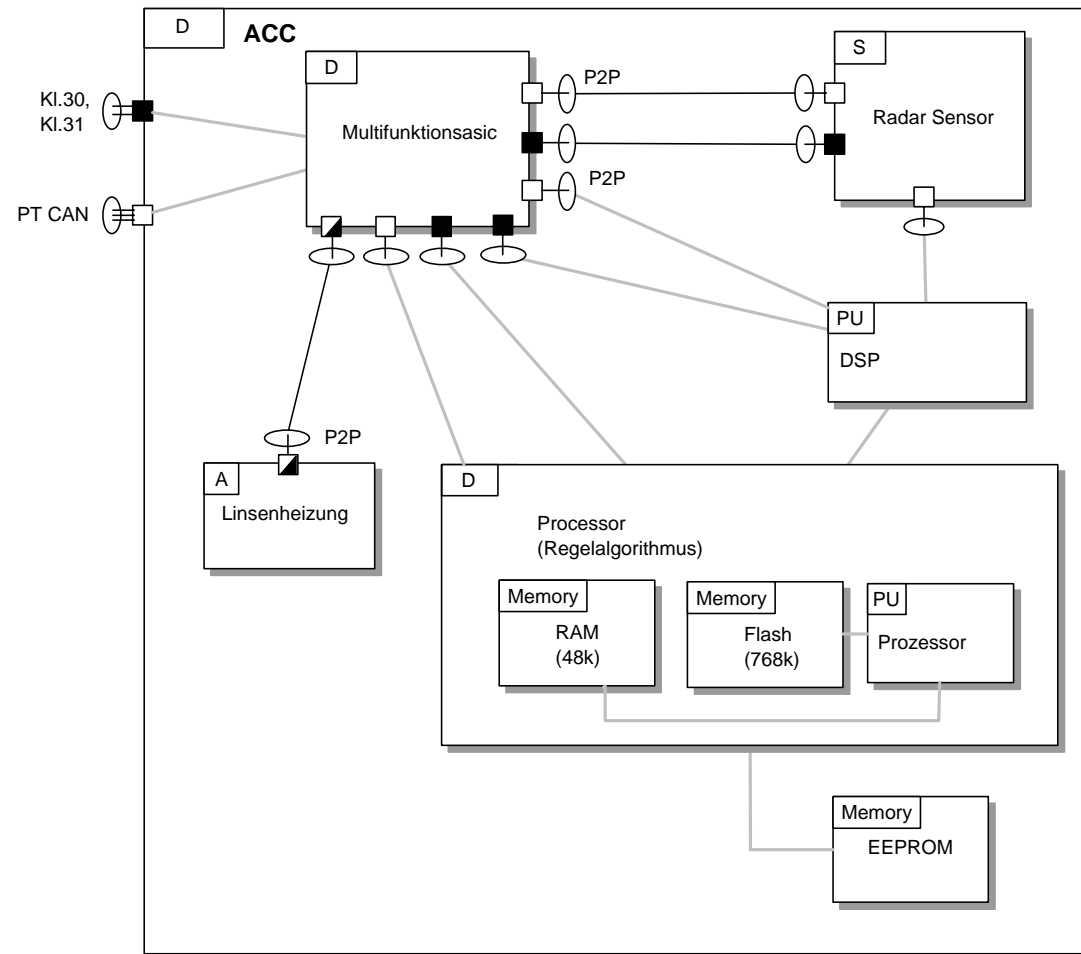


Architekturen



BMW Group



Überblick

Entwicklungsobjekt

Funktionsnetze

Logische Architektur

Implementierungsarchitektur

Deployment

Definition einer Systemarchitektur

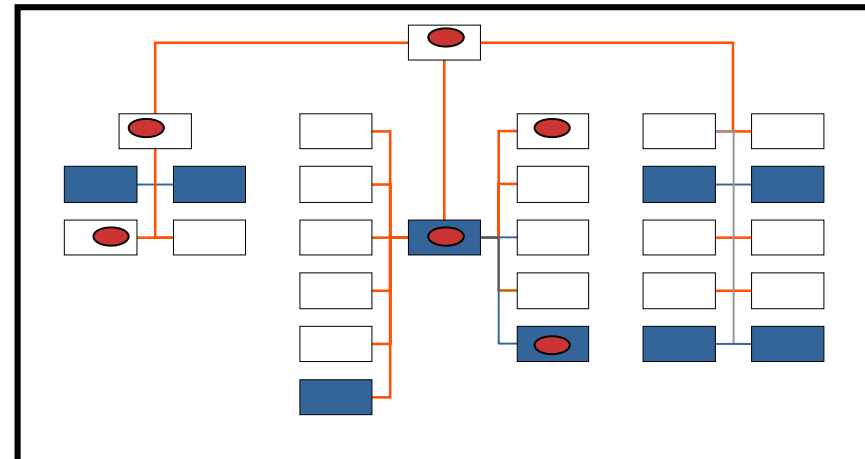
Eine Systemarchitektur

beschreibt die Struktur eines Systems hinsichtlich

- der funktionalen Zusammenwirkens und der Vernetzung der Systemelemente (Steuergeräte, Sensoren, Aktuatoren, Batterie, Rechnerknoten etc.)
- der Schnittstellen zu anderen Systemen
- der Umgebung und des Datenflusses im System
- der Daten- und SW-Architektur

Legende:

- Element
- Schnittstelle
- SW-Funktionen



Entwicklungsobjekt des Systemdesigns

Das Bordnetz

Zentrales Entwicklungsobjekt des Systemdesigns ist die Synthese eines Systems, das aufgrund der Konzentration auf die Schnittstellenfestlegung (System-)Architektur genannt wird.

Welches System?

Alle E/E-Komponenten, die vom Strom durchflossen sind:

- Steuergeräte,
- Sensoren,
- Aktuatoren,
- Batterie,
- Generator,
- Starter,
- Elektromotoren jeglicher Art,
- Busse,
- Leitungen u.v.m.

Die Gesamtheit aller Komponenten des E/E-Systems wird Bordnetz genannt.

.... Aber die Systemarchitektur ist mehr!

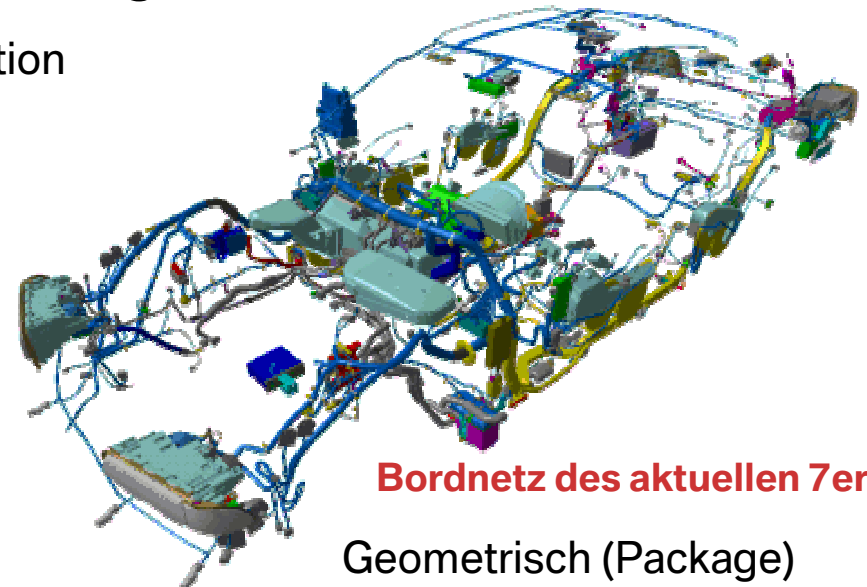
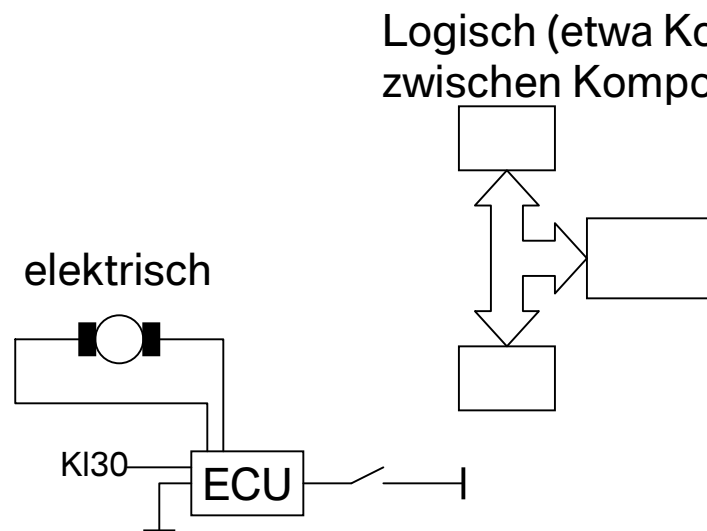
Entwicklungsobjekt des Systemdesigns

Systemarchitektur

Die Schnittstellenfestlegungen sind die zentralen Aufgaben bei der Erstellung der Systemarchitektur.

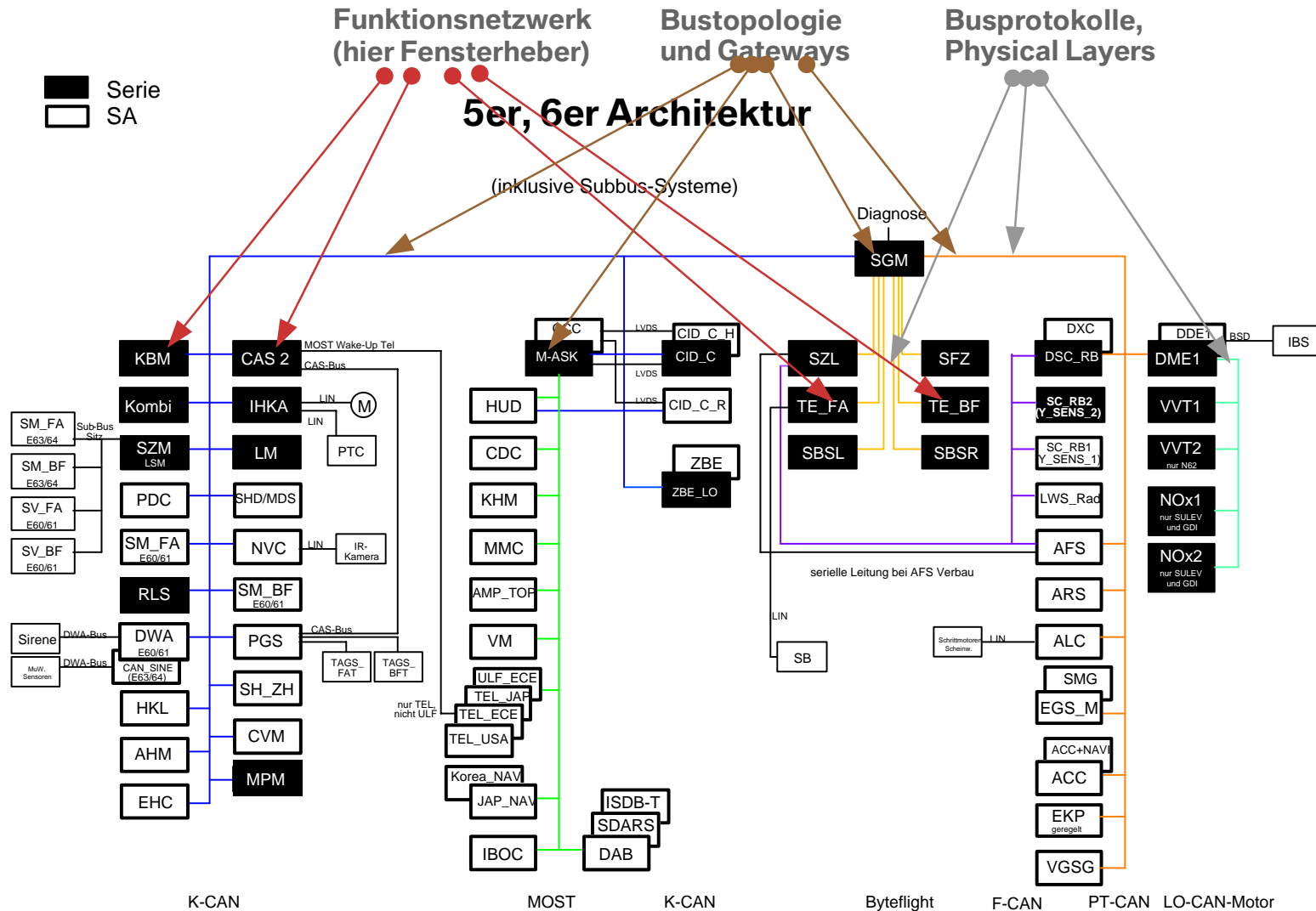
Die Schnittstellen sind **logisch, elektrisch und geometrisch festzulegen**, intern und extern der Komponenten; innerhalb der Systemumgebung und (sofern erforderlich) mit der Umgebung.

Eine Systemarchitektur lässt sich somit nur über verschiedenen **Sichten** vollständig beschreiben



Das Mehr einer Systemarchitektur

Eine Systemarchitektur ist (deutlich) mehr als die Summe der Komponenten! Das Add-On: Funktionen und deren Vernetzung sowie Verteilung, topologische Strukturen, Busprotokolle und Physical Layer, Gateways, Kommunikation und deren Struktur



Überblick

Entwicklungsobjekt

Funktionsnetze

Logische Architektur

Implementierungsarchitektur

Deployment

Definition Begriff Funktion

Eine Funktion ist dadurch definiert, dass sie im System Aufgaben erfüllt, um bestimmte Eigenschaften zu erzielen, die wiederum durch eine gewisse Anzahl wohldefinierter und messbarer Merkmale klassifiziert sind.

Beispiele:

Das Auto beschleunigen

Eigenschaft: in einer gewissen Zeit zu beschleunigen

Merkmale:

- von 0 auf 100 km/h in 6,3s
- von 0 auf 200 km/h in 15,5s
- von 60 auf 120 km/h im 5. Gang in 7s

Das Fenster der Beifahrerseite-vorne ist elektromotorisch zu schließen

Eigenschaften:

- Schließen des Fensters
- Beim Schließvorgang ist ein eingeklemmter Gegenstand zu erkennen.
- Wird ein eingeklemmter Gegenstand erkannt ist der Schließvorgang sofort zu beenden und der das Fenster öffnet wieder.

Merkmale:

- Ein vollständig geöffnetes Fenster schließt in 3,5s.
- Eine Gegenkraft von 100 N ist sicher zu erkennen.
- Der Einklemmschutz erfüllt die Sicherheitsanforderung Anforderungsklasse 5

Definition Begriff Funktion II

Function (IEEE12331, ITEA-EAST- and AUTOSAR-Glossary):

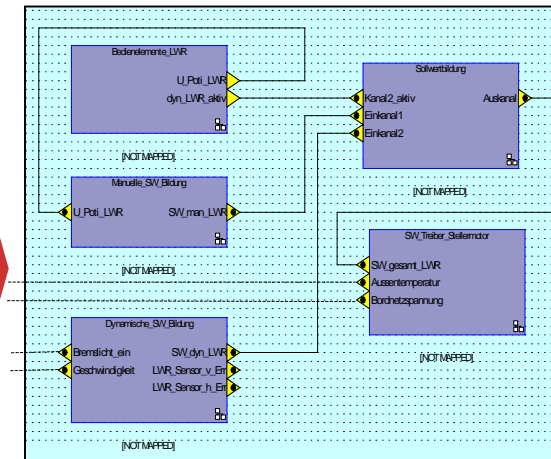
- i. A task, action or activity that must be accomplished to achieve a desired outcome.
- ii. part of programming code that is invoked by other parts of the program to fulfill a desired purpose.
- iii. In mathematics, a function is an association between two sets of values in which each element of one set has one assigned element in the other set so that any element selected becomes the independent variable and its associated element is the dependent variable

Übergang von Anforderungen zur logischen Architektur und Funktionsnetzen

Die Anforderungen werden durch Funktionen und deren Vernetzung in der Systemarchitektur konkretisiert und virtuell realisiert.

