Komponententechnologie

Ralf Schneeweiß

www.oop-trainer.de

„Lebender“ Netzartikel mit den Themen des Workshops:

http://www.oop-trainer.de/Themen/EmbeddedKomponenten.html

Agenda

1. Technik und Systematisierung
2. Motivation für Komponentenbasierung
3. Logische Sicht
4. Physikalische Sicht
5. Entwicklungssicht

Systemdefinition nach N. Luhmann

- Ein System ist eine Grenzziehung zwischen einer Systeminnenwelt und einer Umwelt.
- Systeme reduzieren Komplexität.
- ...

Technikreife durch Systematisierung

- Technik wird einfach
- Technik wird verstehbar
- Technik wird variabel
- Technik wird standardisierbar

Zwei Definitionen des Begriffs *Komponente* in der Software

① Eine Komponente ist eine in sich abgeschlossene Softwareeinheit mit definierter Schnittstelle, die eine semantisch zusammengehörige Funktionalität besitzt[1)]. Eine Komponente kann in diesem Sinne als Architekturbaustein gesehen werden, der das Ergebnis einer Systemzerlegung auf oberster Ebene ist[2)].

② In der Praxis wird der Komponentenbegriff sehr häufig als unabhängig verteilbare Softwareeinheit im Sinne einer bestehenden Komponententechnologie gebraucht (z.B. J2EE, COM etc.)[2)]. Dieser Komponentenbegriff beschreibt die im Rahmen und mit Hilfe der eingesetzten Technologie gebildeten Softwareeinheiten.

1) Vgl. [Balzert00] S. 856 f.

2) Vgl. [Posch07] S.8

Vorteile Komponentenorientierter Entwicklung

„In der industriellen Produktion ist es ... wirtschaftlich, ein Produkt auf Halbfabrikaten aufzubauen. Die Halbfabrikate werden in der Regel von spezialisierten Herstellern bezogen.

... Eine komponentenbasierte Software-Entwicklung erlaubt eine einfachere, schnellere und preiswertere Herstellung von Anwendungen mit Hilfe von vorgefertigten Halbfabrikaten bzw. Komponenten.“

[Balzert00] S. 256.

Motivation für Komponentenbasierung der Software

- Variabilität durch Kombinierbarkeit
- Verteilbarkeit
- Separate Testbarkeit
- Erleichtert ein vollständiges Fehler- und Ausnahmekonzept
- Erleichtert eine Individualisierung von Softwareprodukten

