



) Architektur Kata)

Asiatische Trainingsmethoden in der Softwareentwicklung

Architektur-Kata: 1-800-AMI-SICK

Patrick Rudloff
Falk Sippach

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Java und XML

) Software Factory)

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

) Object Rangers)

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

功夫

Etwas durch harte/geduldige Arbeit Erreichtes

形

...eine Übungsform, die aus stilisierten Kämpfen besteht...

建築形

...eine Übungsform, die aus stilisierten Architekturen besteht...
oder so..

道場

...Dojo, der Ort an dem ein Kata stattfindet...

Ablauf (19 – 20 Uhr)

- Themenvorstellung (5 Minuten)
- Gruppenbildung
- 1. Runde „Architektur“ erstellen (30 Minuten)
- Peer Review (10 Minuten)
- Pause (15 Minuten)

- Theorie Architektur-Dokumentation (15 Minuten) - ***optional***
- 2. Runde „Architektur“ anpassen (20 Minuten)
- Public Review (20 Minuten)
- Feedbackrunde (5 Minuten)
- Get-Together (Open End)

- fiktives Beispiel
- kurze Beschreibung, nicht bis ins kleinste Detail spezifiziert
- Annahmen treffen
 - **entweder** weglassen, was nicht in der Aufgabe steht
 - **oder** explizit Annahmen treffen, wenn es sinnvoll wäre
- Product Owner fragen
 - Wer ist die Zielgruppe für unseren Entwurf?
 - Anzahl Benutzer?
 - Art der Nutzung?
 - ...



Orientation in Objects



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

) 0800 – BIN ICH KRANK)

Sie sind Architekt in einem Unternehmen das Call-Center Software entwickelt. Ihr Unternehmen möchte ein Softwaresystem entwickeln, dass Patientenberater (Krankenschwestern/Pflegekräfte) dabei unterstützt, auf Anfragen von Patienten zu potentiellen Gesundheitsproblemen zu reagieren.

Dafür soll das Softwaresystem die folgenden Möglichkeiten bieten:

- Zugriff auf die Krankenakte des Patienten
- Unterstützung beim Stellen der medizinischen Diagnose
- Terminvergabe bei einem niedergelassen Arzt
- Verständigen des Notdienstes wenn nötig
- Konferenzschaltung zwischen niedergelassenem Arzt und dem aktuellen Anrufer

Technische Informationen

- 250 Patientenberater (weltweit)
- 10.000+ Kunden

Quelle: <http://archkatas.herokuapp.com/kata.html?kata=AmISick.json>

Zukünftige Entwicklung

Teile des Systems sollen später direkt von den Patienten genutzt werden können.

Jetzt seid ihr dran ...

„Architektur“ erstellen

Und jetzt stellen wir uns die Ergebnisse vor ...

Peer Review

) PAUSE)

„Architektur“ verbessern

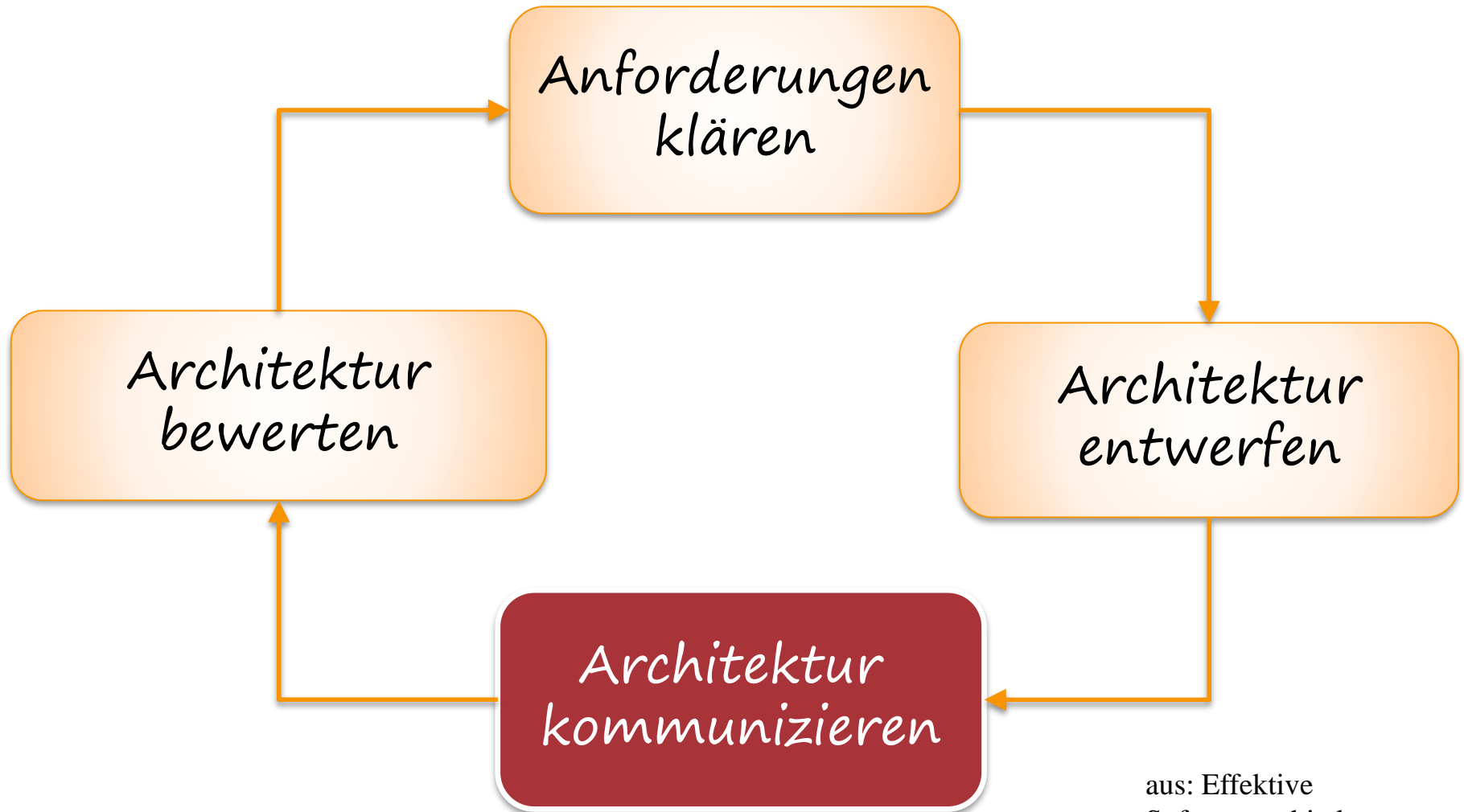
- Randbedingungen herauslesen/erfragen
- Qualitätskriterieren daraus ableiten
- Technologie-Entscheidungen treffen