Funktion	DC_E270_CDI_MJ_2003	a_ev_pl4_basic	a_ev_pl4_mid	a_ev_pl4_high
2.3.1.2 Senderauswahl				
Anzeige der Feldstärke	nein			
Senderauswahl nach Programmtyp	ja	nein	ja	ja
Darstellung des Sendernamens	ja	ja	ja	ja
Ein/Ausschalten der RDS-Erkennung	ja	nein	nein	nein
Anzeige einer Regionalisierung	nicht ermittelbar	ja	ja	ja
Ein/Ausschalten der Regionalisierung	ja			
Darstellung der Senderfrequenz als Zahlenwert	ja	ja	ja	ja
Direkte Frequenzeingabe	ja	ja	ja	ja
Automatische Suche von stärksten Sendern	ja	ja	ja	ja
Automatische Suche von Sendern auf der nächst höheren oder der nächst niedrigeren Frequenz	ja	ja	ja	ja
Automatische Suche der gespeicherten Sender	?	ja	ja	ja
Stationspeicher	ja, 4 x 10	ja, 10 Speichertasten	ja, 10 Speichertasten	ja, 10 Speichertasten
Speicherung der letzten gehörten Station	ja	ja	ja	ja
Speicherung des Lieblingsenders	ja	nein	ja	ja
2.3.1.3 Senderliste				
Senderlistenfunktionalität	ja, 10	nein	nein	ja
Senderlistenaufbau ohne Unterbrechung des gehörten Programms	nein	nein	nein	ja
Dynamische Anpassung an vorhandene Sender	nicht ermittelbar	nein	nein	ja
Vorauswahl der am besten zu empfangenden Sender	nein, nur nach Frequenzen	ja	ja	nein
Vorauswahl von Lieblingssendern (Hörgewohnheitsliste)	nicht ermittelbar	nein	nein	ja
Sortierung der Senderliste nach Alphabet	nein	nein	nein	ja
Sortierung der Senderliste nach Frequenz	ja
2.3.1.4 Verkehrsfunk				
Vorauswahl von Verkehrsfunksendern	ja?	nein	nein	ja
Kennzeichnung der Verkehrsfunksender	ja	ja	ja	ja

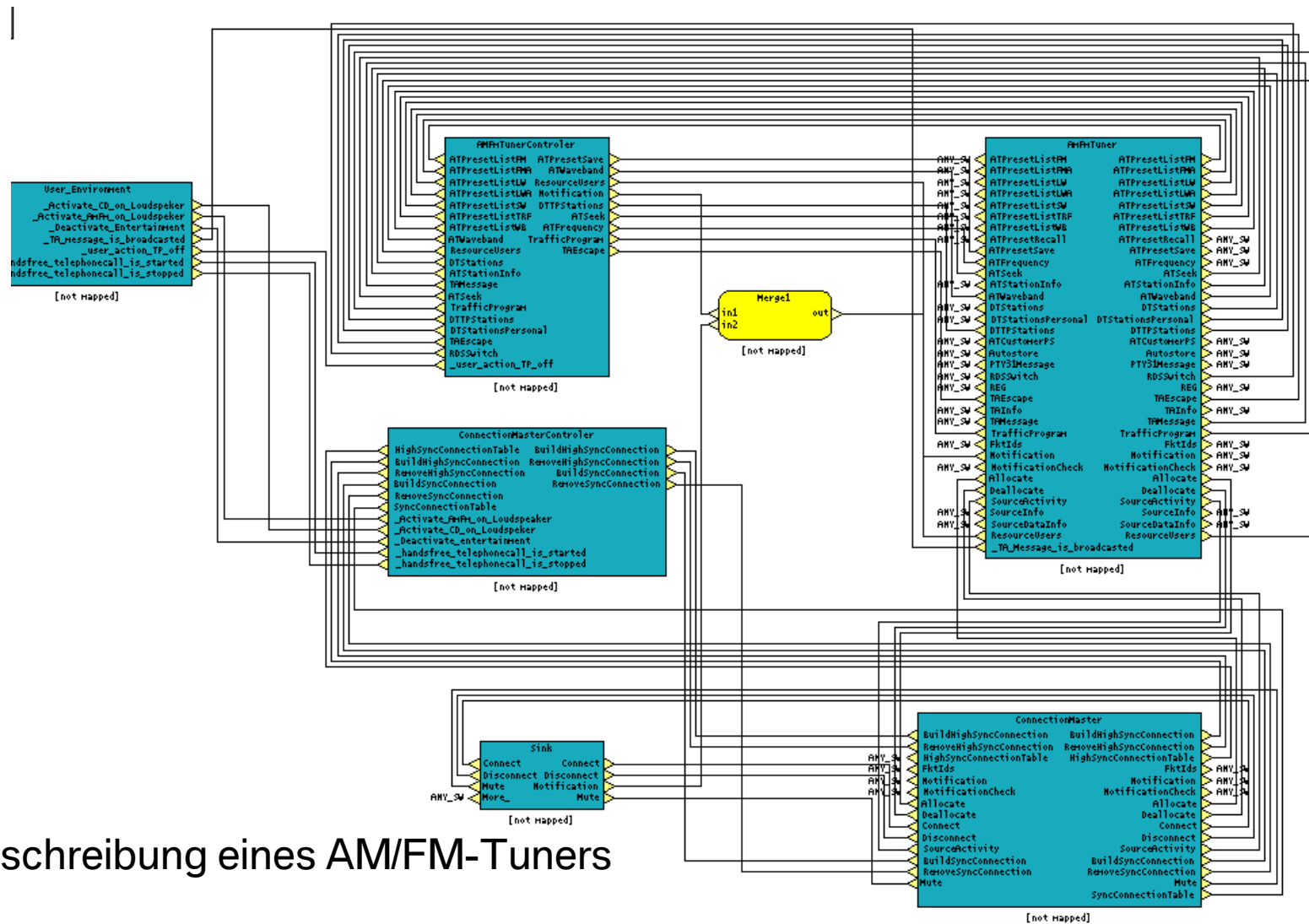
Funktionsnetze



Technische Anforderungen

Beschreibungsmittel für Funktionsnetze

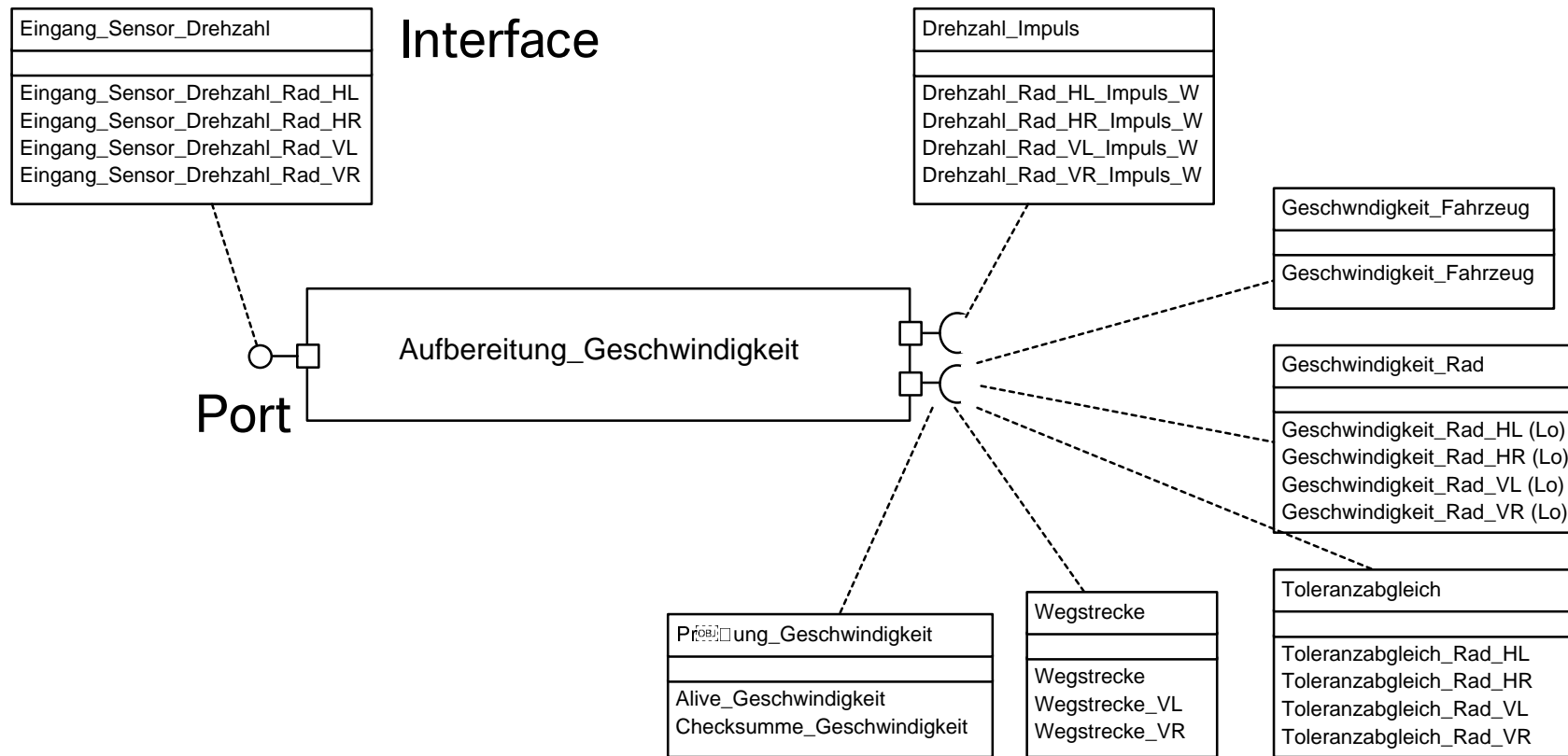
Datenflussbeschreibungen sind dem Ingenieur sehr vertraut.



Datenflussbeschreibung eines AM/FM-Tuners

Beschreibungsmittel für Funktionsnetze

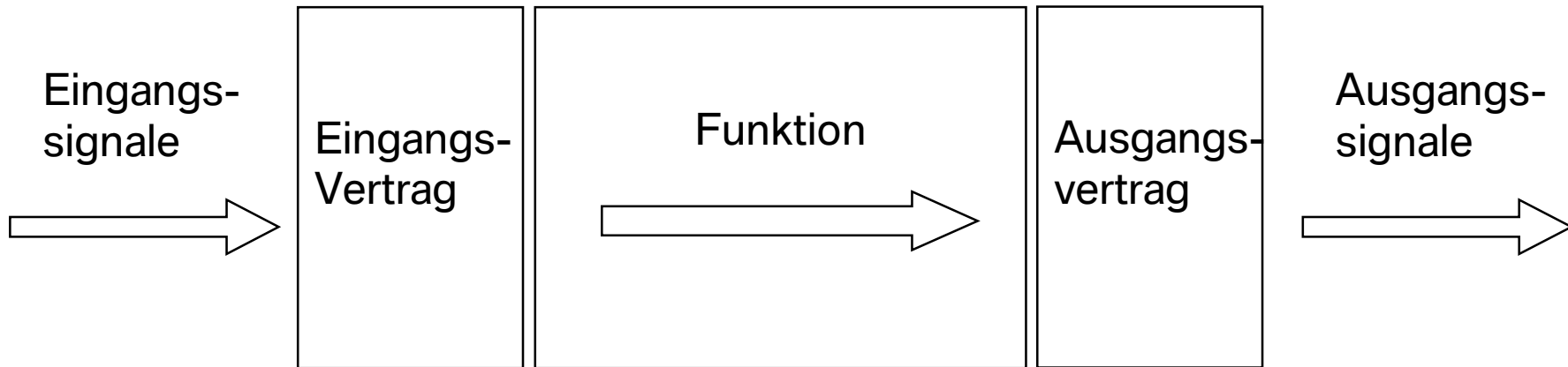
Komponentenbasierte UML-Beschreibung sind aus dem SW-Engineering bekannt



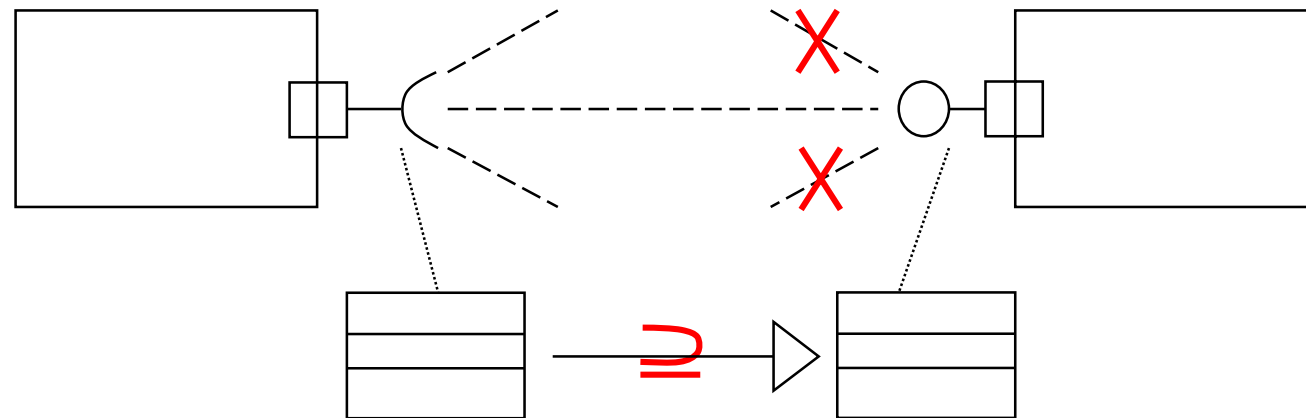
Interfaces implementieren Ports

Beispiel aus DSC (Dynamische Stabilitätskontrolle)