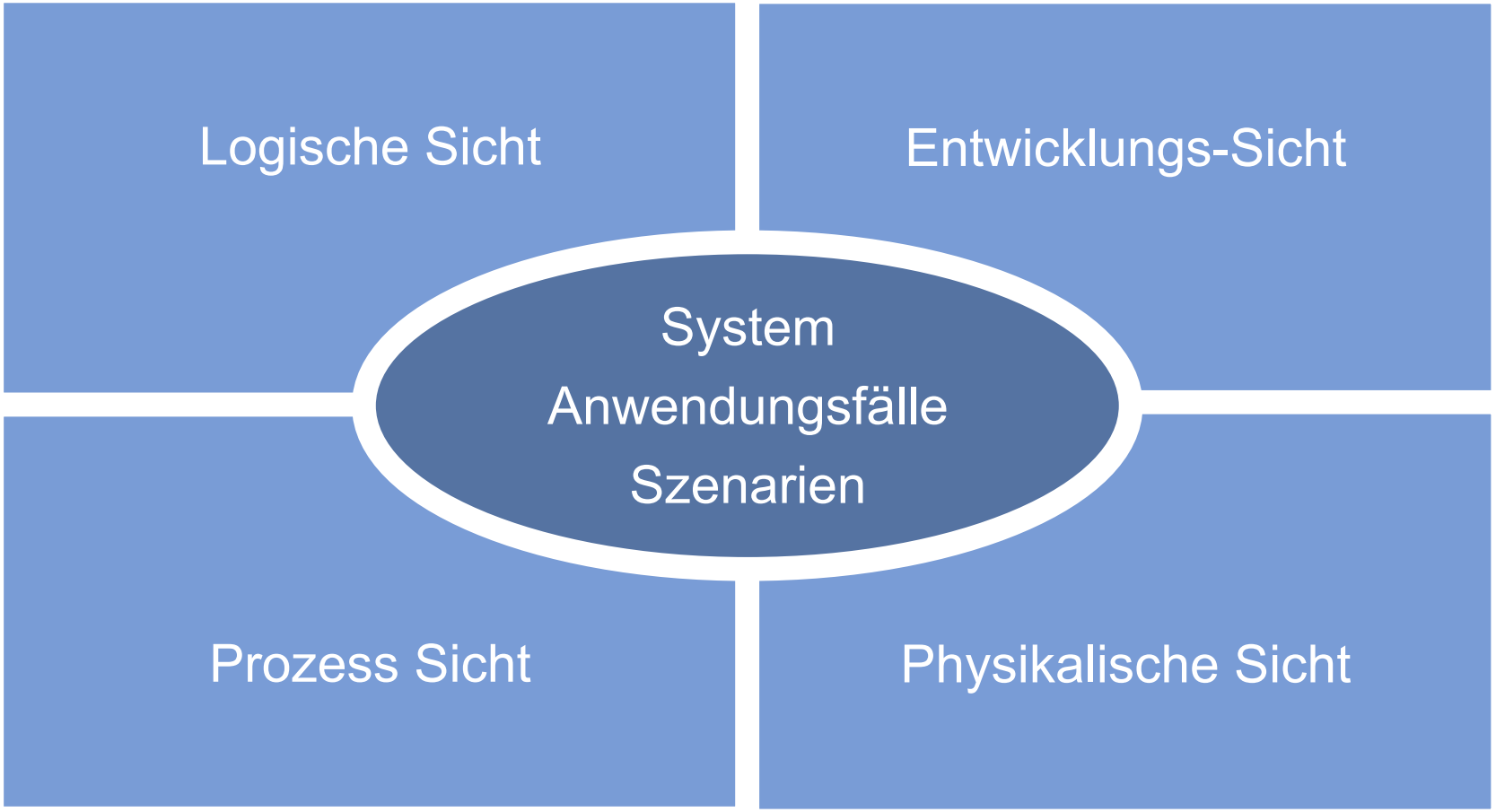
- Wiederverwendung
- Komplexitätsreduktion für den Entwickler
- Komplexitätsreduktion für das Verständnis des gesamten Systems
- Effektivere Arbeitsteilung in der SW-Entwicklung

Was ist ein eingebettetes System?