NUR Vorschläge, KEIN Muss

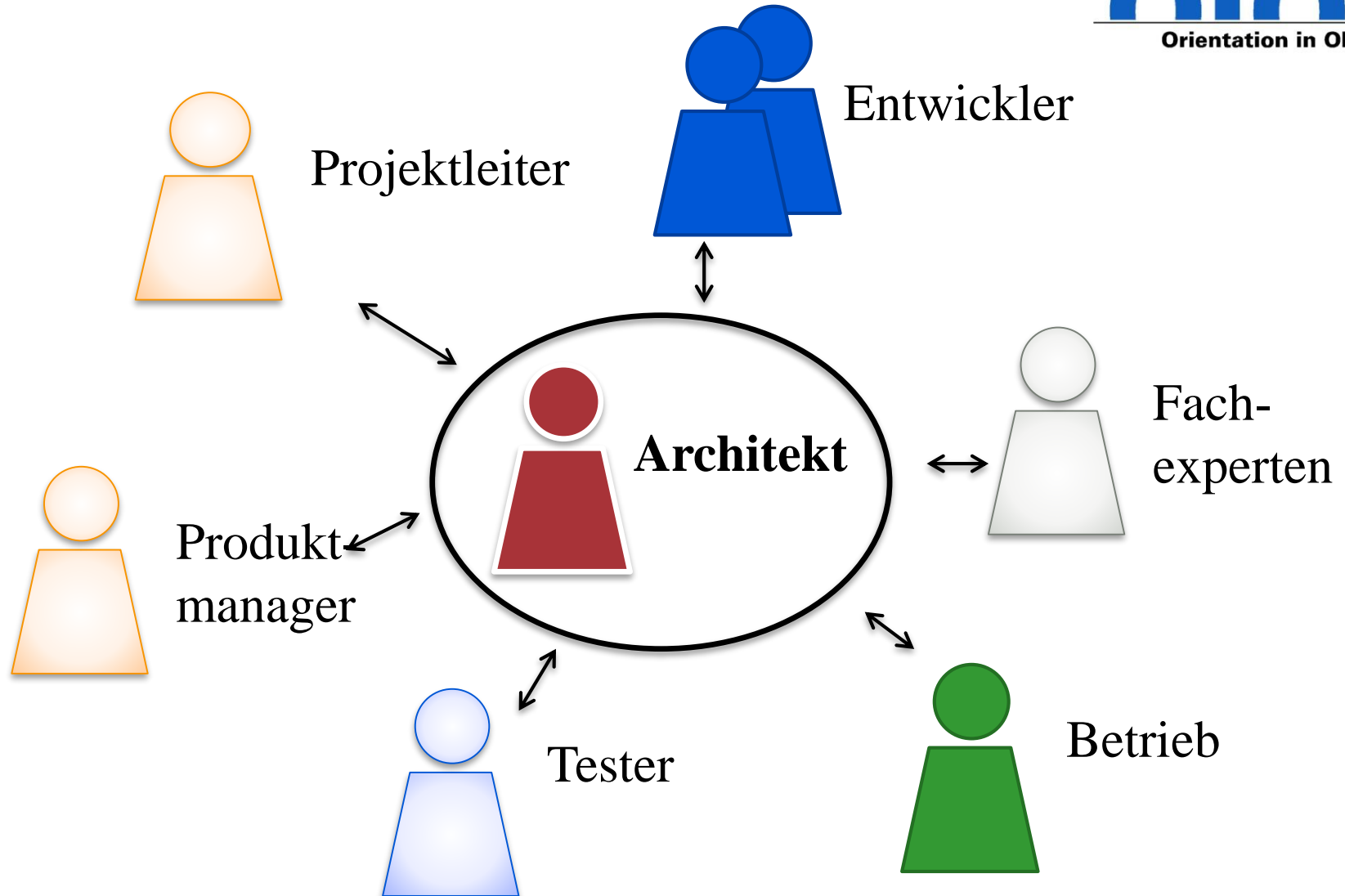
- Anwendergruppen identifizieren
- typische Anwendungsfälle identifizieren

