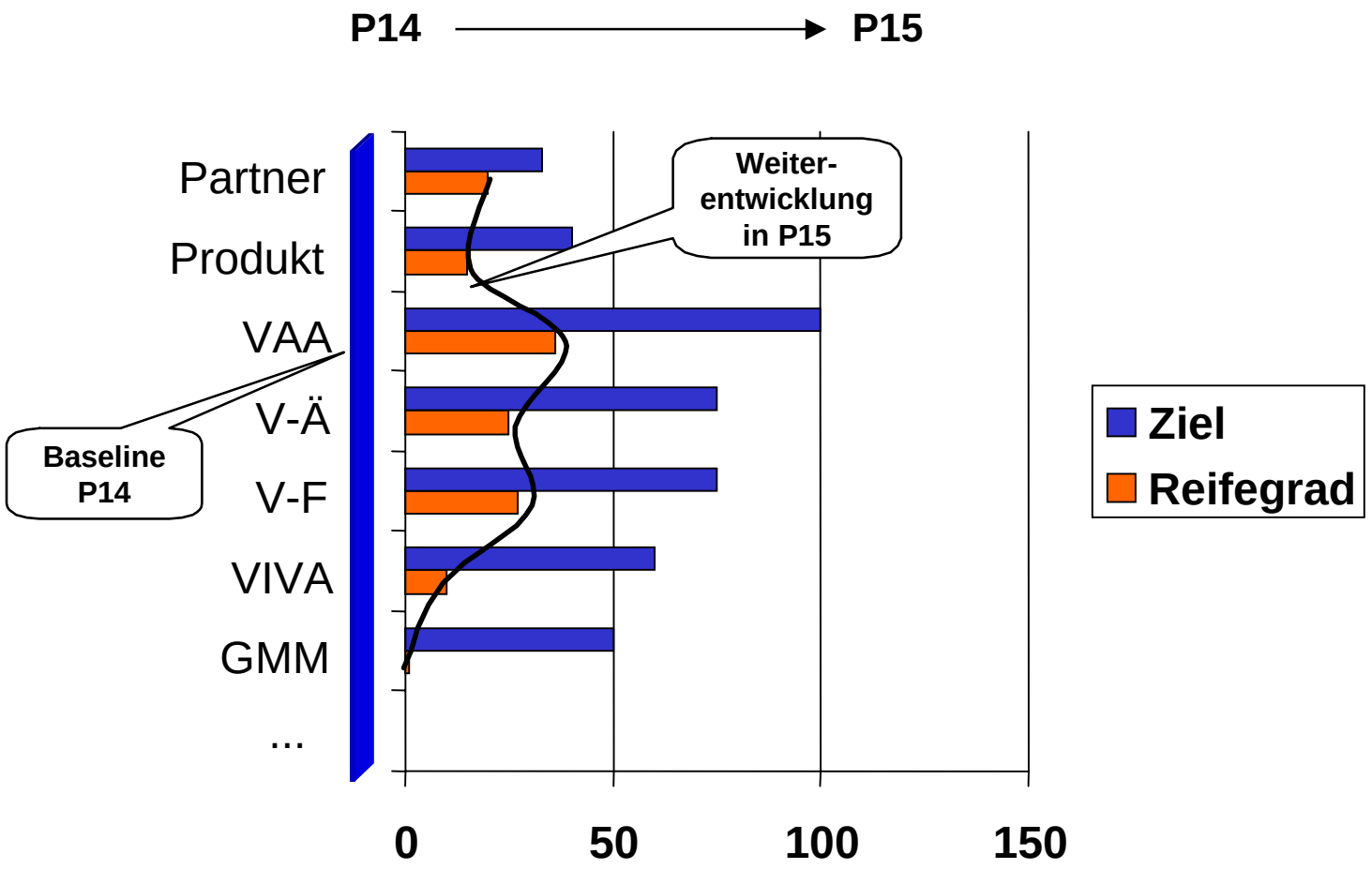
Beherrschen der Komplexität
großer Software Entwicklungsprojekte