Vertragsmetapher für Funktionen

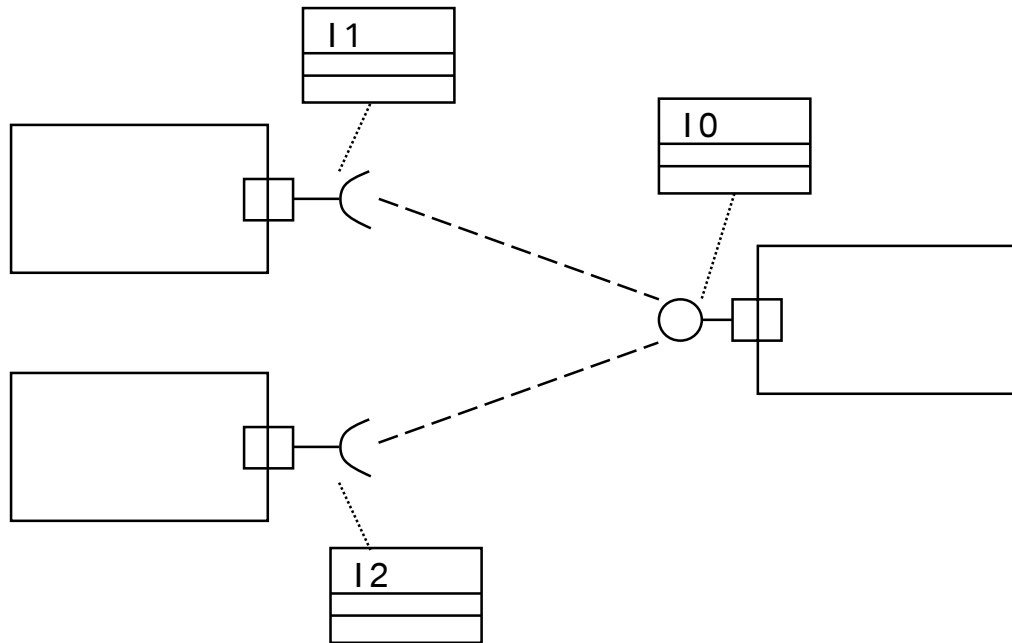


Signalflüsse zwischen Funktionen einer Hierarchieebene



➔ Eindeutige Rückverfolgbarkeit von Signalen

Signale von verschiedenen Funktionen



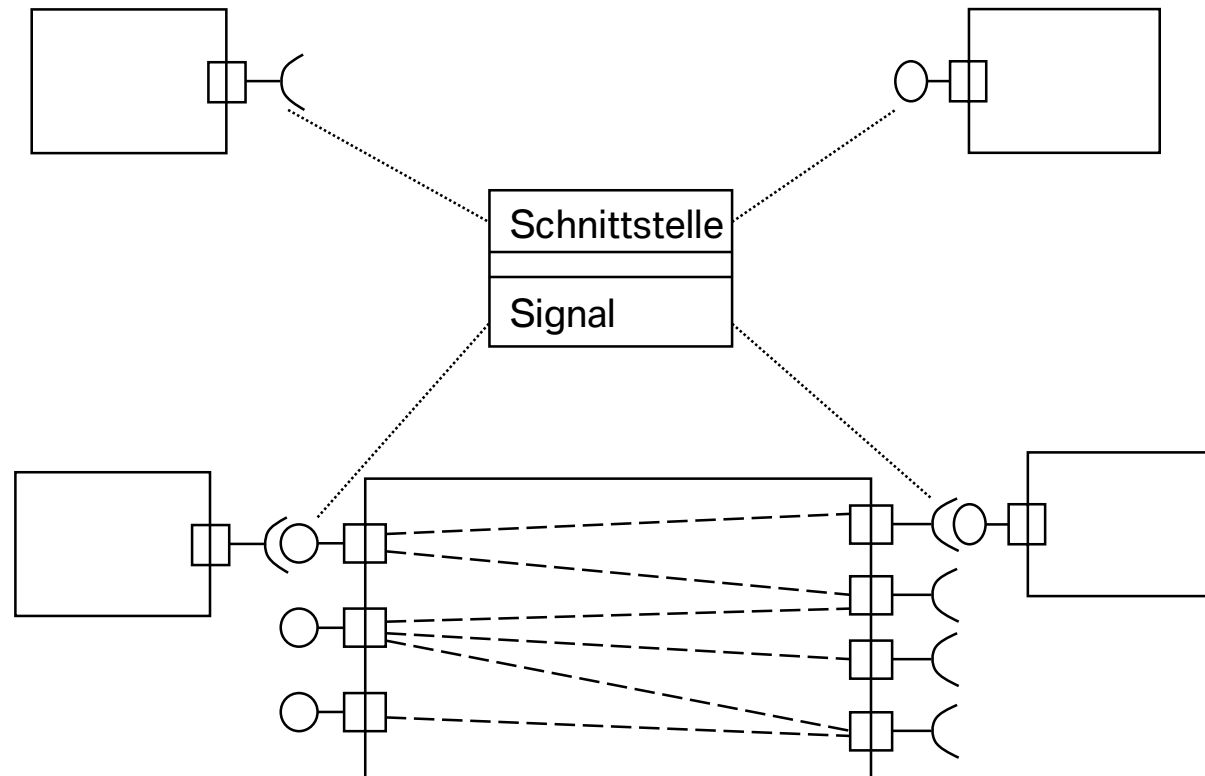
Verbindung

$$I1 \cup I2 \ni IO$$

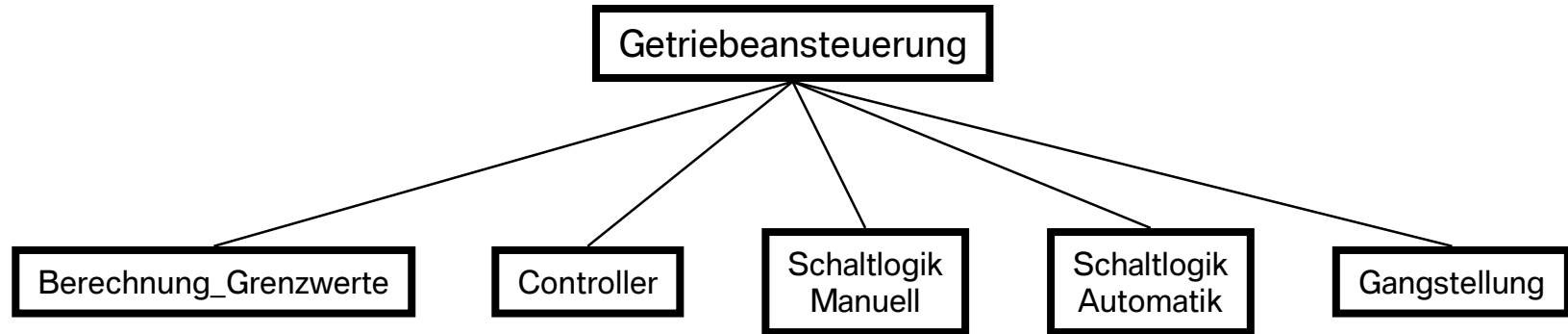
Rückverfolgbarkeit

$$I1 \cap I2 = \emptyset$$

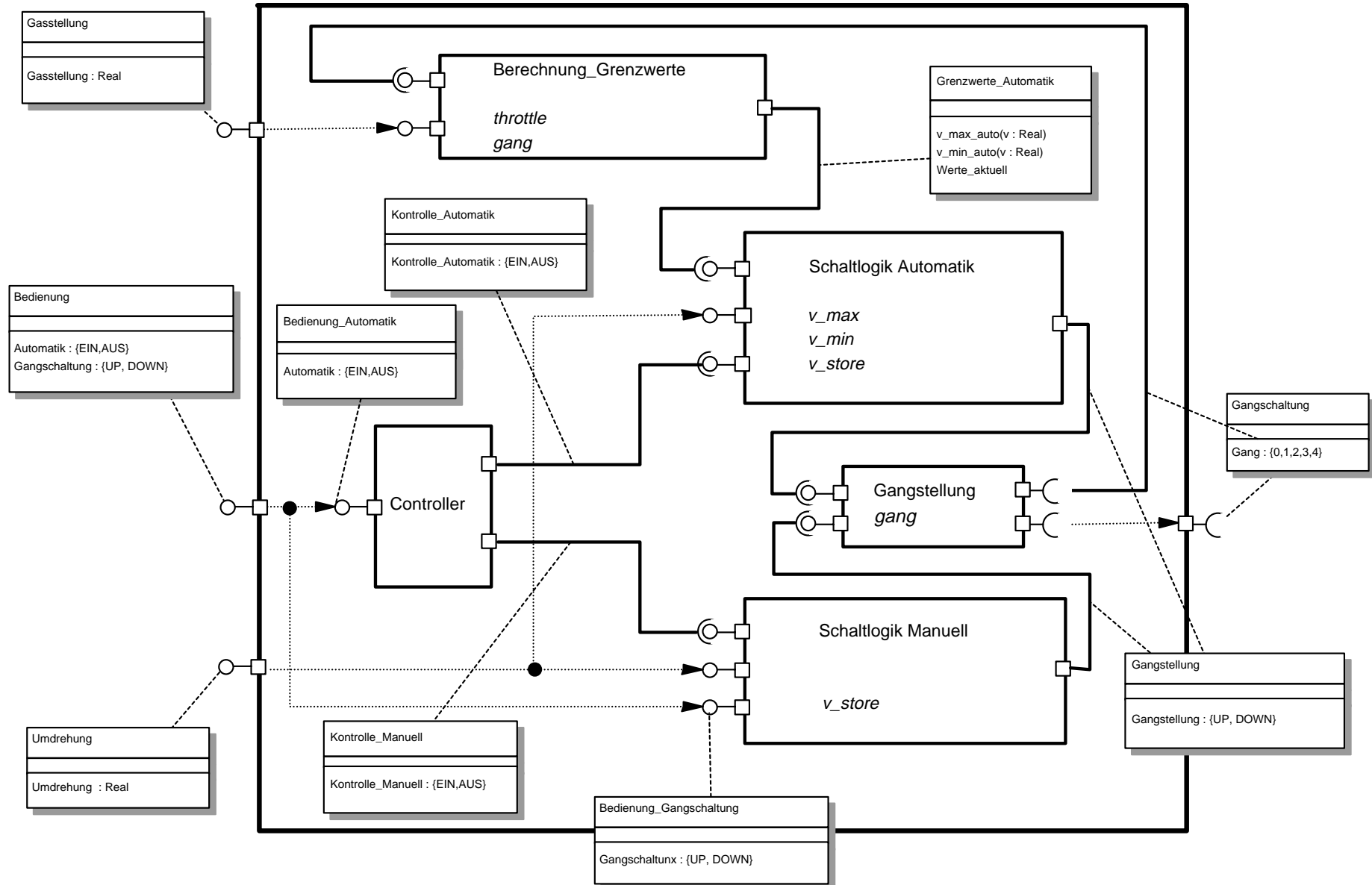
Signalflüsse innerhalb von Funktionen



Hierarchische Funktionen : Baumdarstellung



Hierarchische Funktionen (z.B. Getriebeansteuerung)



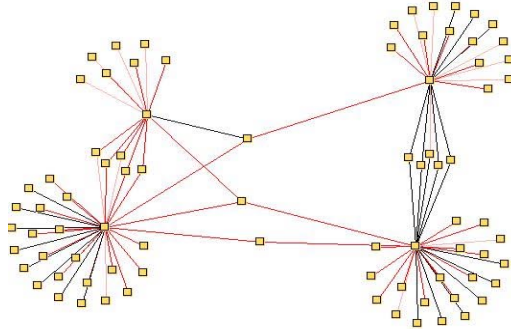
Komplexität von Bordnetzen

Anhand von Funktionsnetzen lässt sich viel besser die Komplexität die Komplexität von heutigen E/E-Systemen beschreiben und illustrieren.

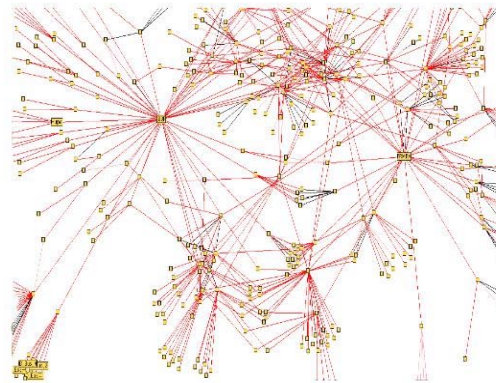
Ein relativ einfaches Komplexitätsmaß ist gegeben durch

$$K = N_{\text{Function}} * M_{\text{Signal}} \quad (\text{Dynaxity})$$

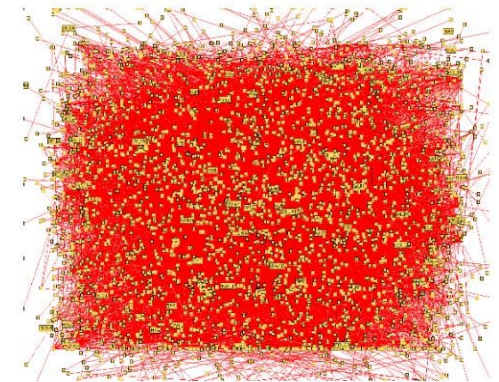
Fensterheber



DSC-System



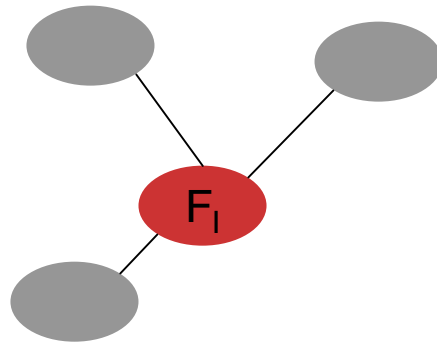
Gesamtes Bordnetz



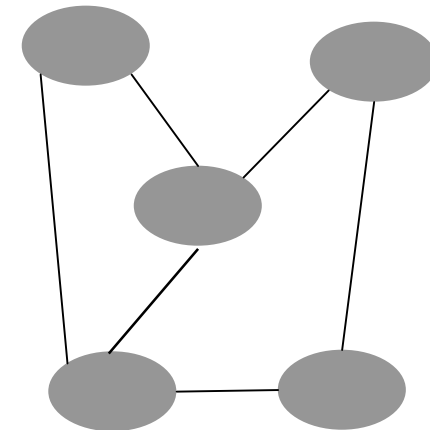
Höhere Komplexitätsmaße lassen sich aus Funktionen und ihre unmittelbare Umgebung ableiten

Es lassen sich dann generalisierende Durchschnittswerte für das Gesamtbordnetz ableiten.

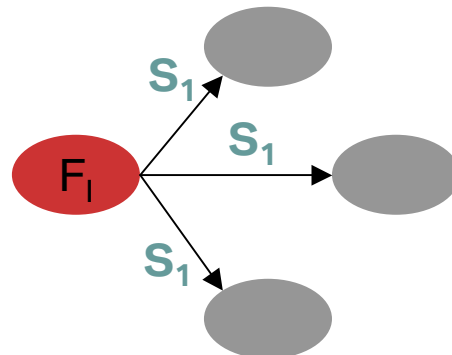
Mit wievielen Kommunikationspartnern ist jede Funktion verbunden?



Ist alles mit allem vernetzt oder gibt es logische Inseln im System?



Wieviele Funktionen erreicht ein ausgesendetes Signal? Wie wichtig ist somit dieses Signal?