NUR Vorschläge, KEIN Muss

Architektur – Ein Crash Kurs



aus: Effektive
Softwarearchitekturen



Strukturen, Entscheidungen, Konzepte

Begründungen, Vor- und Nachteile

Verteidigen gegen Widerstände

Architektur vermarkten

fundamentale
Strukturen,
Konzepte,
Entscheidungen
und Lösungsansätze

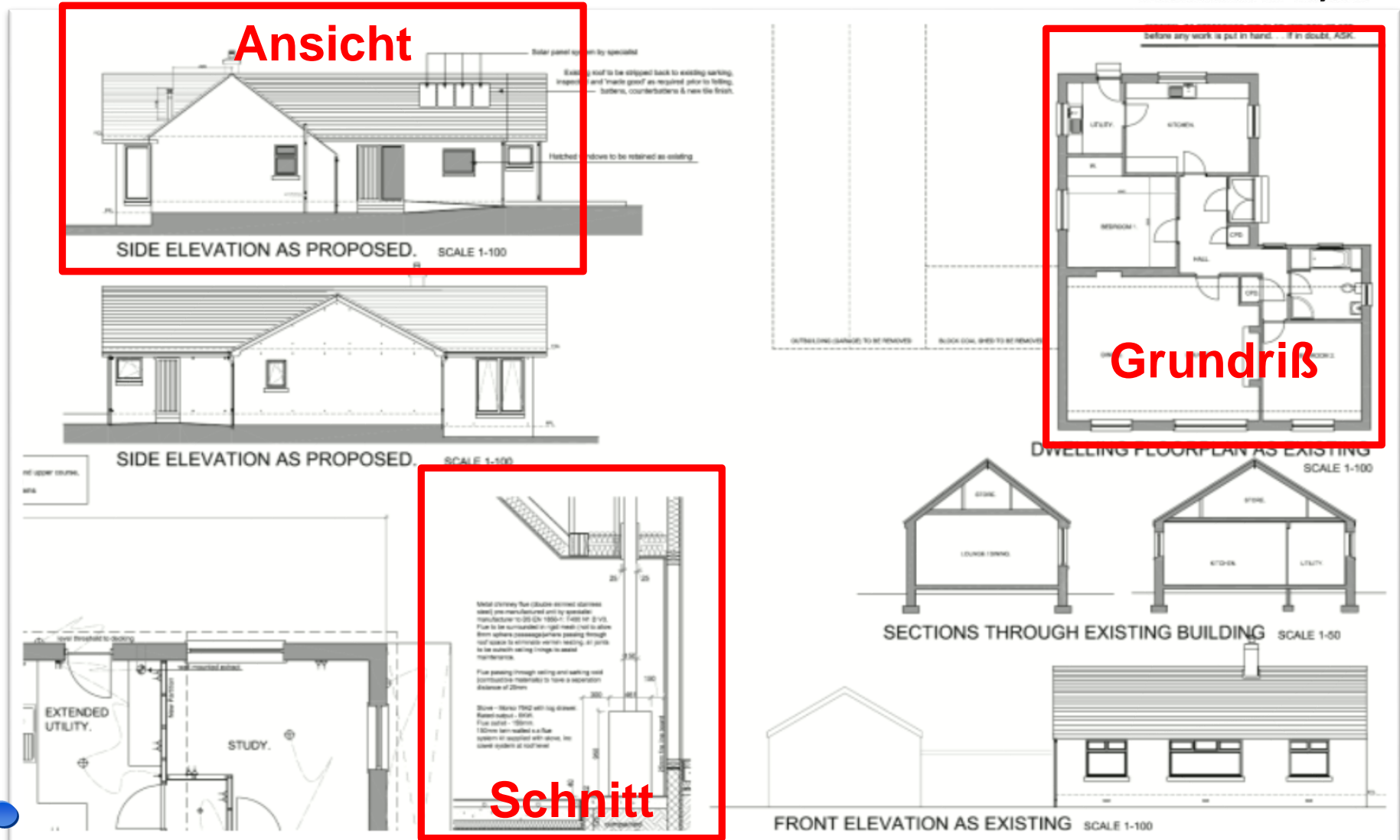
... die **wesentlich** sind, damit Systeme
ihren **Anforderungen** genügen!

Wie sieht Dokumentation nun aus?

Alles eine Frage der Sicht!

Foto von [Thomas Leth-Olsen](#), available under a [CC-BY-SA](#) license.

Sichten in anderen Domänen



Pläne sind Modelle (Abstraktionen)

vernachlässigen gewisse Details (Komplexität)

jede Zielgruppe erhält zugeschnittene Pläne

= **Sichten** (spezifische Perspektive auf ein System)

z.B. Grundriß, Elektroplan,
Statik, Sanitär, 3D-Modell ...

Architektur ist vielschichtig und komplex ...

... eine einzelne Darstellung reicht nicht aus.

Konzentration auf einzelne Aspekte ...

... reduziert Darstellungskomplexität.

Unterschiedliche Informationsbedürfnisse ...

... von den verschiedenen Projektbeteiligten.