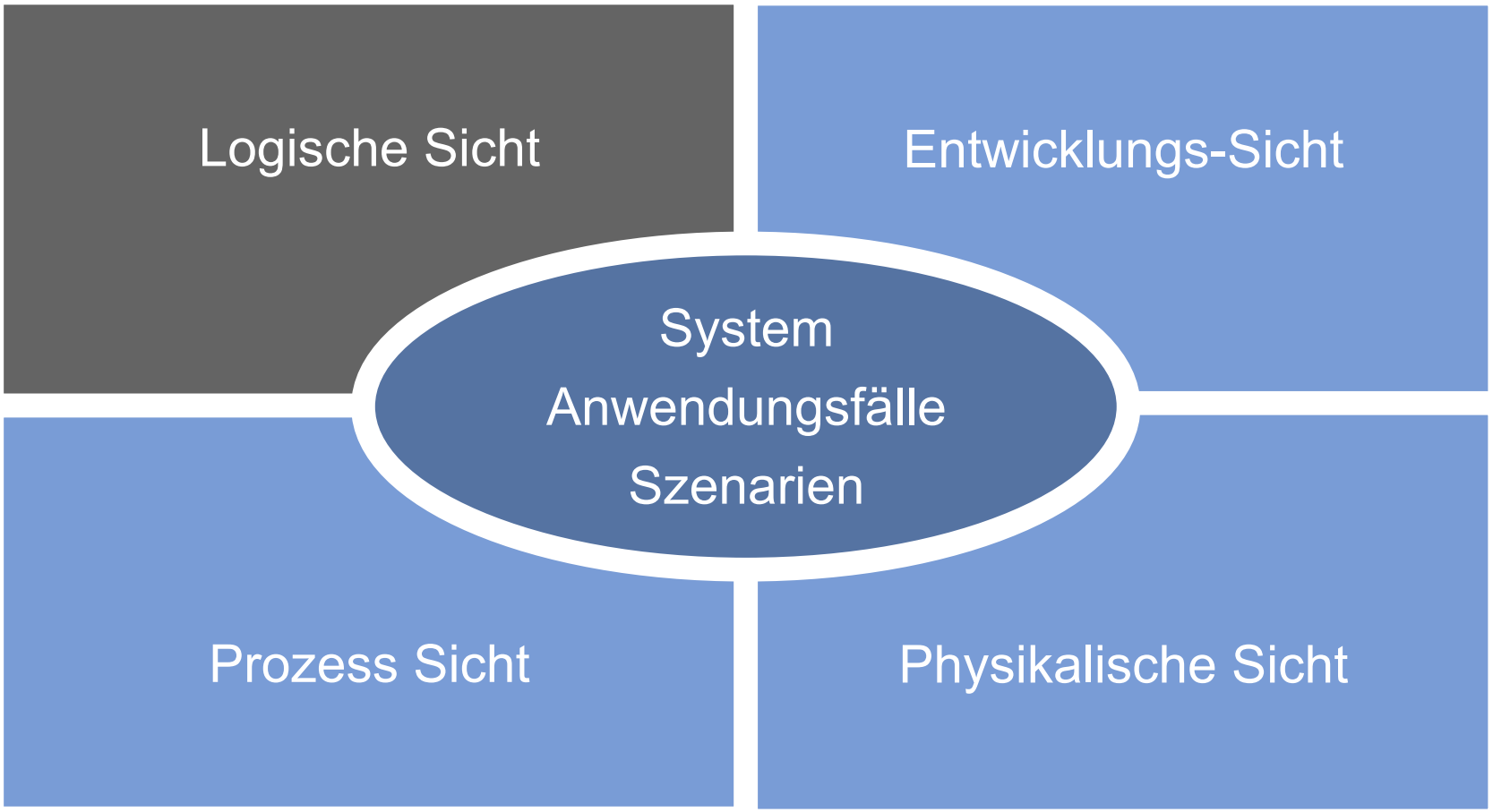
Die technische Anforderung aus der Einbettung eines Systems

Vollständiges Ausnahmekonzept.