Überblick

Entwicklungsobjekt

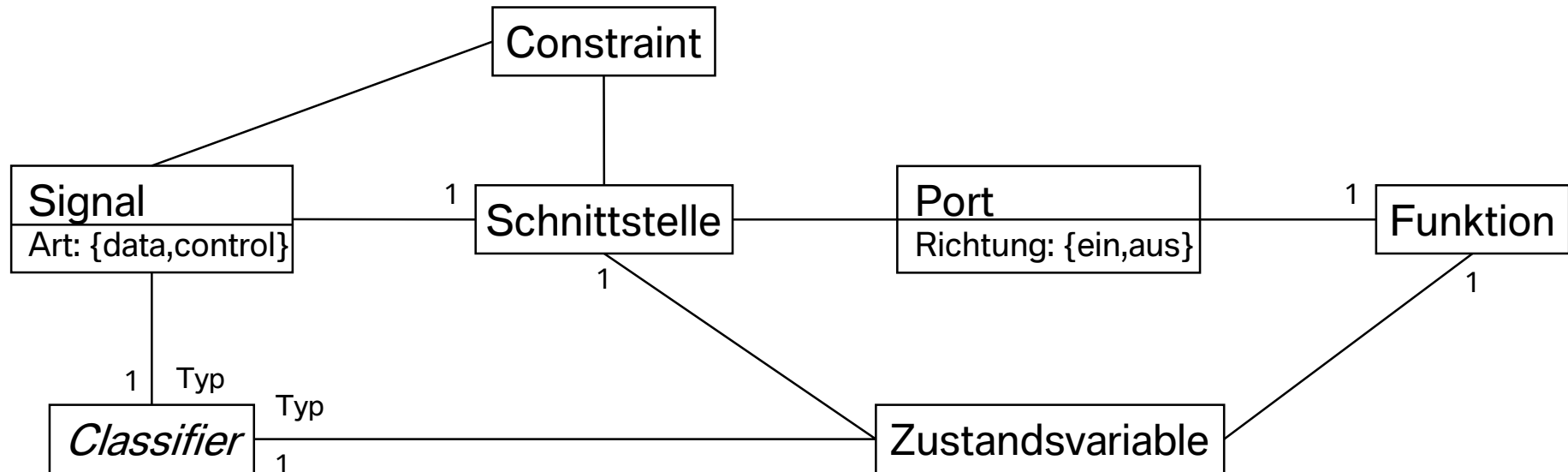
Funktionsnetze

Logische Architektur

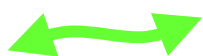
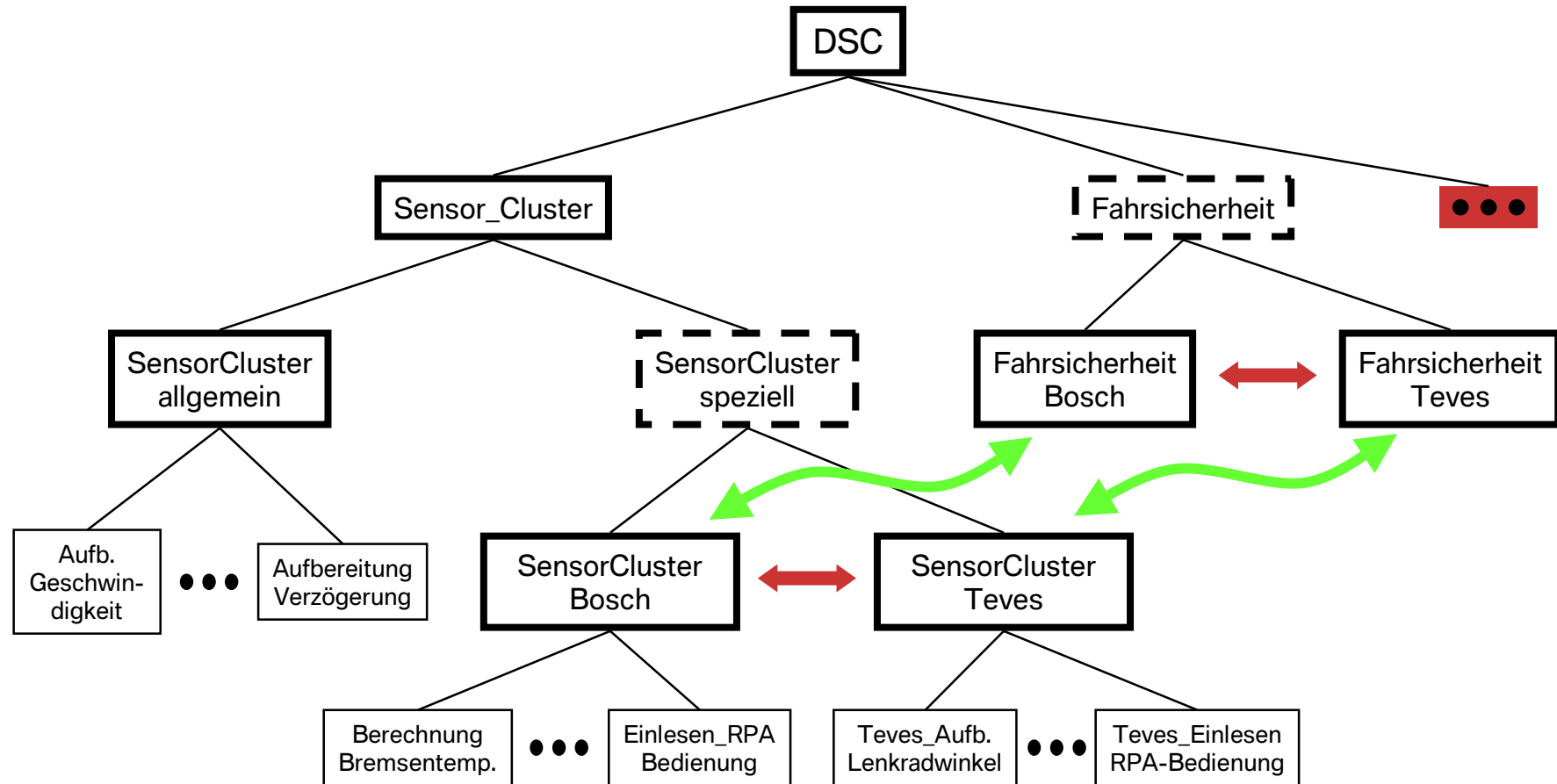
Implementierungsarchitektur

Deployment

Struktur von Funktionen und Signalen als Metamodell



Varianten: Eine Besonderheit in der Fahrzeugtechnik stellt die Behandlung von Varianten dar.

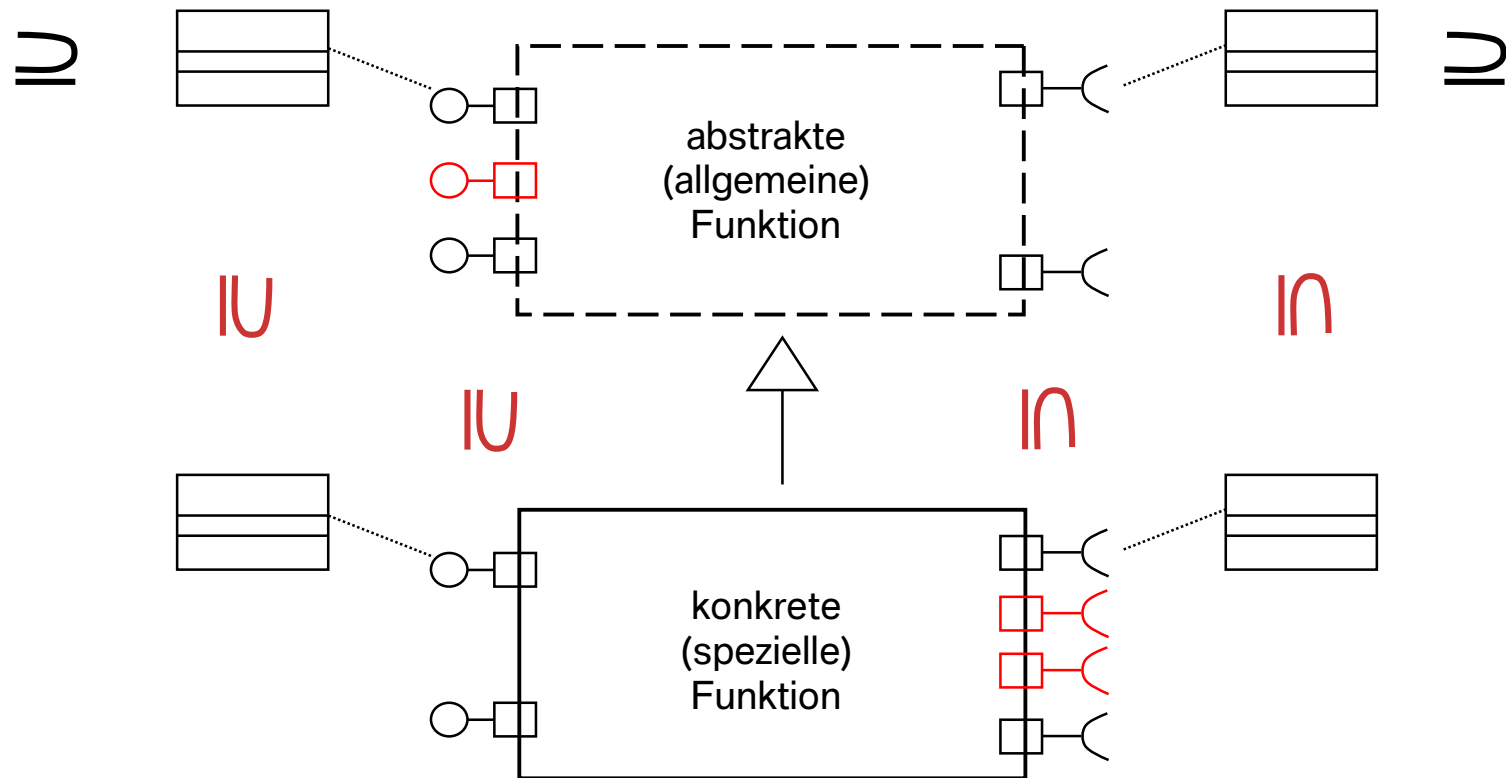


Needs-Beziehung



Excludes-Beziehung

Applikationsentwicklung: Spezialisierung von Funktionen



Überblick

Entwicklungsobjekt

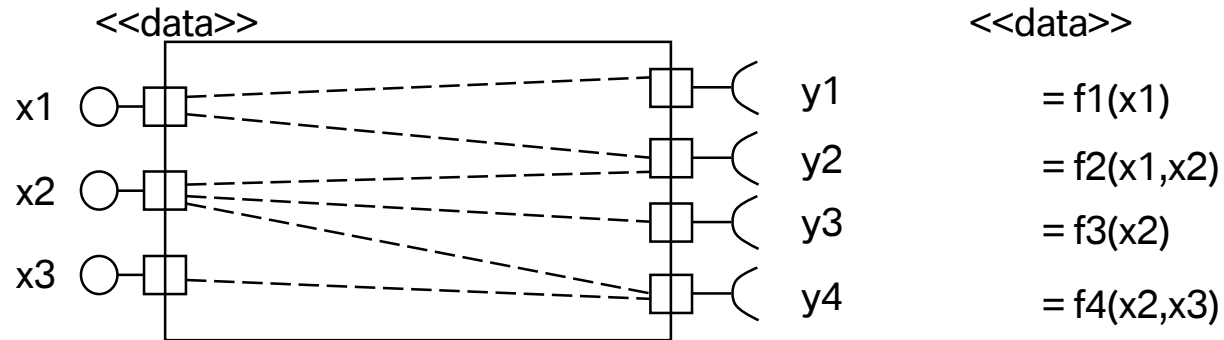
Funktionsnetze

Logische Architektur

Implementierungsarchitektur

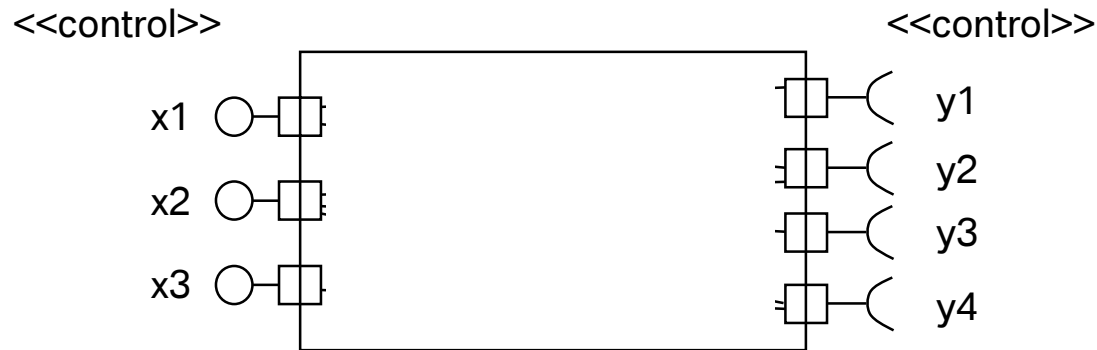
Deployment

Internes Verhalten (Signal-Abhängigkeiten in) einer Funktion

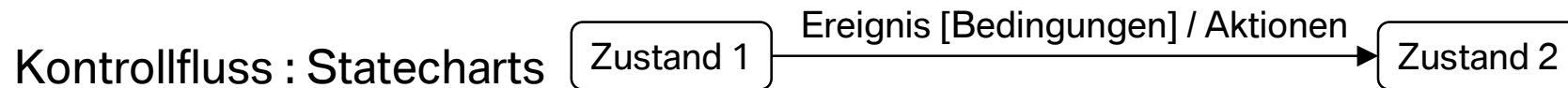


Datenfluss : math. Beschreibung $y_i = f_i(x_1, \dots, x_n)$

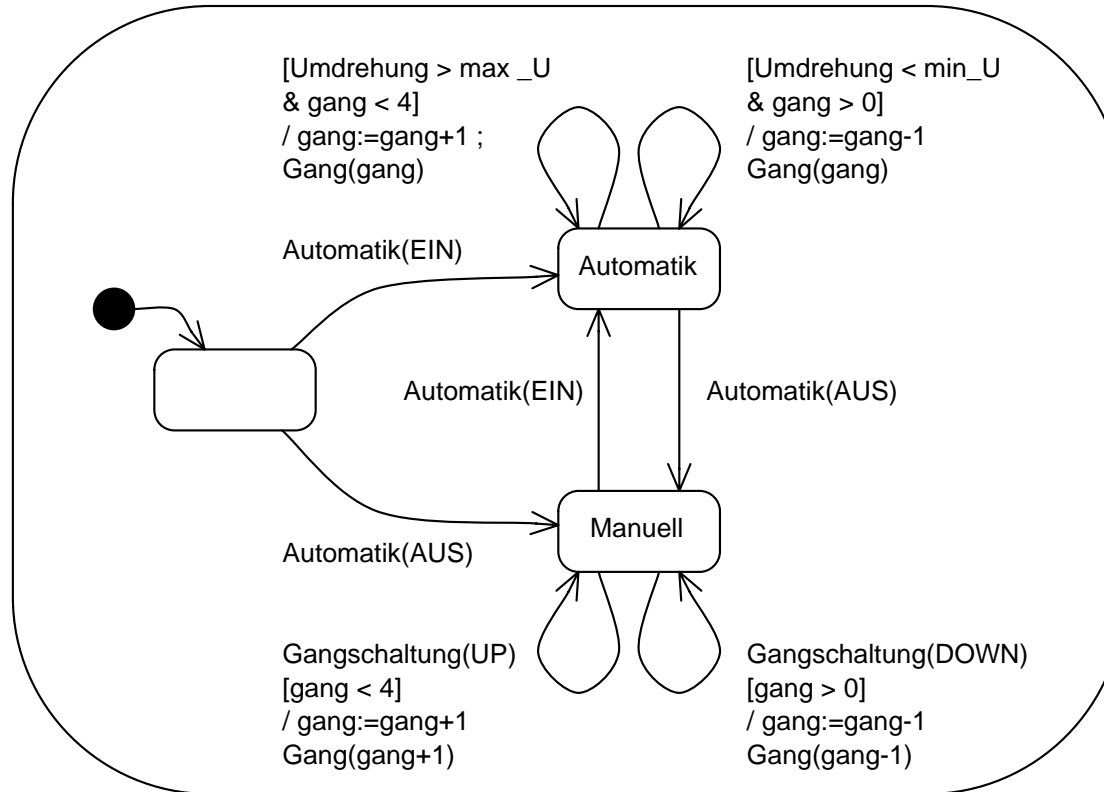
Internes Verhalten (Signal-Abhängigkeiten in) einer Funktion