Sichten in der Softwarearchitektur

Kontextsichten

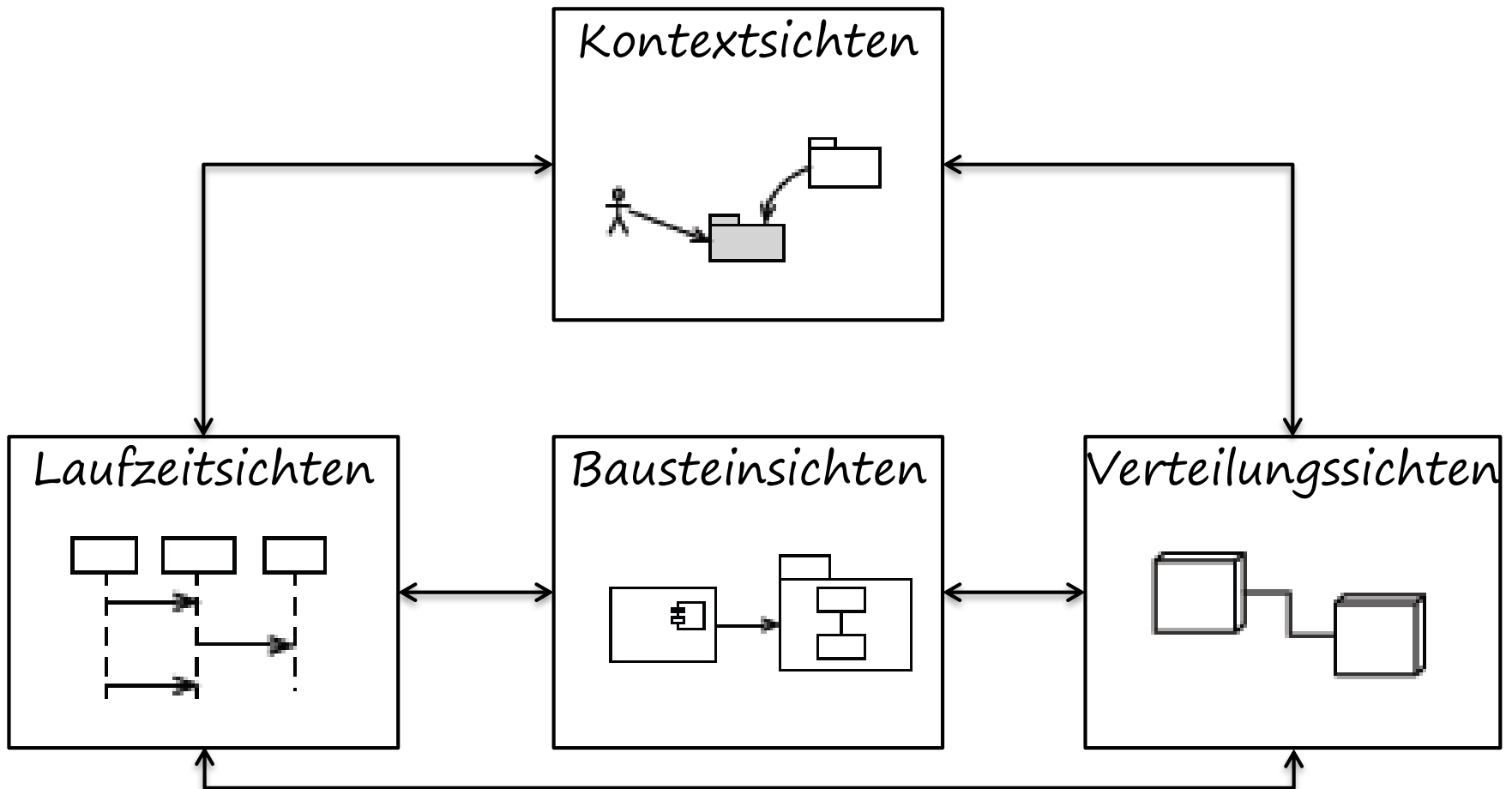
Bausteinsichten

Laufzeitsichten

Verteilungssichten



Foto von Petr Dosek, available under a [CC-BY-SA](#) license.



aus: Effektive
Softwarearchitekturen

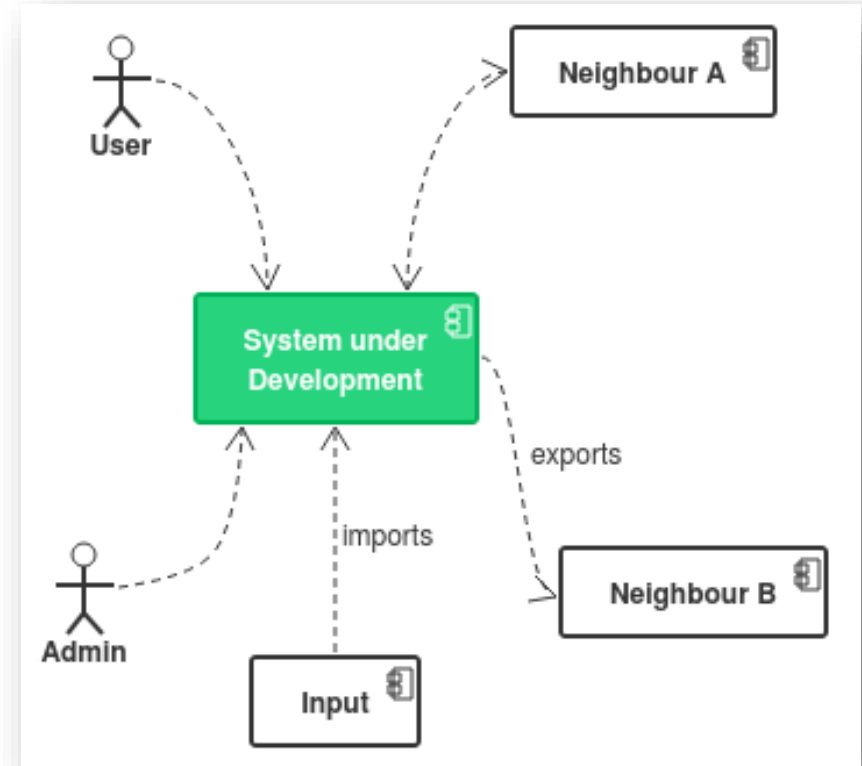
= **Kontextabgrenzung/Kontextsicht**

konzeptionelle Übersicht, hohe Abstraktionsebene

Foto von Marufish, available under a [CC-BY-SA](#) license.