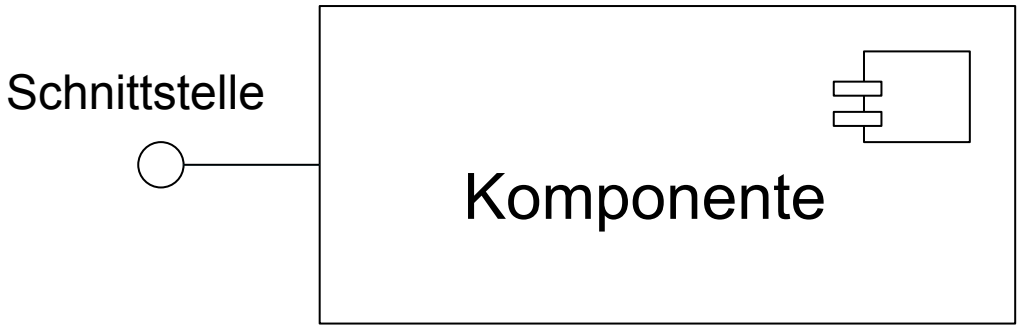
4+1 Architektursichten nach Kruchten



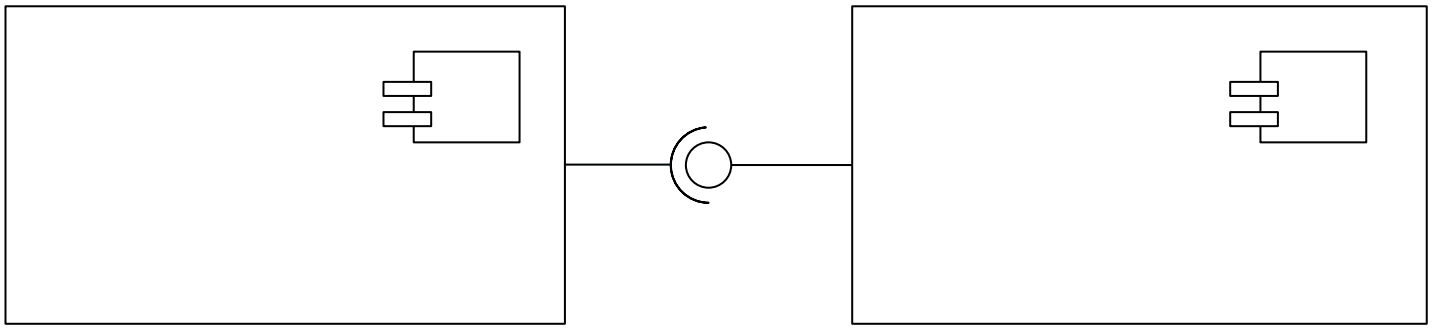
4+1 Architektursichten nach Kruchten



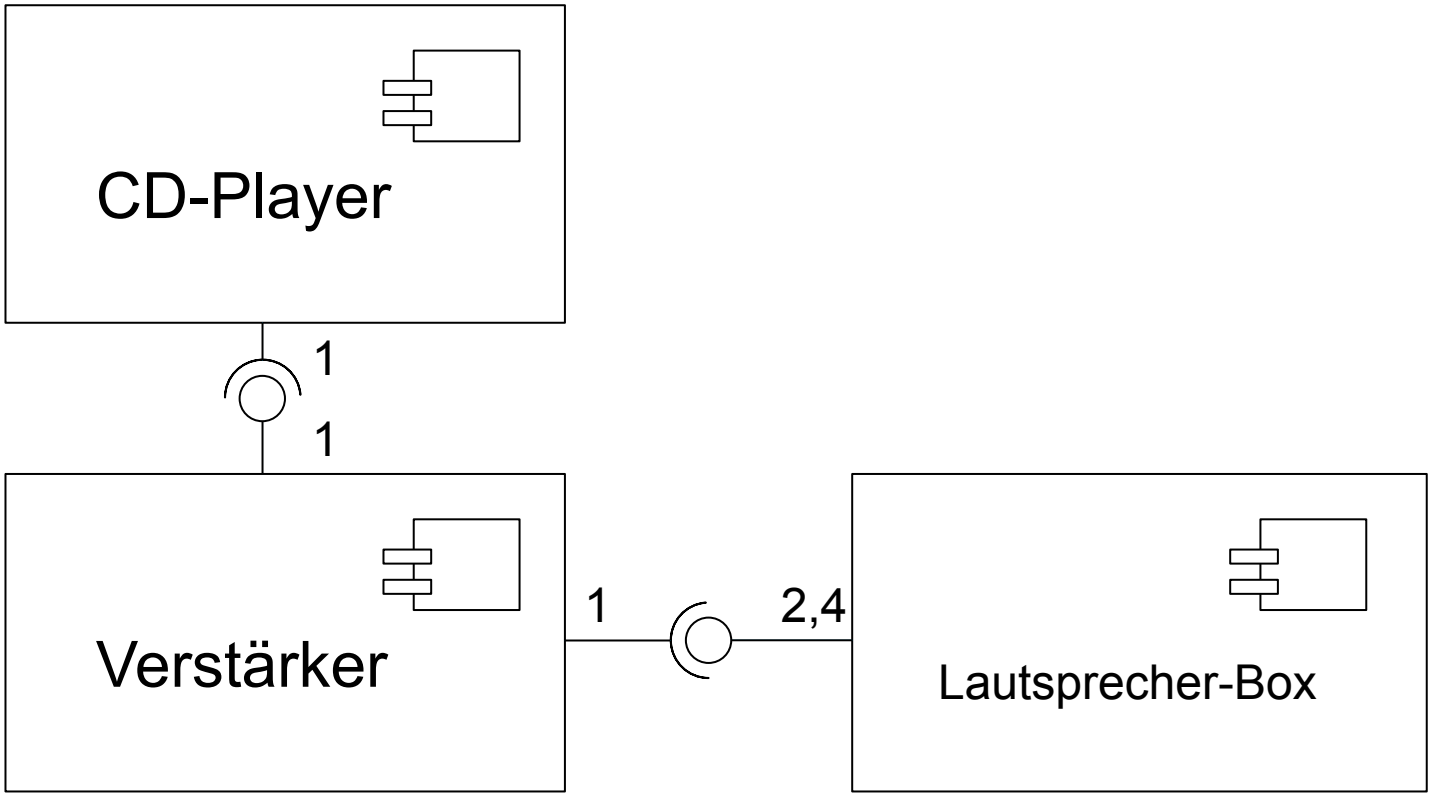
Komponente mit Schnittstelle



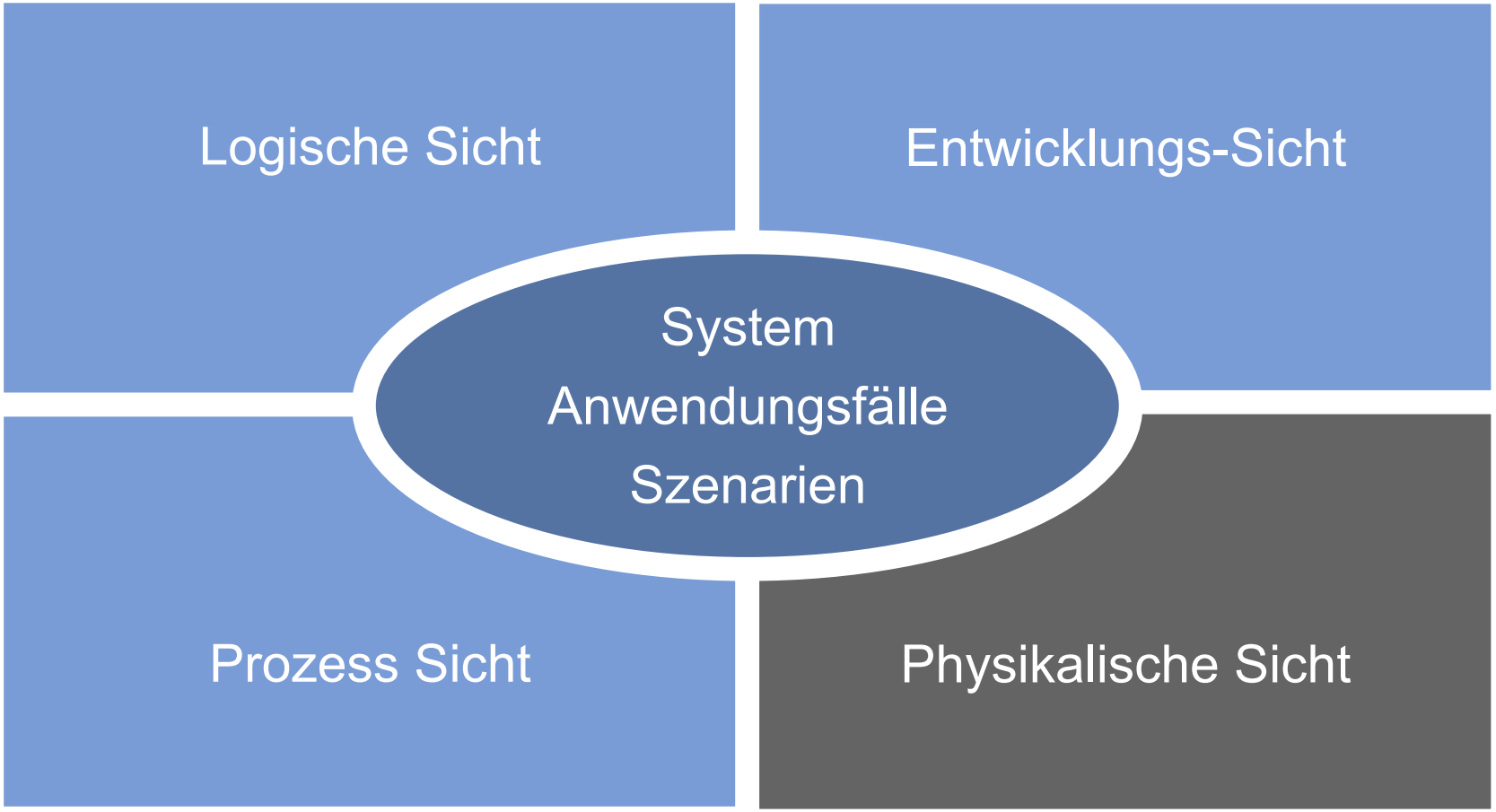
Komponente mit Schnittstelle



Beispiel Stereoanlage