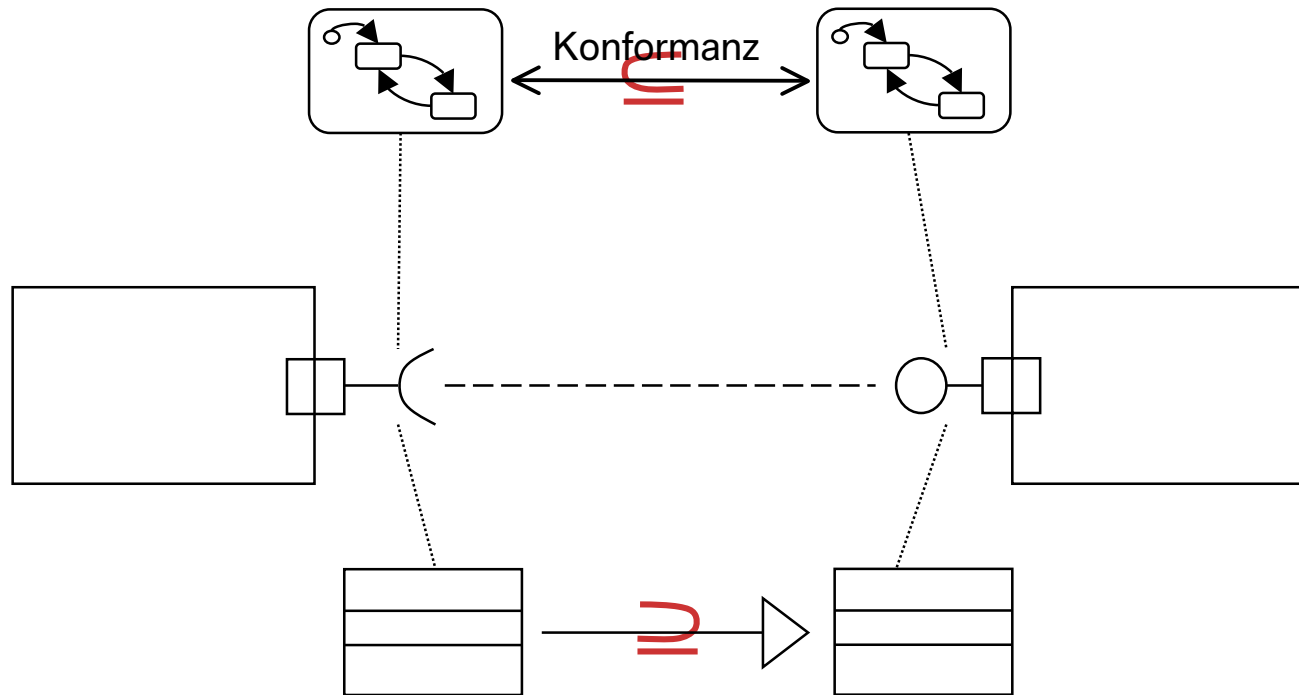
Datenfluss : math. Beschreibung $y_i = f_i(x_1, \dots, x_n)$



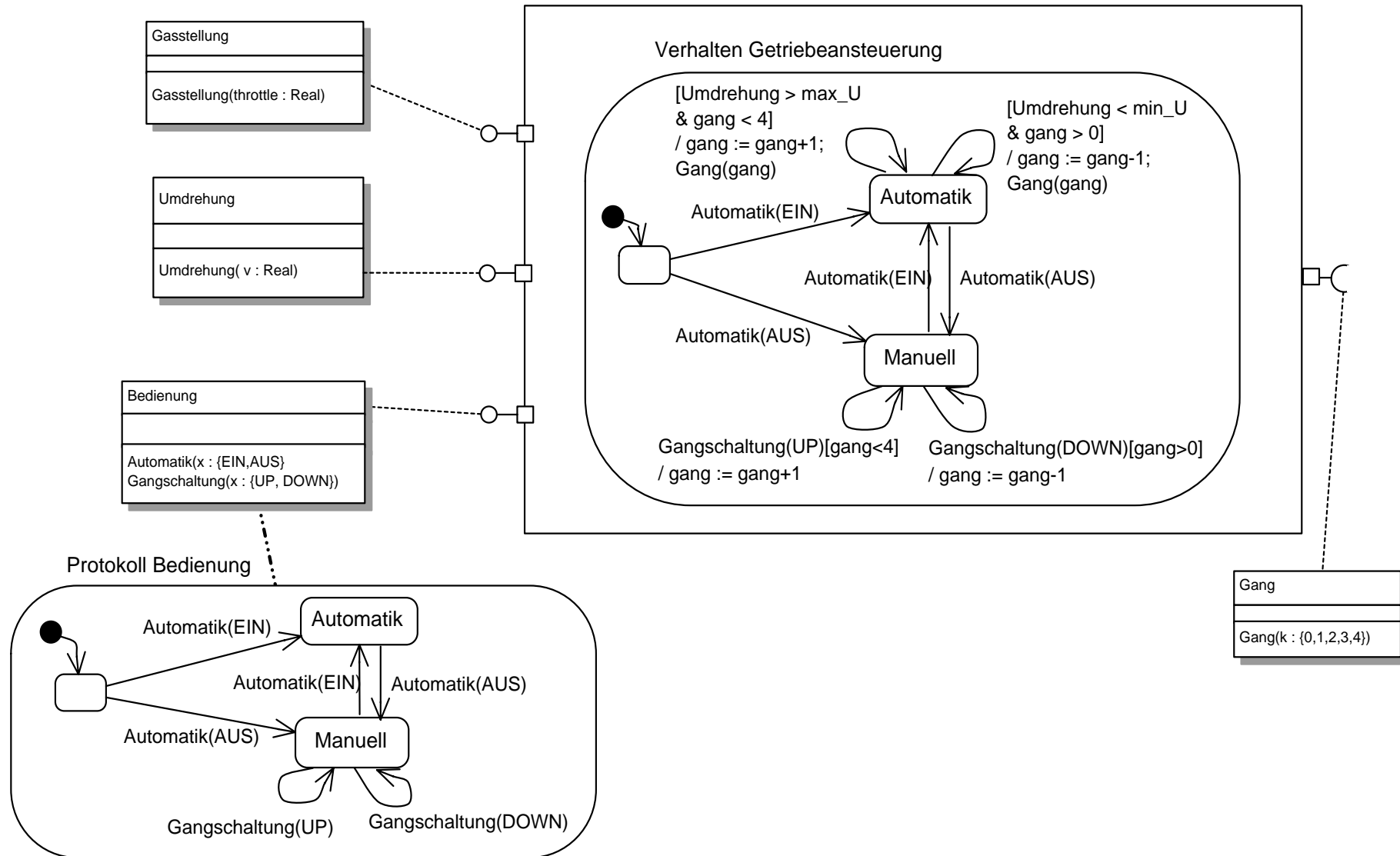
Bsp. Statechart der Getriebeansteuerung



Protokoll-Statecharts als Vertragsanteile



Externes Verhalten (Protokoll-Statecharts)



Abstraktes Verhaltens- und Fehlermodell

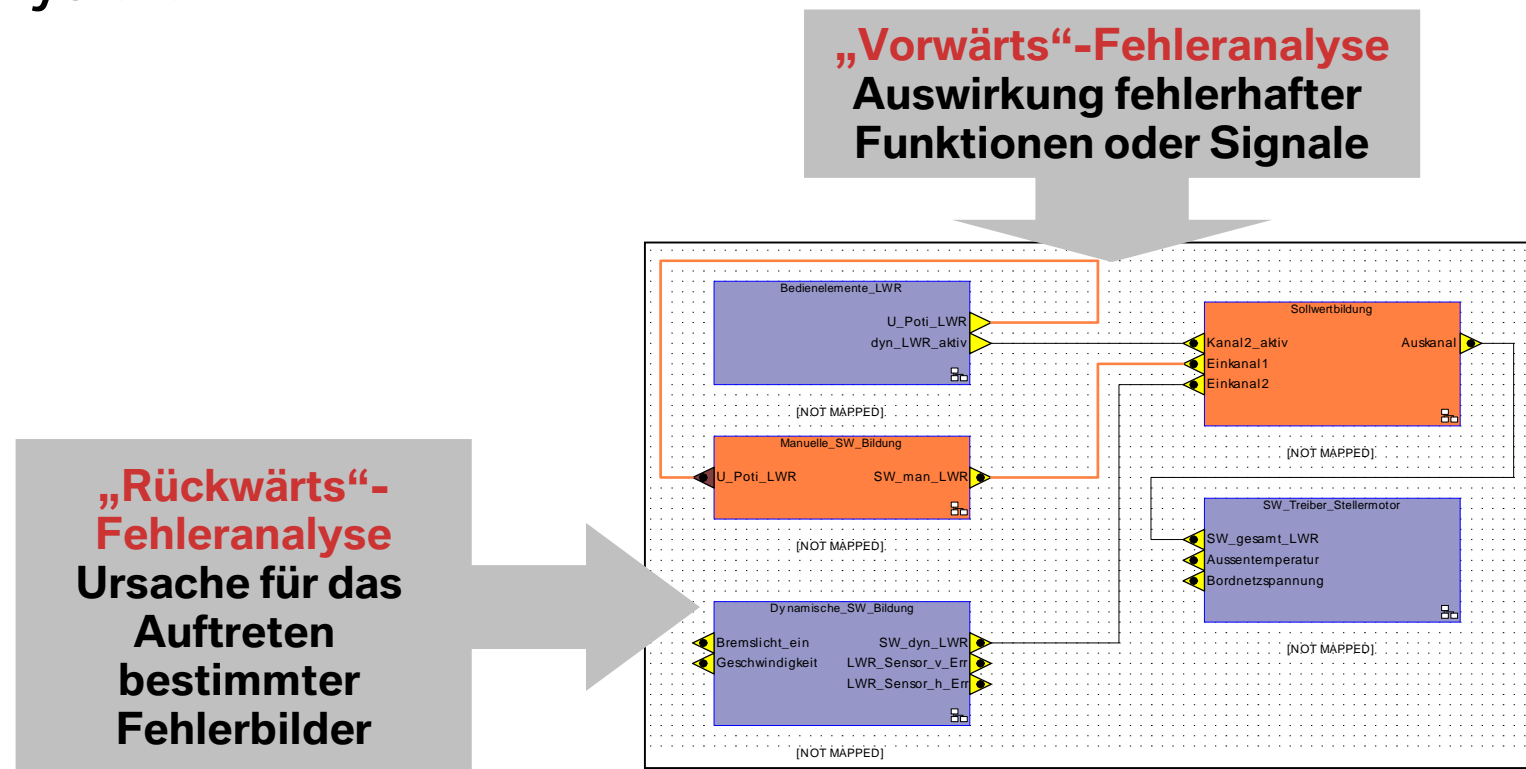
- Abstraktion
 - Signal JA / NEIN
- Erweiterung
 - Fehler = Verletzung des Eingangsvertrags
 - Fehlerklassen

Vertrag

- **Signaldeklaration**
- Wertebereich
- Statische Abhängigkeit (Constraint)
- Zykluszeit
- Reihenfolge

Fehleranalyse auf Basis von Funktionsnetzen

Eine sehr mächtige entwicklungsbegleitende Methodik ist die Hinterlegung von Fehlermodellen (z.B. in Form von Fehlermatrizen), um schon sehr früh Ursache-Wirk-Zusammenhänge durch das gesamte Bordnetz zu analysieren



Fehlermatrizen

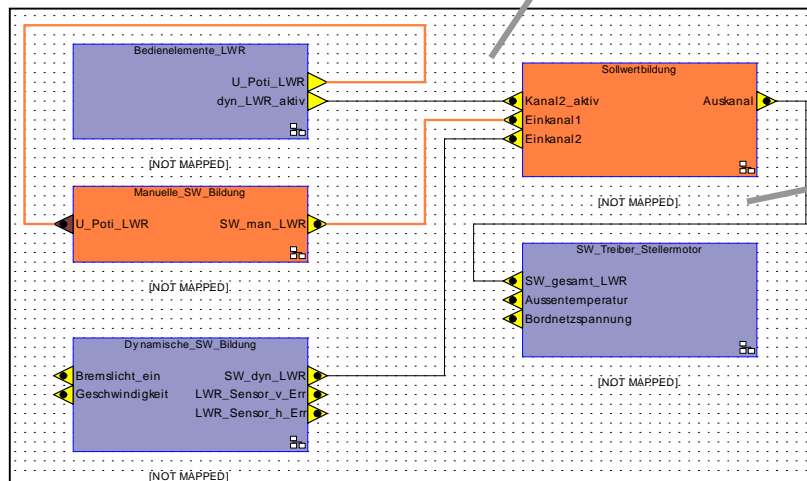
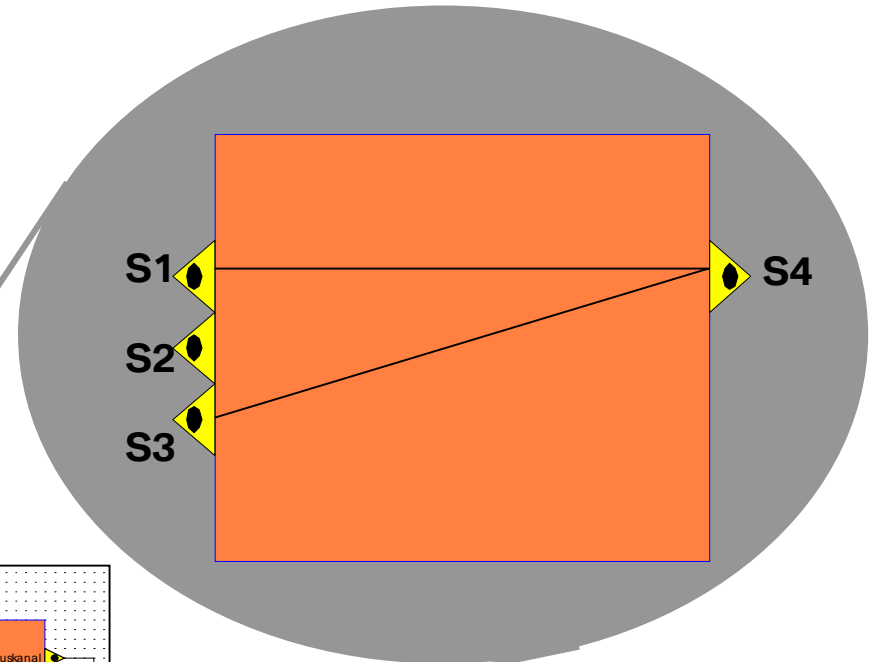
Sie stellen Abhängigkeiten der Ausgangssignale von fehlerbehafteten Eingängen dar.

Einfachste Form!