Top-Level-Überblick,
sehr **hohe Abstraktionsebene**

zeigt
System als Blackbox,
wichtige Anwendungsfälle,
externe **Schnittstellen**,
Nachbarsysteme,
und Typen von **Benutzern**





Das wird teuer ...

Zusammenwirken mit **externen**
Systeme birgt immer **Risiken**

technische und organisatorische
Reibungspunkte

Kontextabgrenzung liefert **Überblick**

Foto von Colin, available under a [CC-BY-SA](#) license.

Hilfsmittel um System kennenzulernen
Erstkontakt für Neulinge

Verständnis für:
Mit wem interagiert das System warum?

gut geeignet für Kommunikation mit:
Kunden, Projektleiter, Entwickler, Betrieb

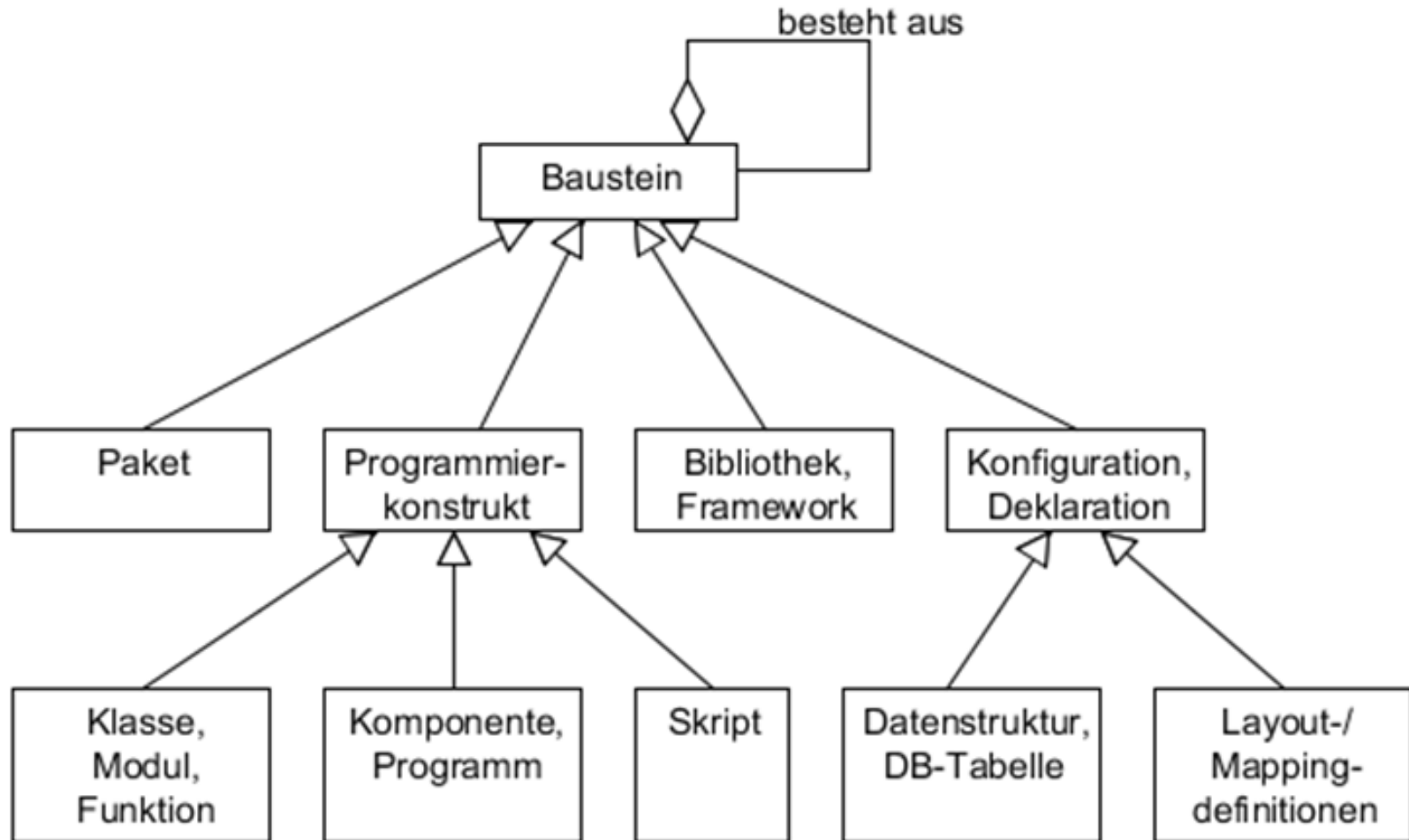
Foto von Aidan Jones, available under a [CC-BY-SA](#) license.