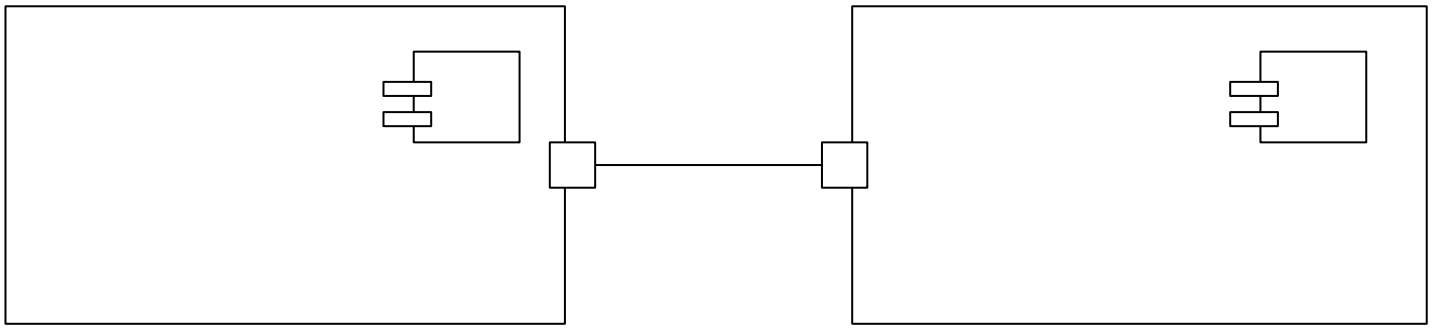


4+1 Architektursichten nach Kruchten

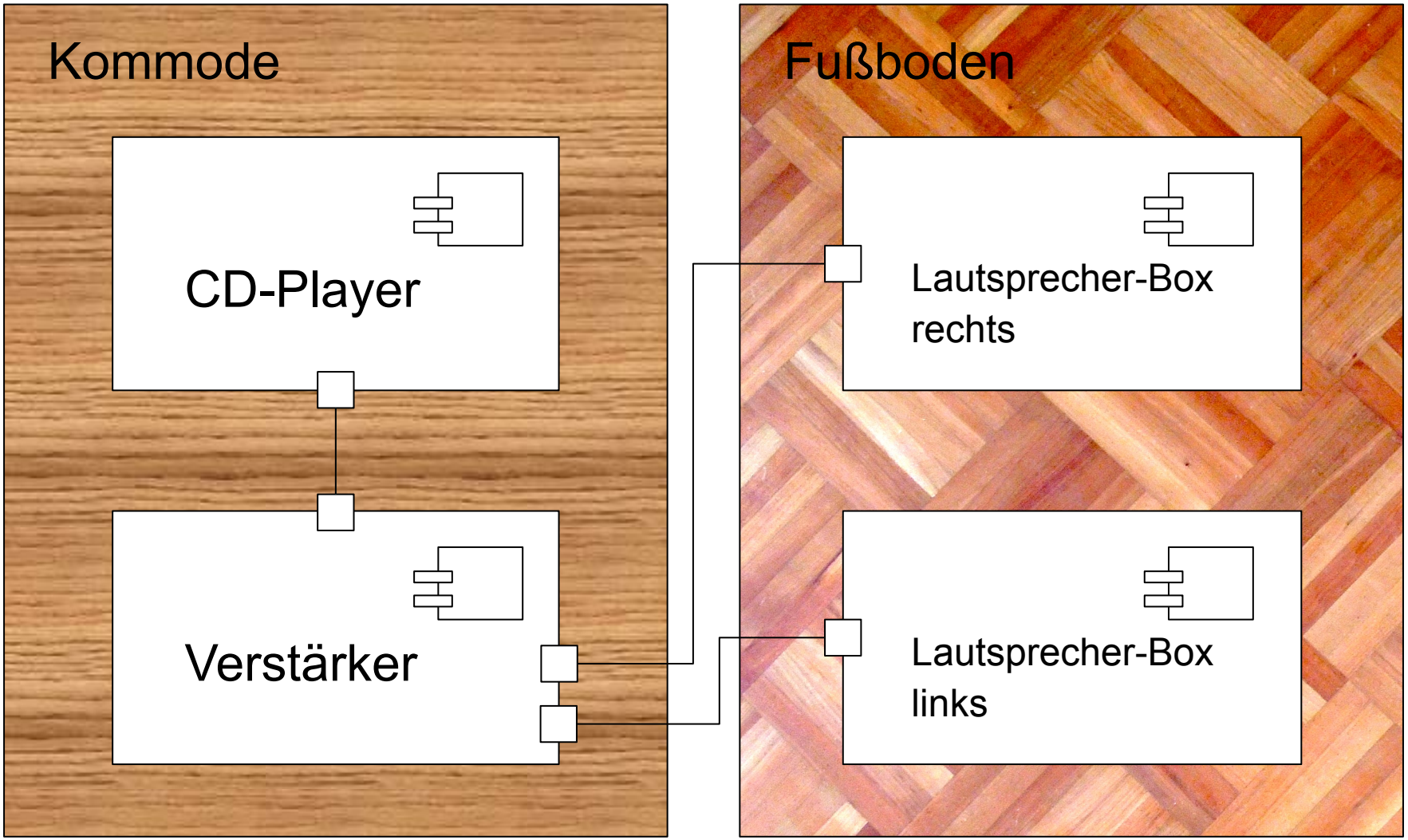


Komponente mit Port

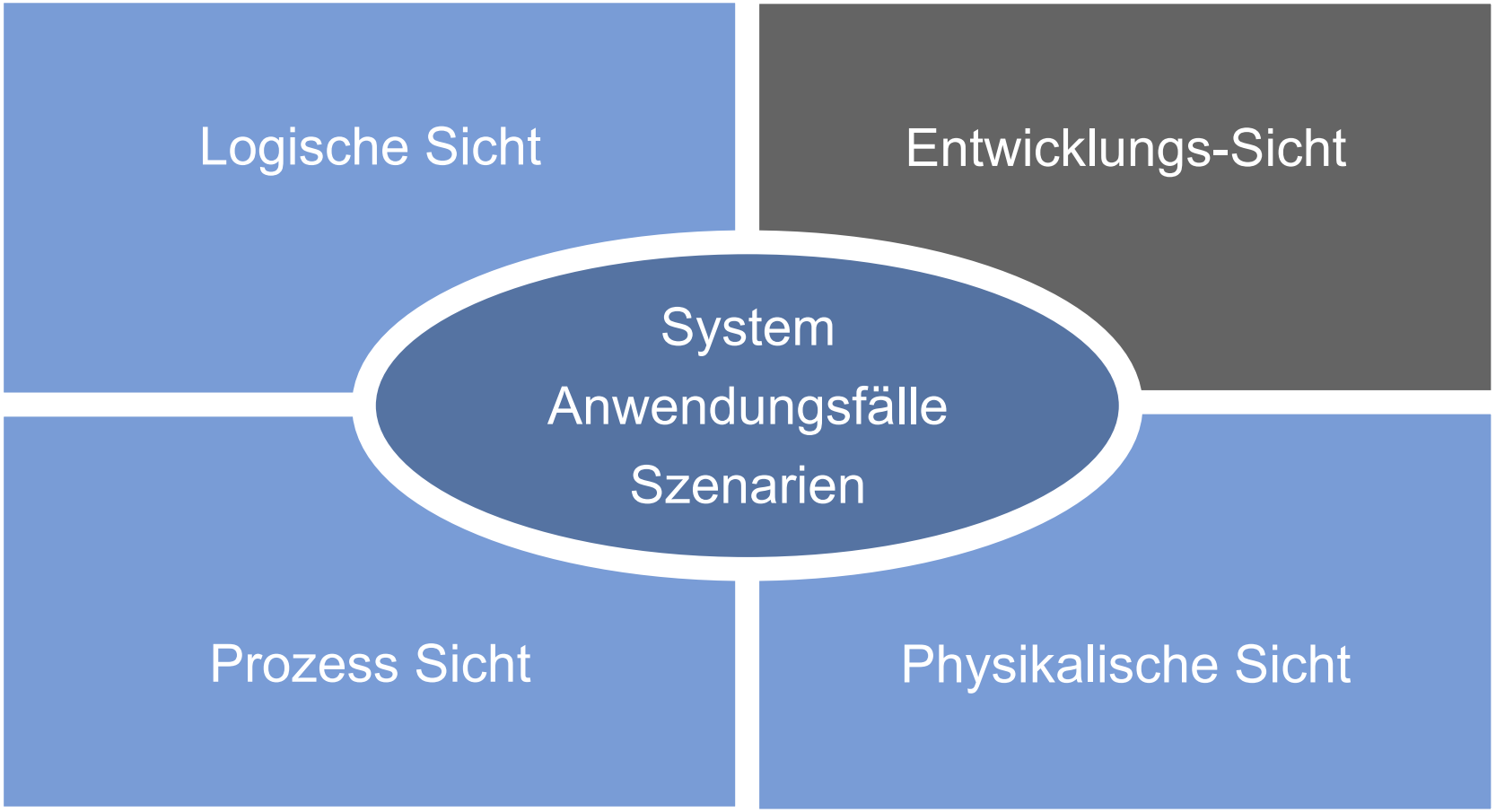
Über Ports verbundene Komponente



Beispiel Stereoanlage



4+1 Architektursichten nach Kruchten



Wichtige Aspekte für die Entwicklung

- Separate Entwicklung der Systembausteine