Weitere Merkmale:
Signalqualitäten
Wertebereiche

In Darstellung sind folgende Aussagen möglich:
S4 = S1 oder S3
S4 = S1 und S3
S4 = S1(t =>to)
Fehler

Redundanz!
Time-out-



Überblick

Entwicklungsobjekt

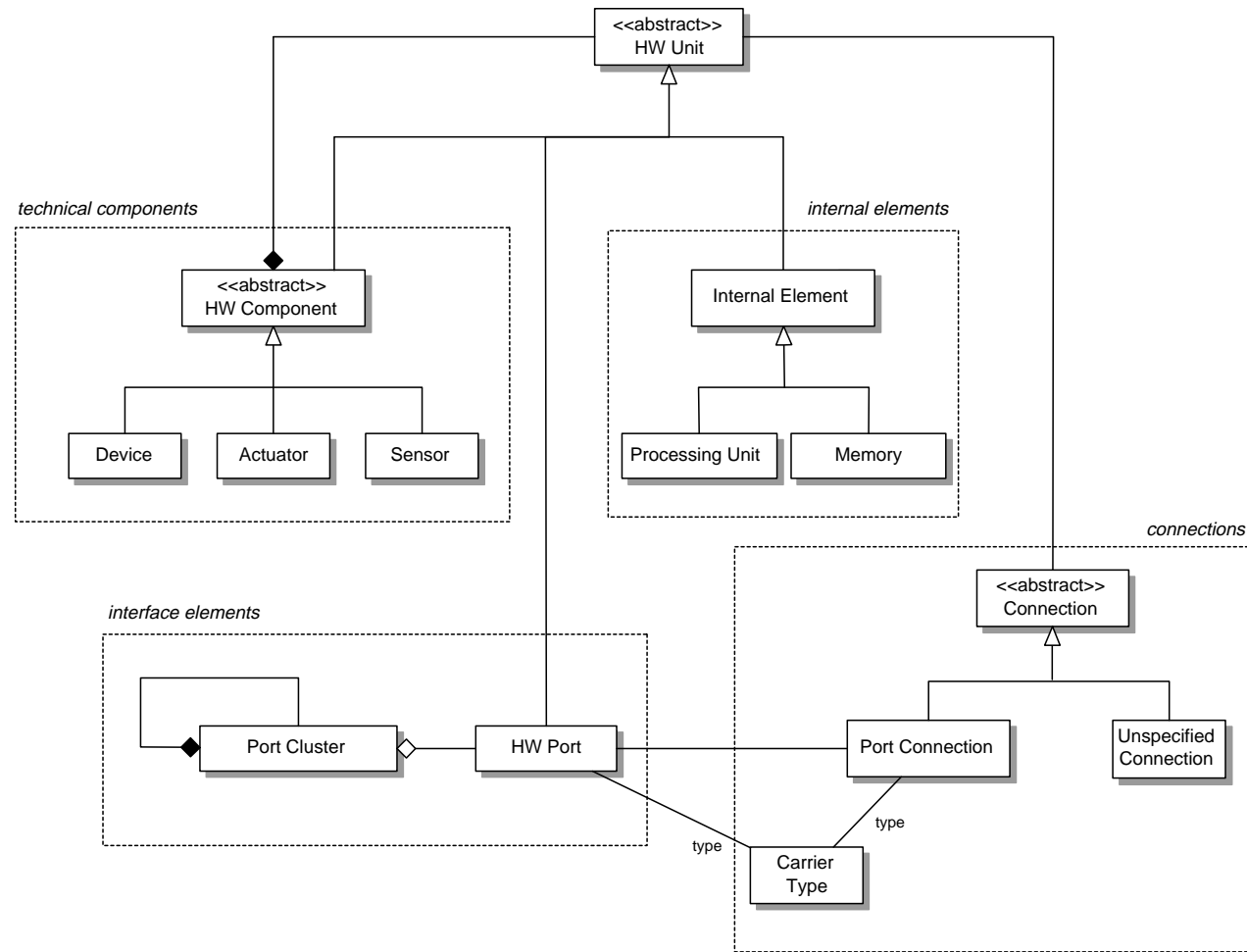
Funktionsnetze

Logische Architektur

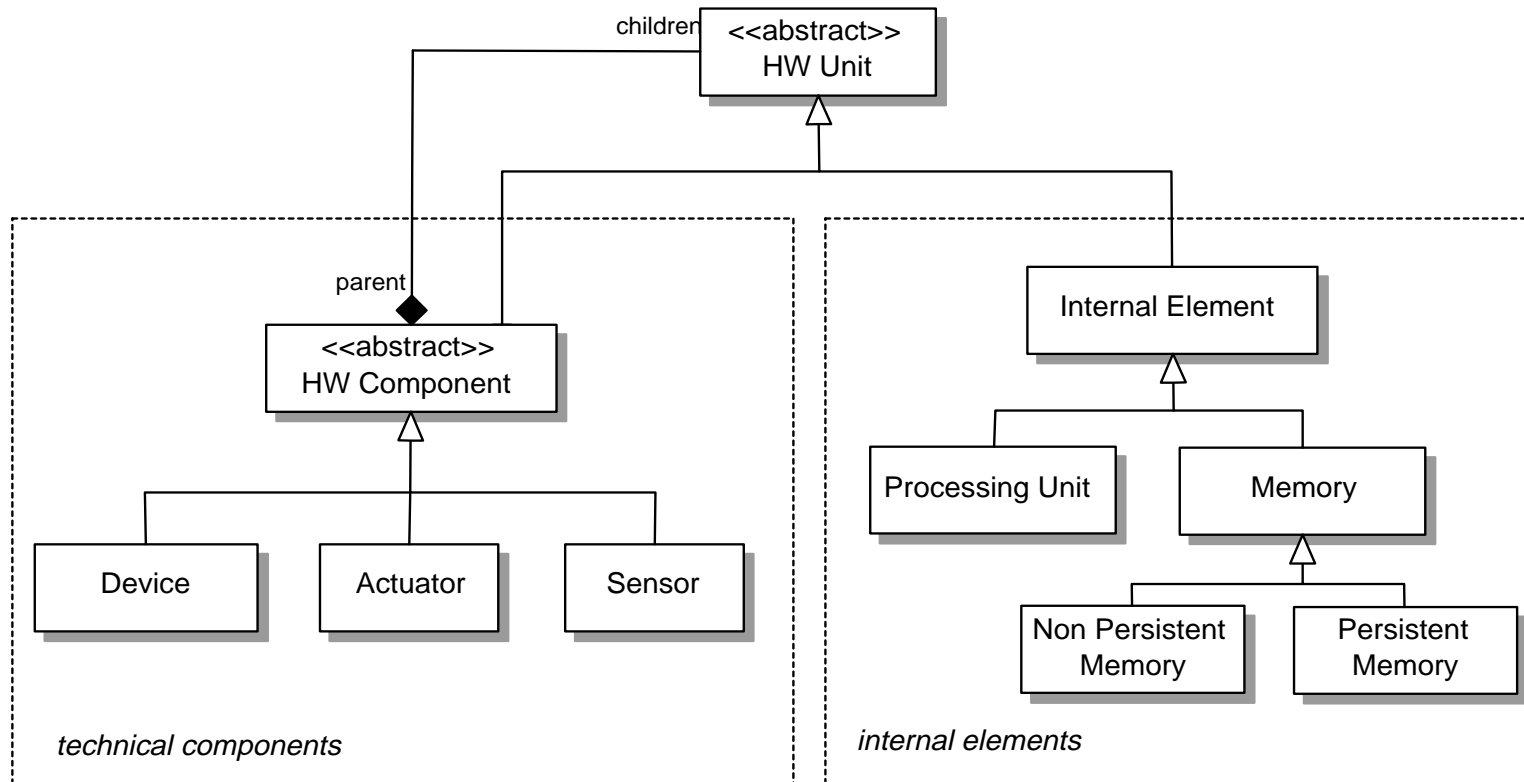
Implementierungsarchitektur

Deployment

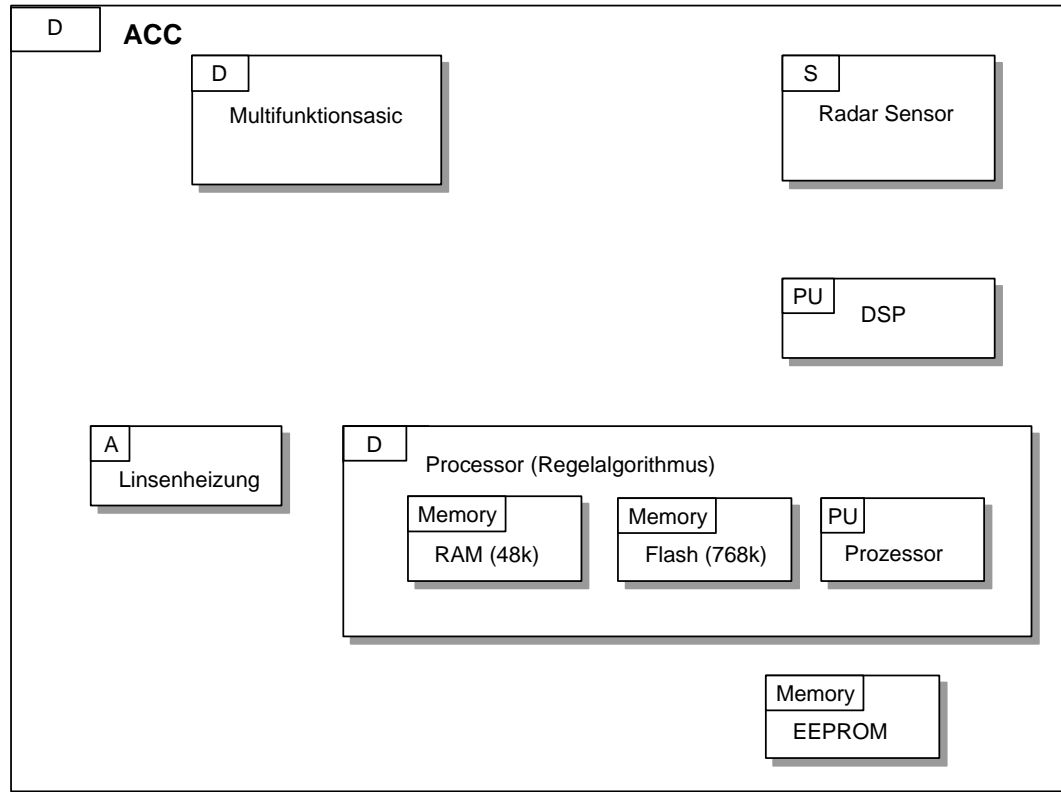
Übersicht eines Metamodells für technische Architekturen



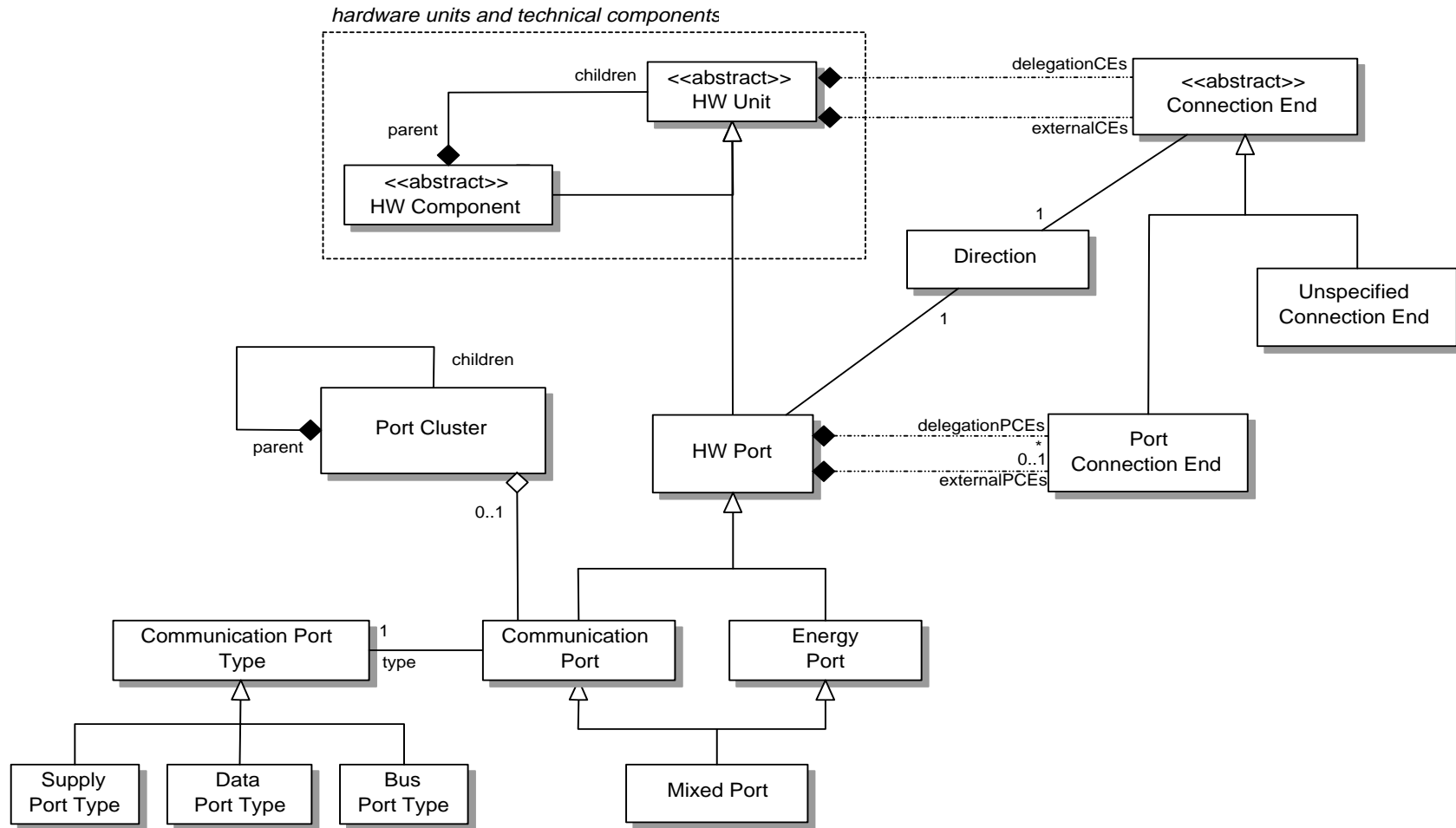
Technische Komponenten und interne Elemente



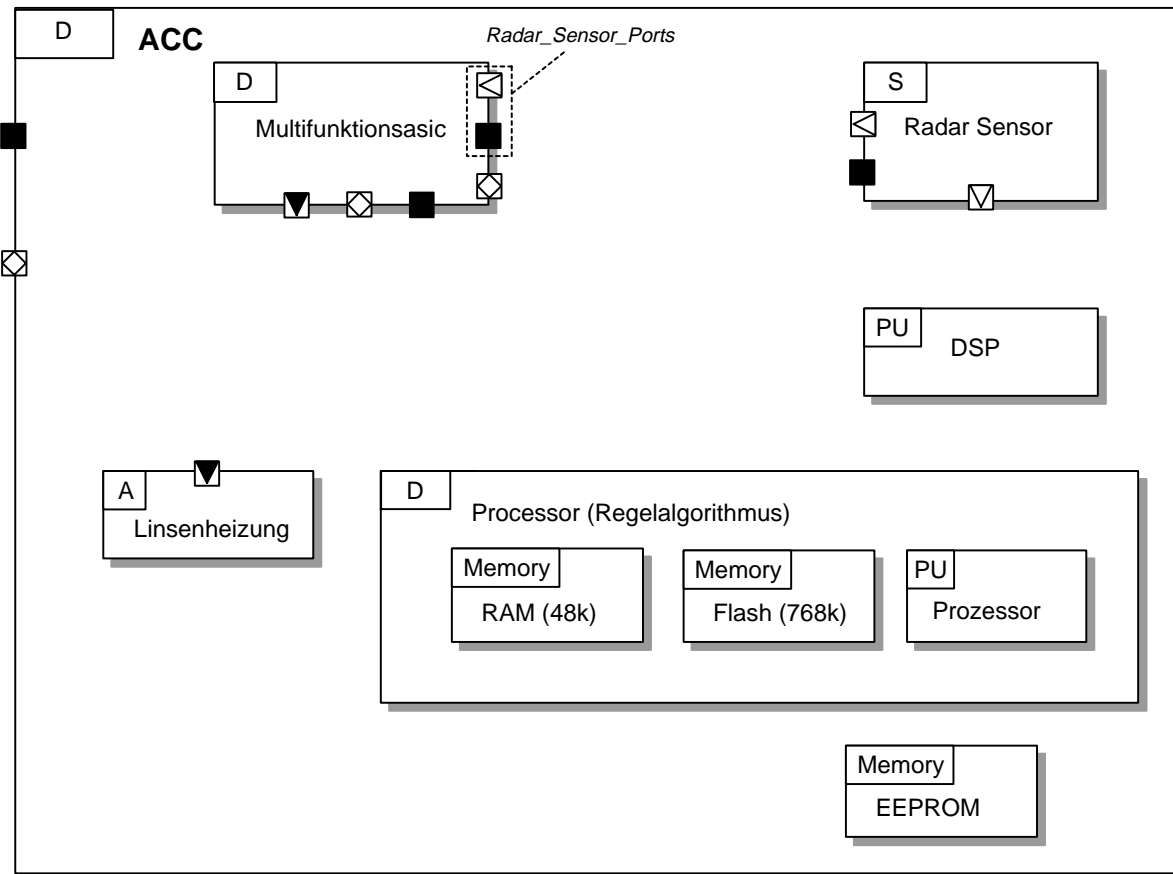
Graphische Notation für technische Komponenten und interne Elemente