Bausteinsichten

Statische Strukturen des/der
Systems
Subsysteme
Komponenten
Schnittstellen

Ist-Zustand dokumentieren,
nicht den Wunschtraum.



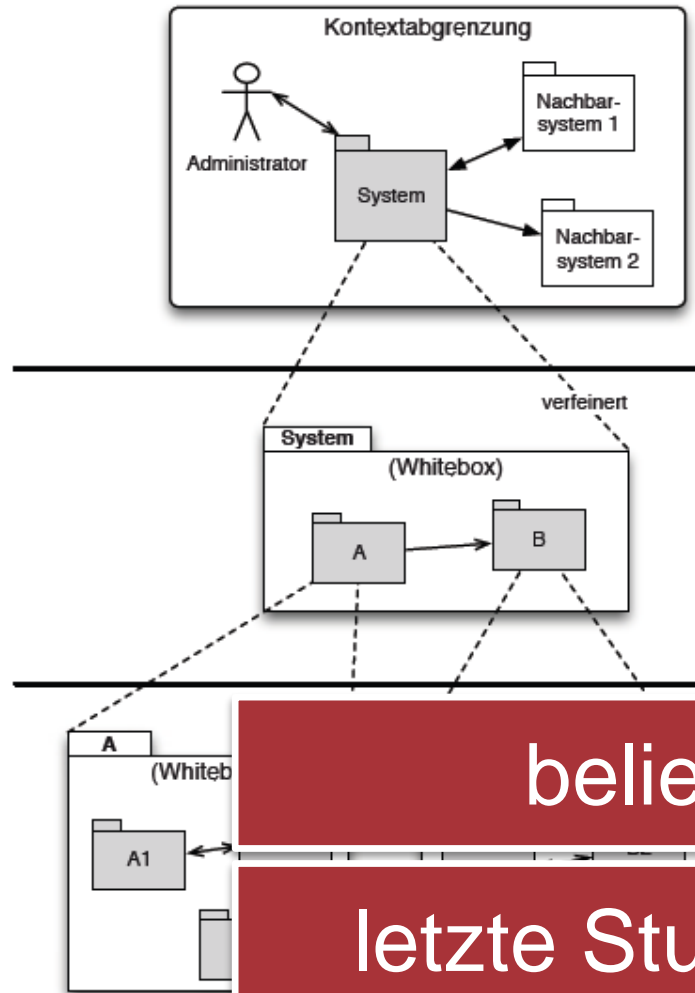
aus: Effektive Softwarearchitekturen

- **Abbildung der Aufgaben** des Systems auf Software-Bausteine/Komponenten
- Verdeutlichung der Struktur und Zusammenhänge **zwischen** den Bausteinen
- zeigen statische Aspekte der Funktionalitäten und nichtfunktionalen Anforderungen

- aus welchen Komponenten, Paketen, Klassen, Subsystemen, Partitionen besteht das System?
- welche Abhängigkeiten bestehen zwischen den Bausteinen?
- welche Bausteine müssen
 - implementiert,
 - konfiguriert oder
 - gekauft werden?

aus: Effektive
Softwarearchitekturen

Level 0
Level 1
Level 2



letzte Stufe ist Quellcode

Elemente typischerweise größer als Javaklassen

Zeigen von logischen Bestandteilen

hoher Pflegeaufwand wenn
Abstraktionsniveau wie Quellcode

Foto von John Lester, available under a [CC-BY-SA](#) license.



Laufzeitsichten

Wie läuft das System ab?

Welche Bausteine existieren zur Laufzeit?

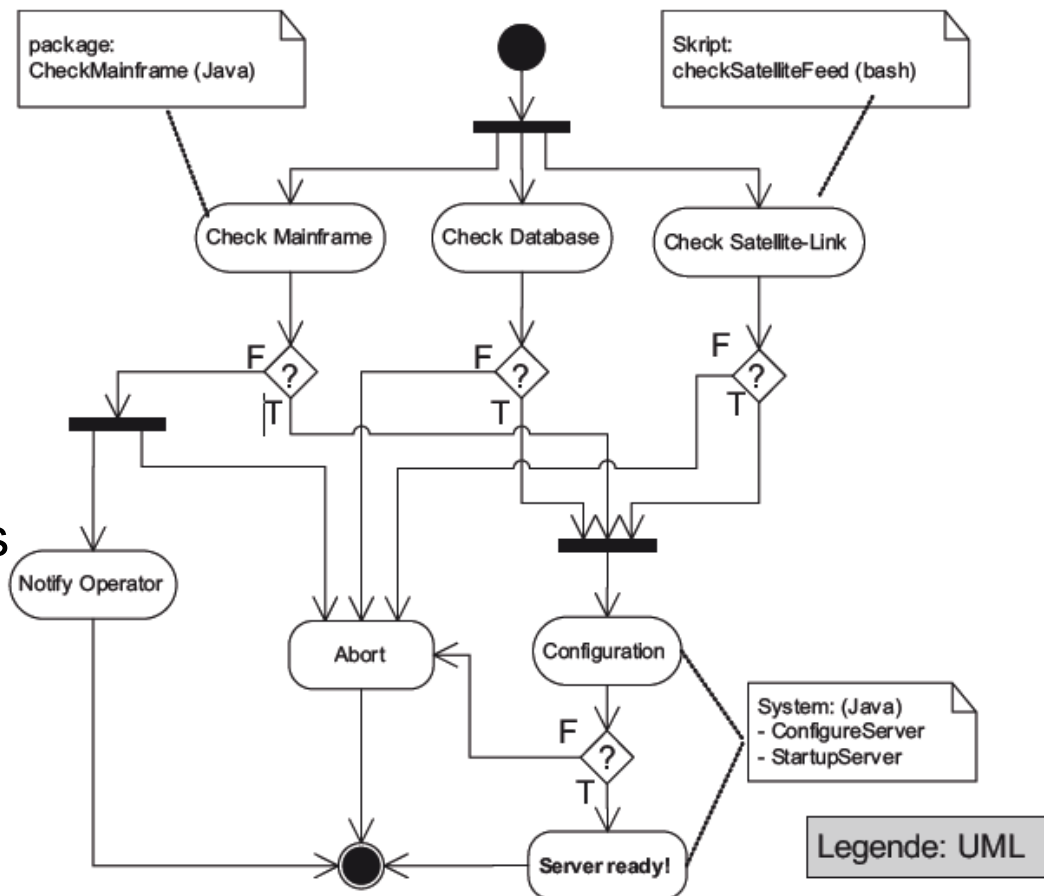
Wie wirken Bausteine zur Laufzeit zusammen?