Mengengerüst

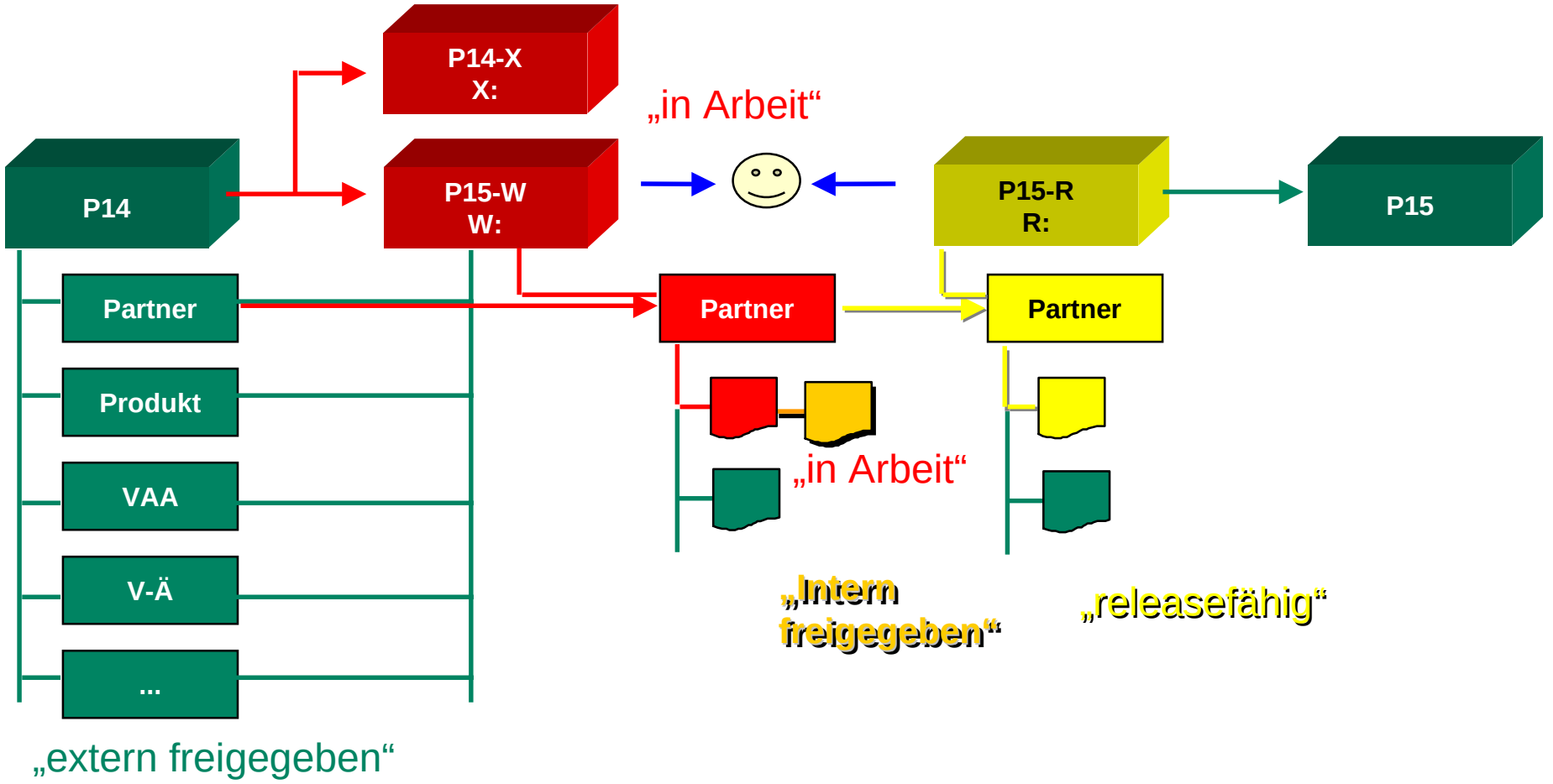
- **50 Entwickler**
- **ca. 17.000 Java Klassen**
- **Projektlaufzeit bis heute: 3 Jahre**

„Seit der Einführung von professionellem Konfigurations-management mit Enabler Blue konnten Leerlaufzeiten durch Release-Bildung, Test und Fehlerbehebung um mehr als 60 % reduziert und die Qualität deutlich verbessert werden.“