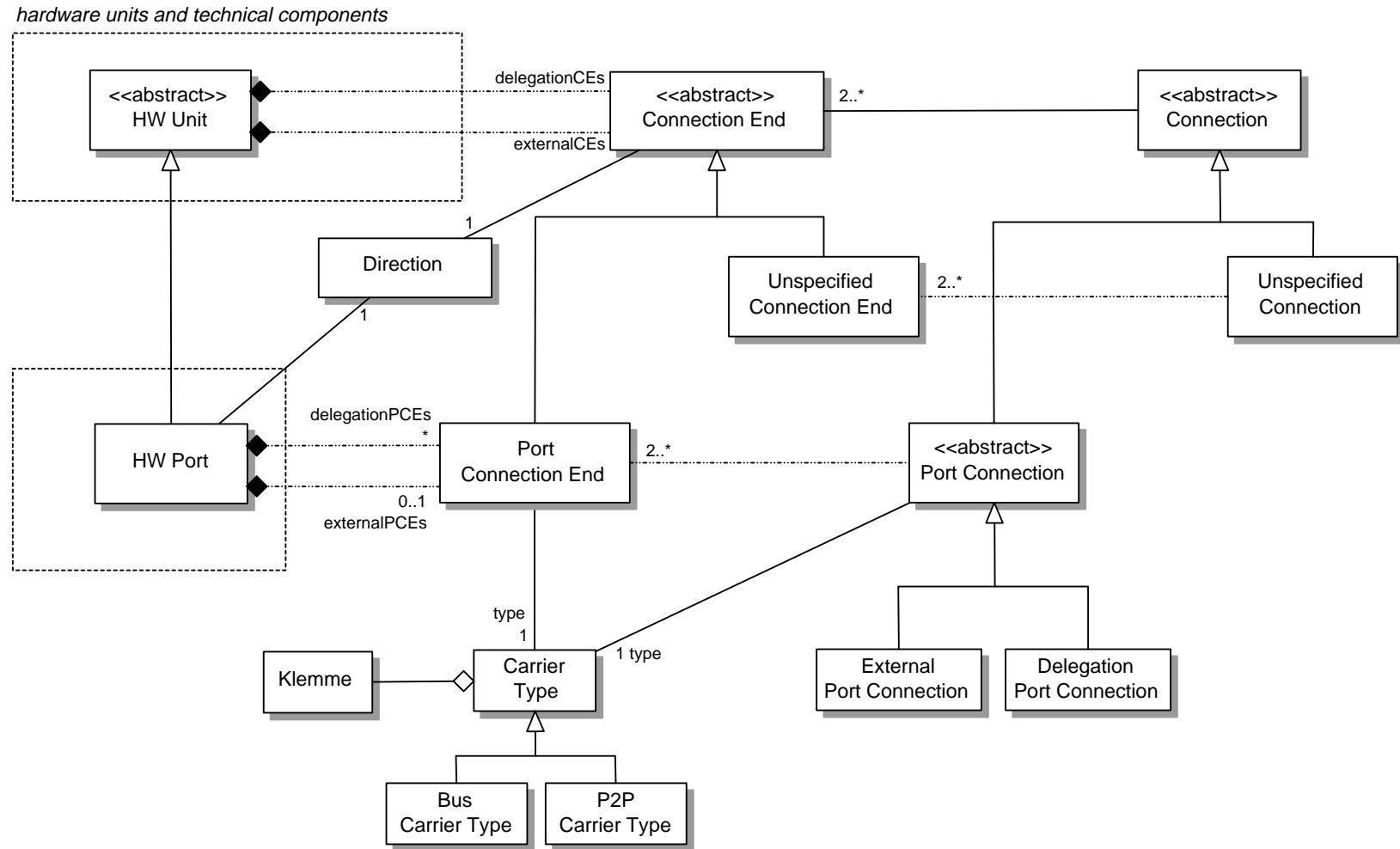
Schnittstellenelemente



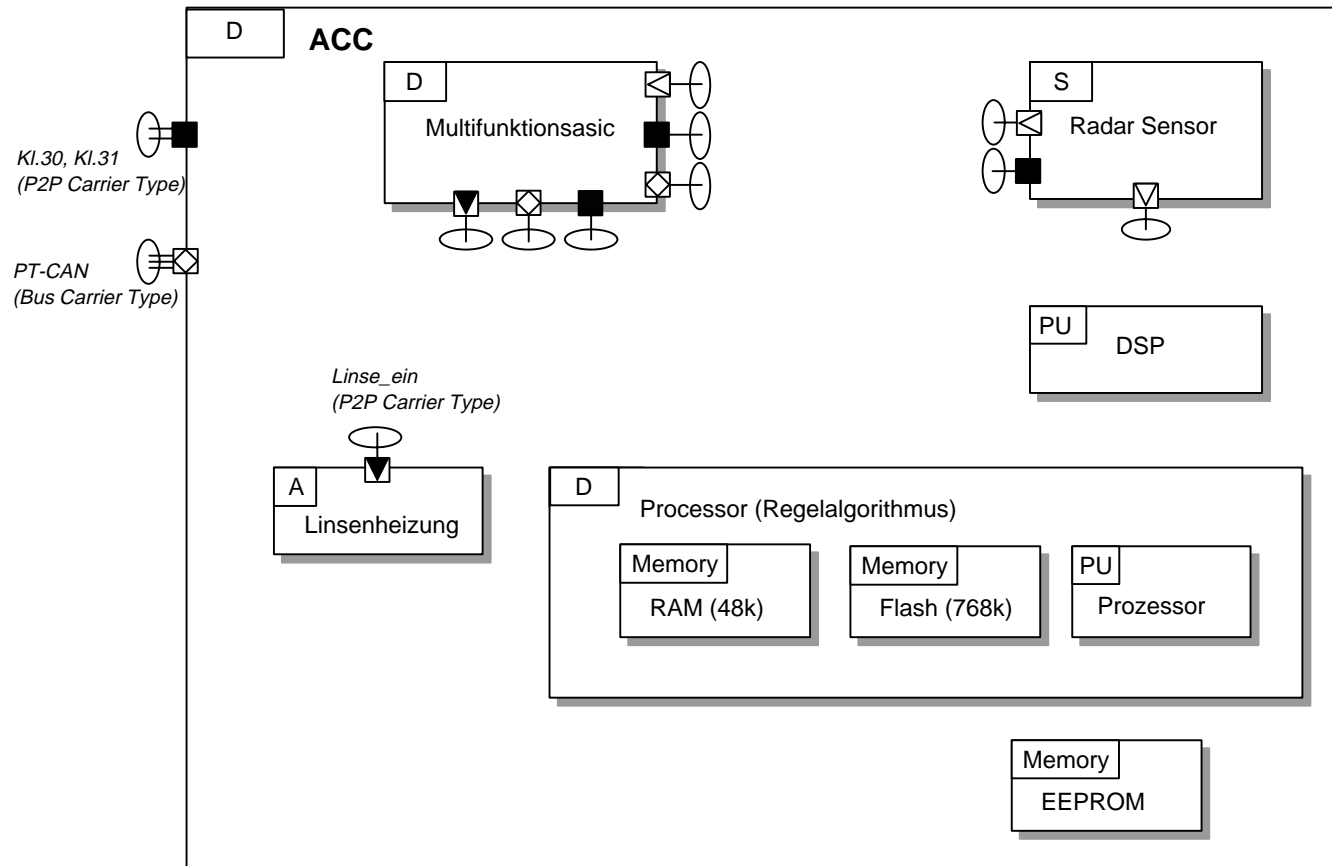
Graphische Notation: Schnittstellenelemente