Wie werden wichtigste Use-Cases bearbeitet?

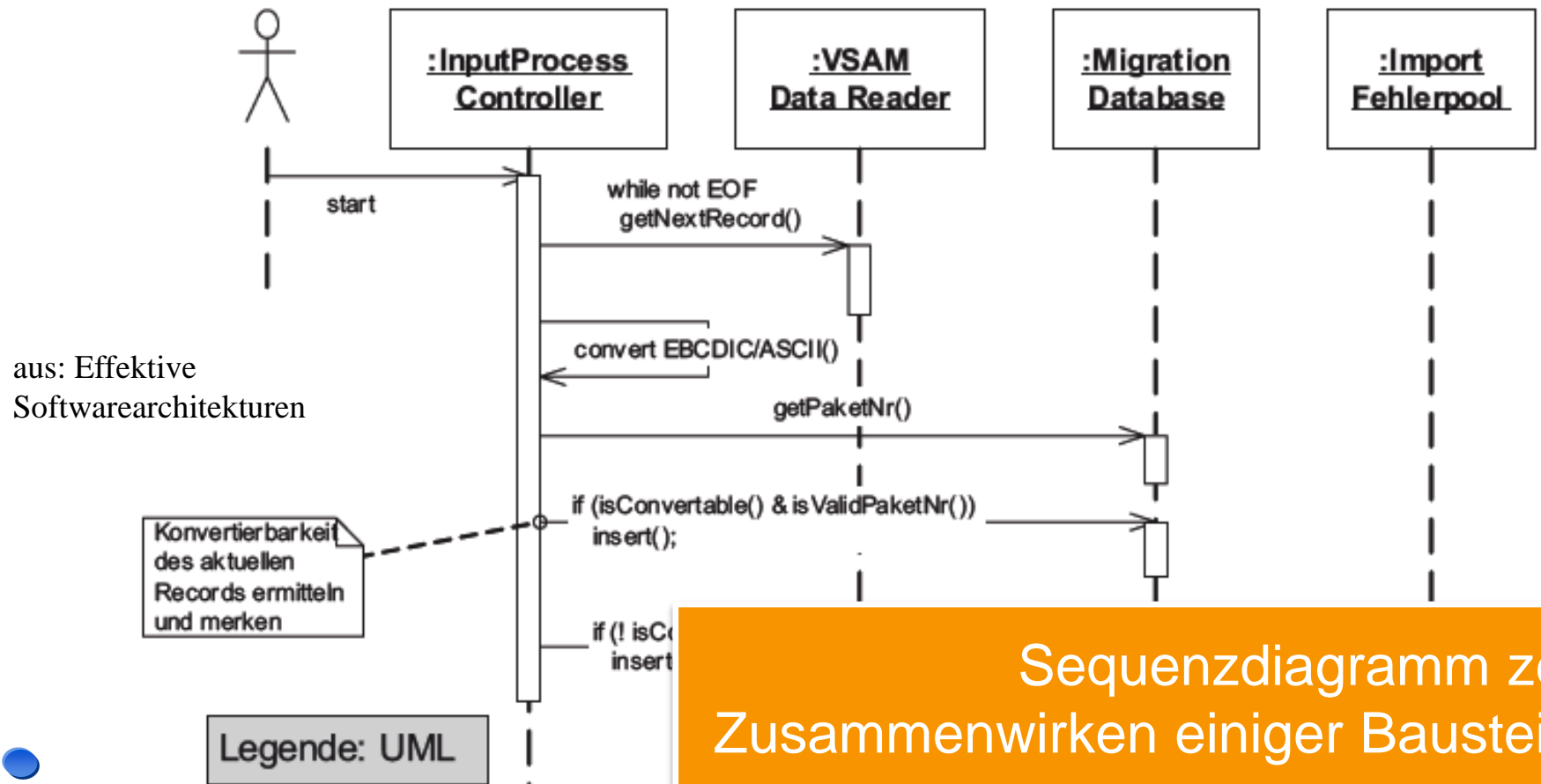
Beschreibung der dynamischen Struktur

Foto von kris krüg, available under a [CC-BY-SA](#) license.