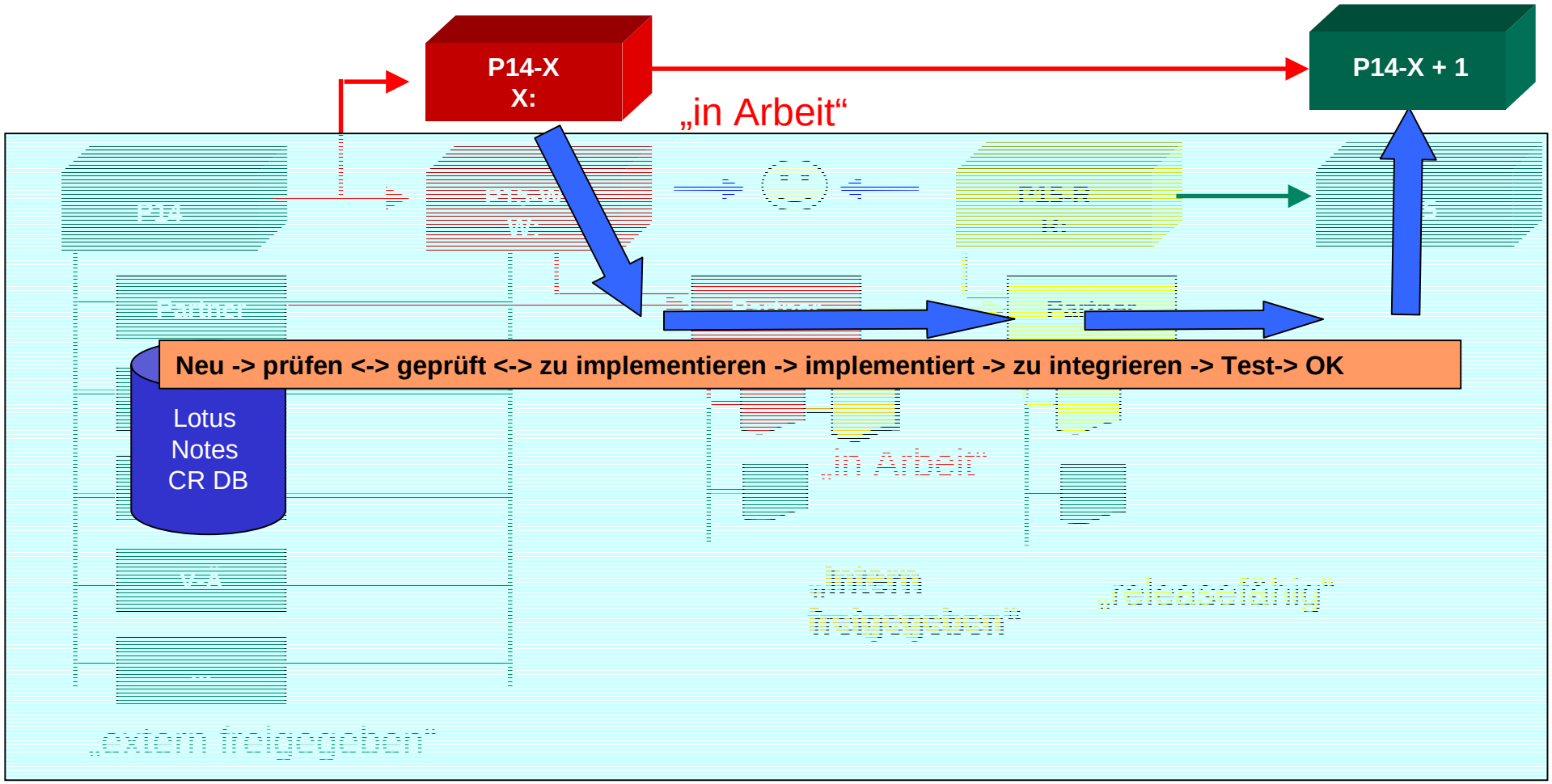
Baselining und Weiterentwicklung



Weiterentwicklung und Releasebildung



Bugfixing und Releasebildung



Releasebildung und Test

Neu -> prüfen <-> geprüft <-> zu implementieren -> implementiert -> zu integrieren -> Test-> OK

Lotus Notes
Change Request Board
Fachbereich
Entwicklung

Entwicklungsumgebung
Entwicklung

Enabler, WinRunner
Konfigurationsmgmt.
Test Management

täglich

- **Buildprozess (weitgehend automatisiert)**
  - **Generierung**
  - **IDL to Java**
  - **Übersetzung**
- **Automatischer Testlauf (WinRunner Scripts)**
  - **Lauffähigkeit**
  - **Stabilität & Korrekheit vs Change Requests**
  - **Logging der Test-Ergebnisse in Enabler**



Kontakt

Siegfried Löhr
Lebensversicherung von 1871 a. G., München
Maximiliansplatz 5
80333 München

(+49)(x89) 55167-290
siegfried.loehr@lv1871.de