- Separate Testbarkeit
- Kombinierbarkeit unterschiedlicher Versionen

Technische Aspekte

- Variierbarkeit
- Verteilung
- Kommunikation
- Nebenläufigkeit
- Synchronisierung
- Kombinierbarkeit
- Testbarkeit
- Wartbarkeit
- Erweiterbarkeit zur Laufzeit

Beispielhafte Entscheidungsmatrix zur Komponententechnologie

Ziel	Architekturmuster oder Prinzip
Kombinierbarkeit und Rekombinierbarkeit des Produkts (Produktlinien)	Formale Trennung von Interface und Implementierung der Teile
Testbarkeit eines Einzelteils (Halbfabrikat)	Testumgebung (Sandbox)
Verteilung und Kommunikation	Proxy
Erweiterbarkeit der Software zur Laufzeit	Introspektion, Reflektion
Nebenläufigkeit, Synchronisierung	Aspekttrennung zwischen fachlichem Code und technischer Synchronisierung

Einige Komponententechnologien

- Java
- J2EE
- .NET
- CORBA
- OSGi

Literatur

[Balzert00] H. Balzert: Lehrbuch der Software-Technik. Spektrum Akademischer Verlag, 2000.

[Posch07] T Posch, K. Birken, M. Gerdom: Basiswissen Softwarearchitektur. dpunkt.verlag, 2. Auflage 2007.