- Systemstart/-ende, Überwachung
 - notwendige Skripte
 - Abhängigkeiten von externen Subsystemen
- Laufzeitkonfiguration
- Administration des Systems
- Zusammenarbeit mit externen und vorhandenen Komponenten



(UML 1.0) (UML 2.0)
Sequenz-, Aktivitäts- und Kollaborations-/Kommunikationsdiagramm.



Sequenzdiagramm zeigt
Zusammenwirken einiger Bausteine.

Verteilungssichten (Infrastruktursichten)

In welcher Umgebung läuft das System ab?

Beschreibung der Hardwarekomponenten

Dokumentation der Rechner, Netztopologien

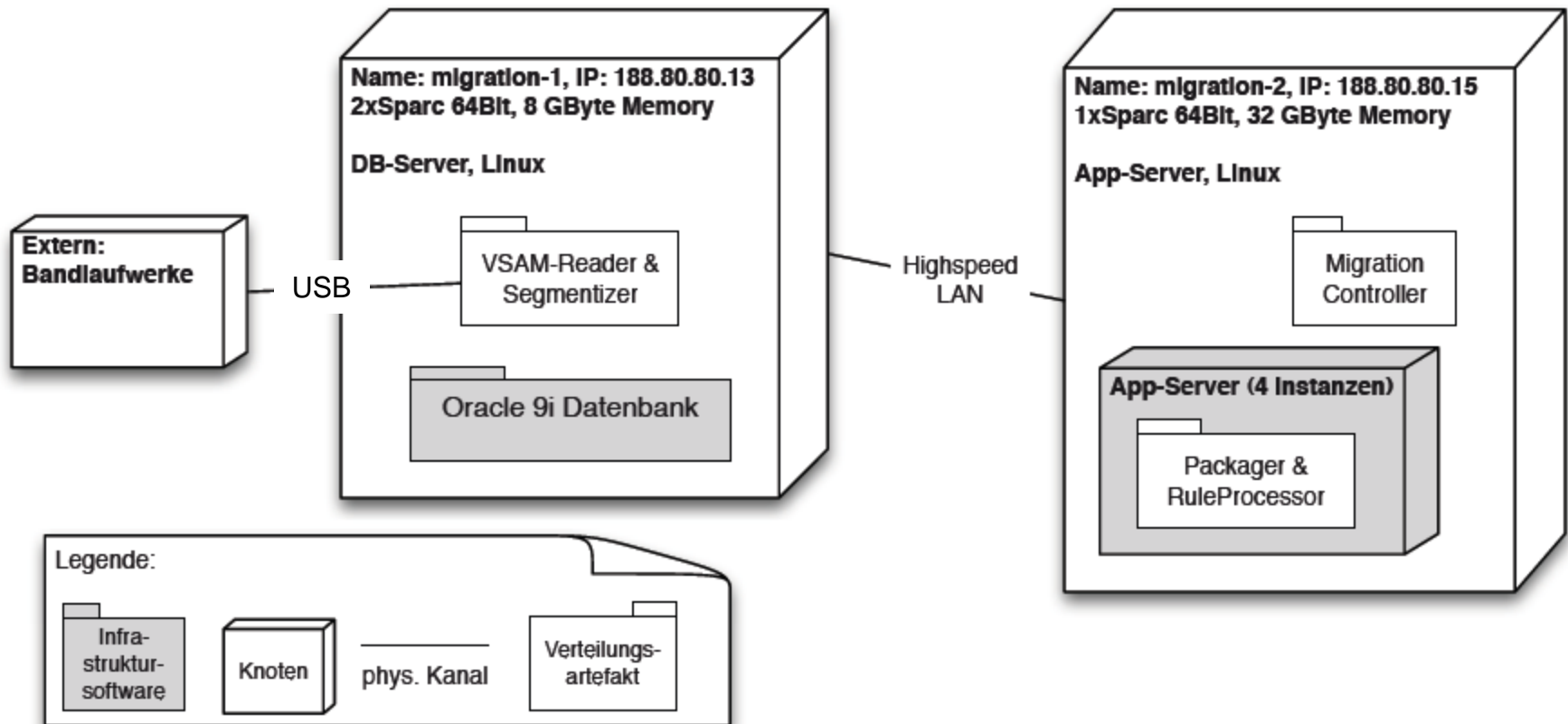
Darstellung der Leistungsdaten, Mengengerüste

Zeigt System aus Betreibersicht

Foto von ACE Solid Waste , available under a [CC-BY](#) license.

- Knoten als Bestandteile der technischen Infrastruktur
 - Rechner oder Prozessoren, sonstige Hardware (Router, Firewall, ...)
- Laufzeitelemente/-artefakte: Instanzen von Bausteinen die auf Knoten ablaufen/deployed werden
- physische Kanäle (zw. Knoten) und logische Kanäle (zwischen Laufzeitelemente)
 - logische Kanäle werden immer über physische realisiert

Beispiel Verteilungssicht



aus: Effektive
Softwarearchitekturen

Ein paar neue / geänderte Anforderungen

- Webanwendung
- Internationaler Marktstart in 6 Monaten
- Regelmäßige Deployments (mind. wöchentlich)
- Anbindung an CallCenter Lösung verschiedener Hersteller
- Auditierung der Patientenberater soll möglich sein
- Einbindung der Gesundheitskarte bei deutschen Kunden

Jetzt seid ihr wieder dran ...

„Architektur“ verbessern

Public Review

Wie war es?

Was habt ihr mitgenommen?

Was kann man besser machen?!



Orientation in Objects



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Vielen Dank für ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de