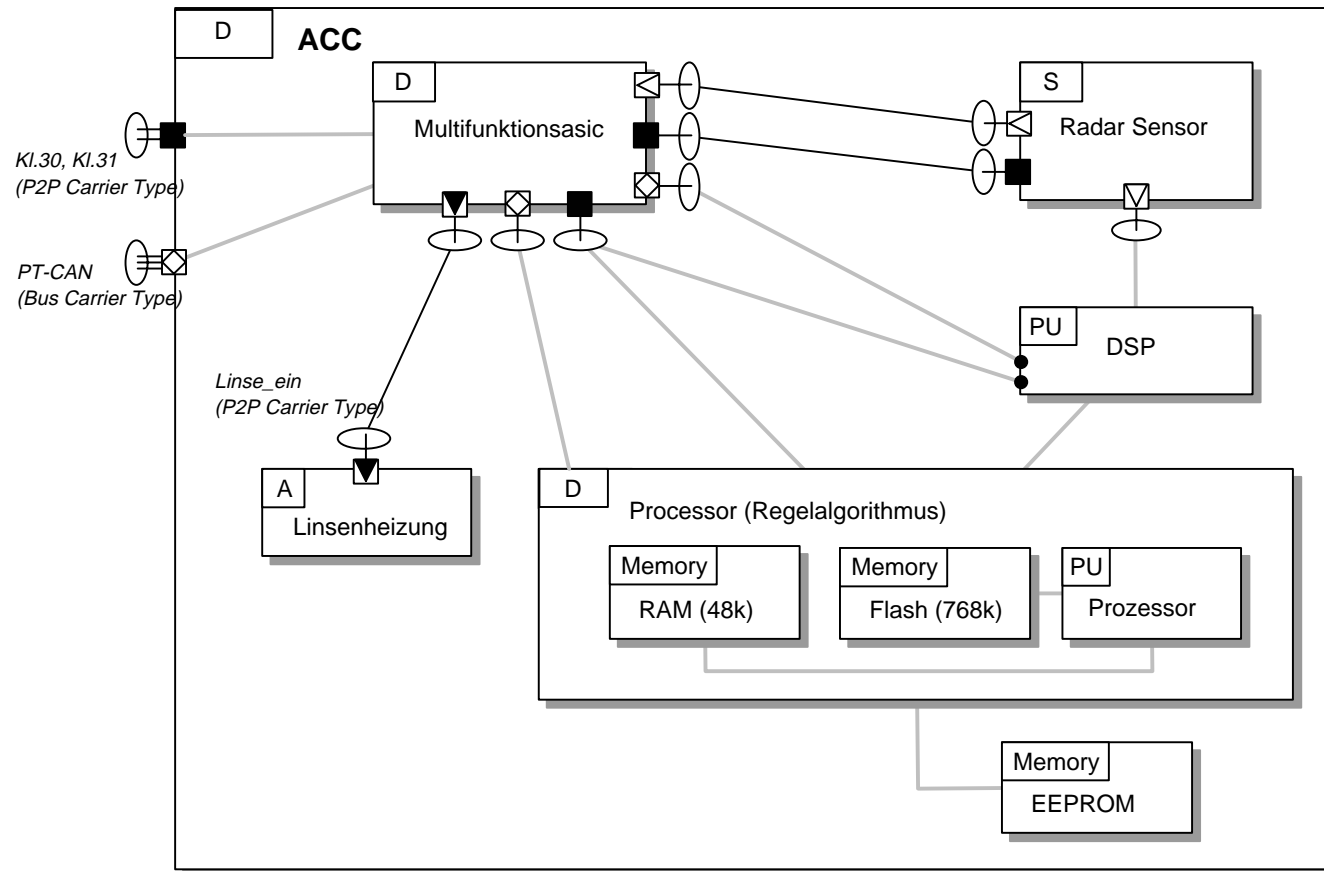
Verbindungen



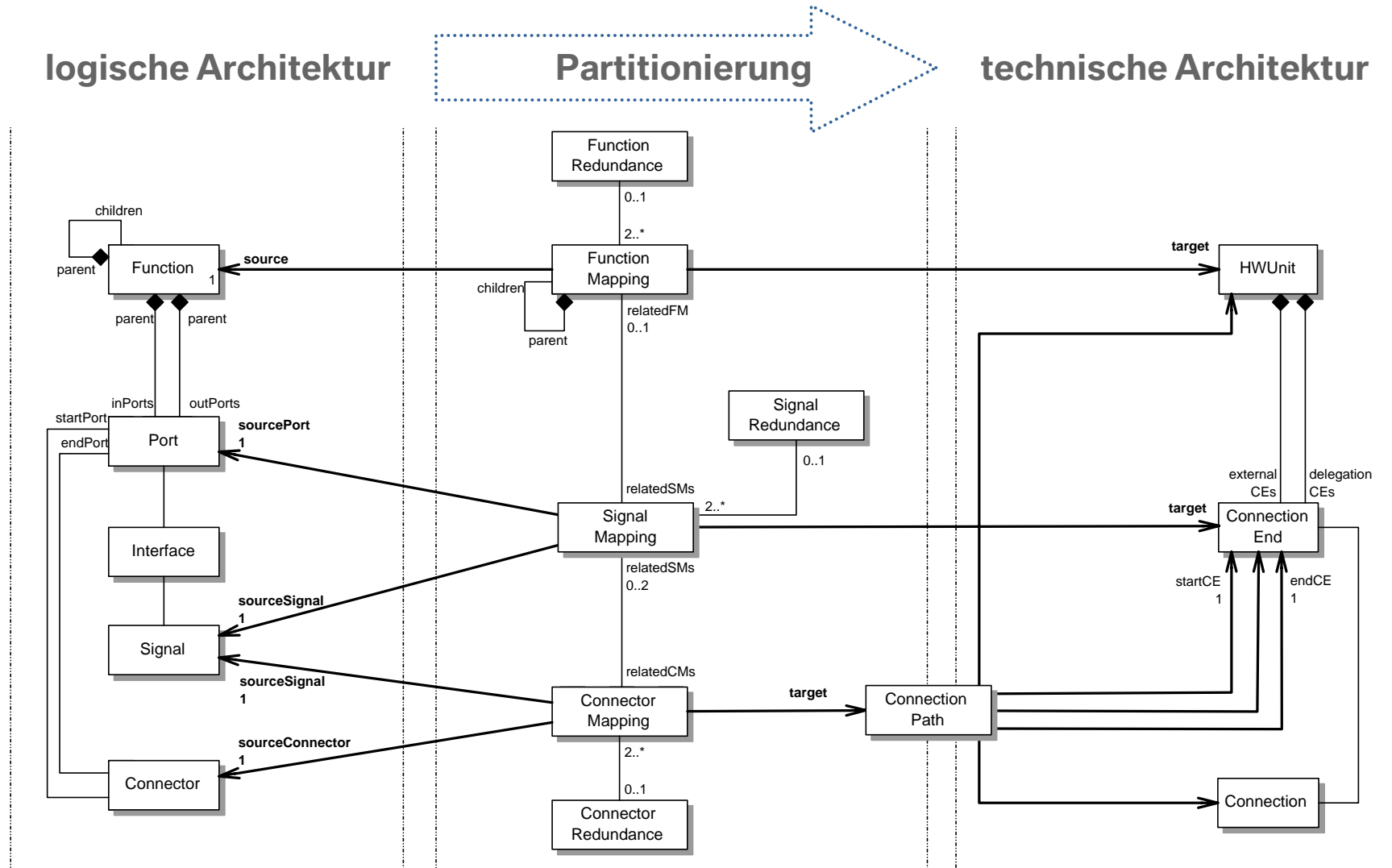
Graphische Notation: Carriertypen



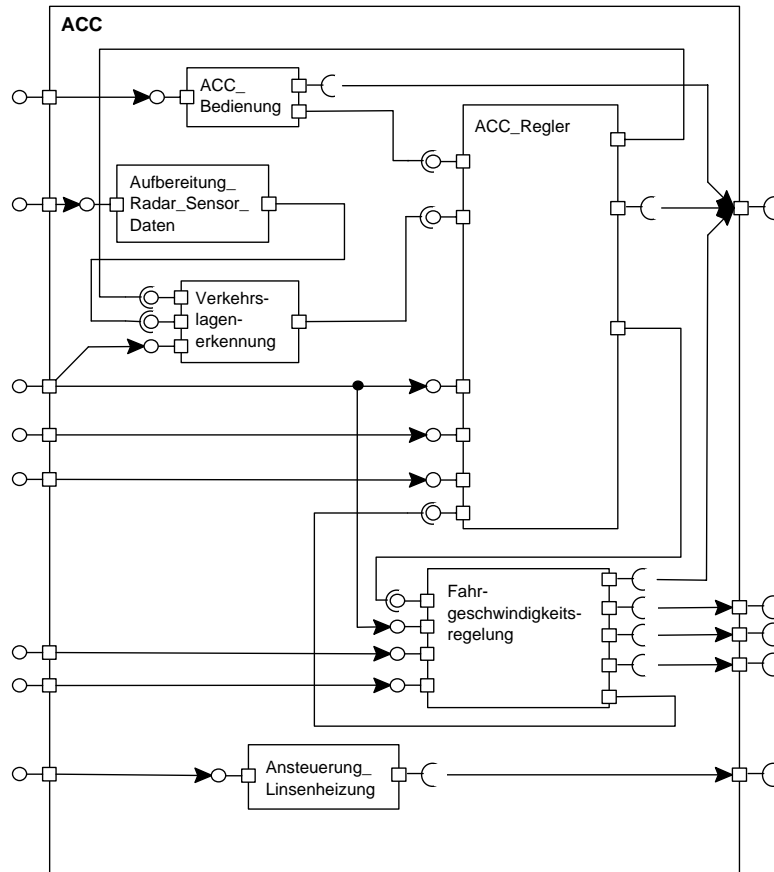
Graphische Notation: Verbindungen



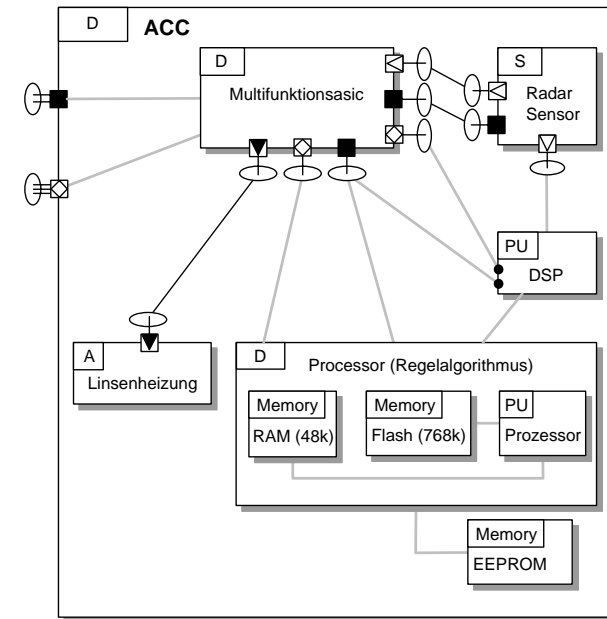
Deployment: Übergang von der logischen Architektur auf die technische Architektur. Dies wird auch Partitionierung genannt.



Partitionierungsbeispiel: ACC

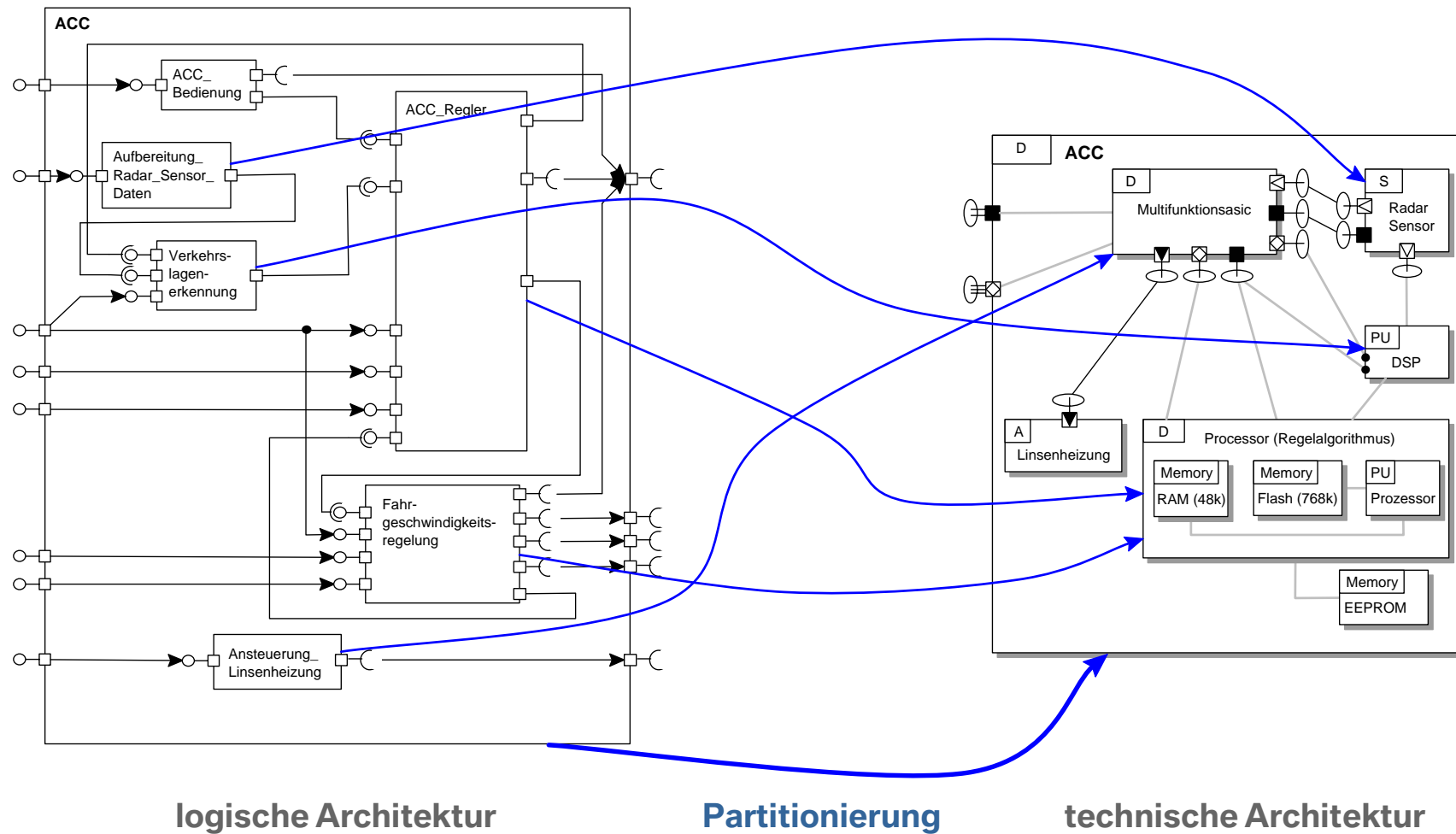


logische Architektur

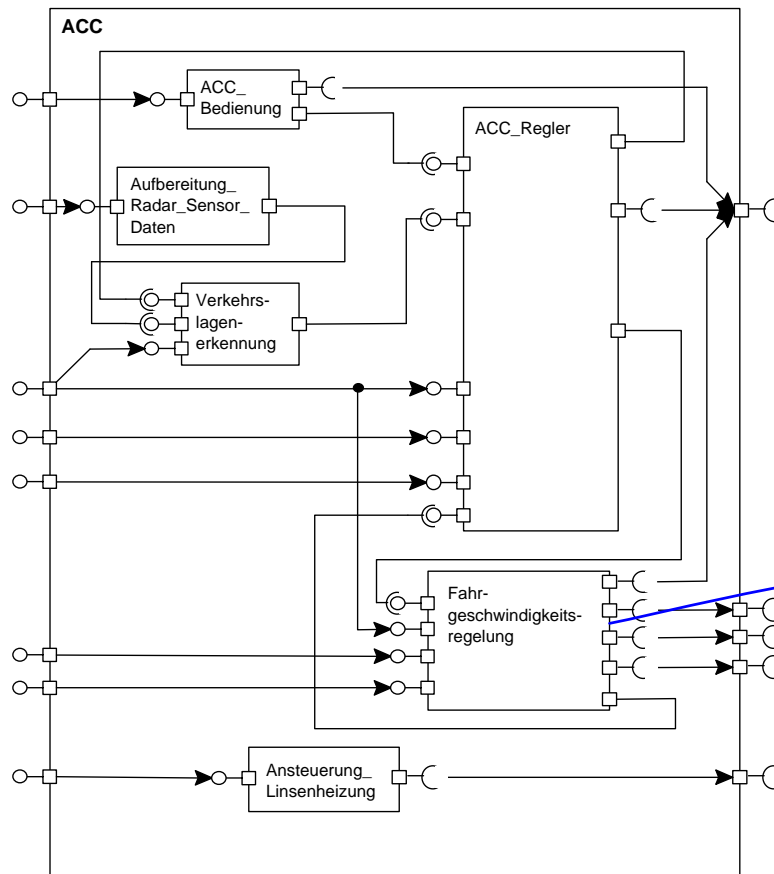


technische Architektur

Einfache Funktionspartitionierung (ACC- Beispiel)

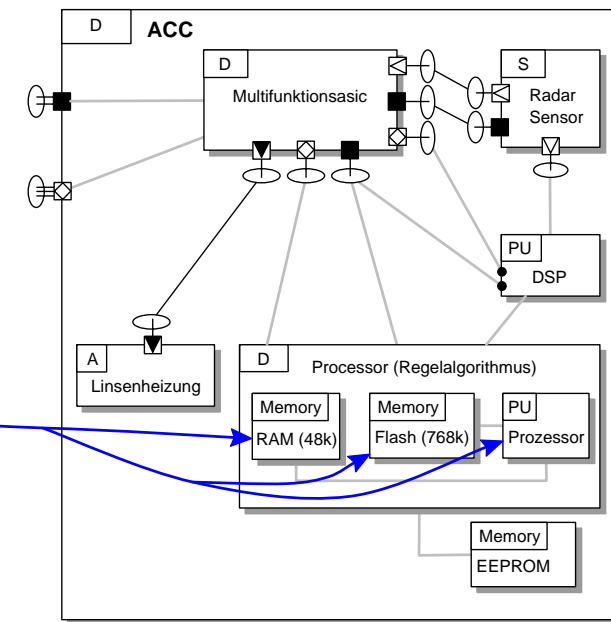


Verteilte Funktionspartitionierung (ACC- Beispiel)



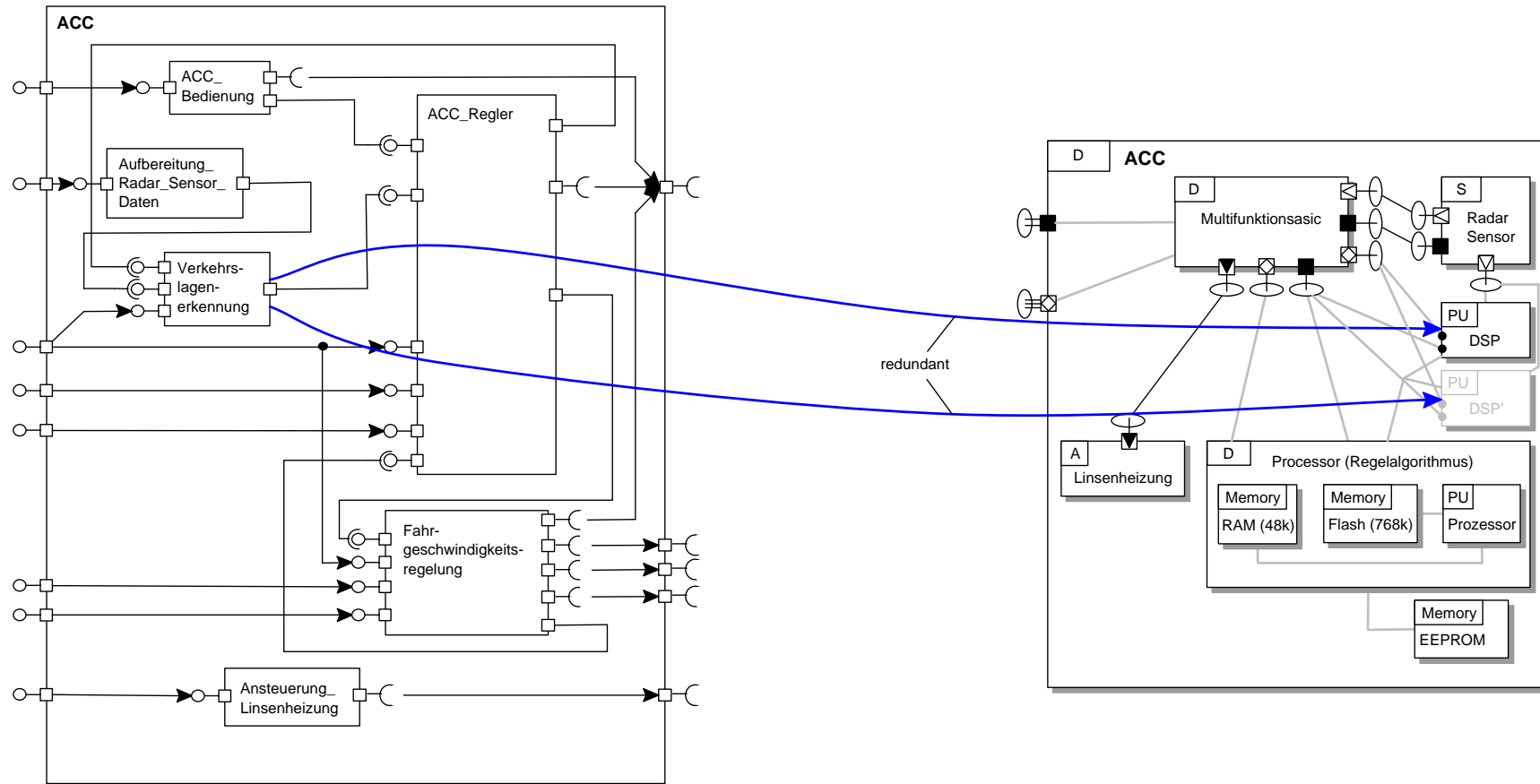
logische Architektur

Partitionierung



technische Architektur

Redundante Funktionspartitionierung (ACC-Beispiel)

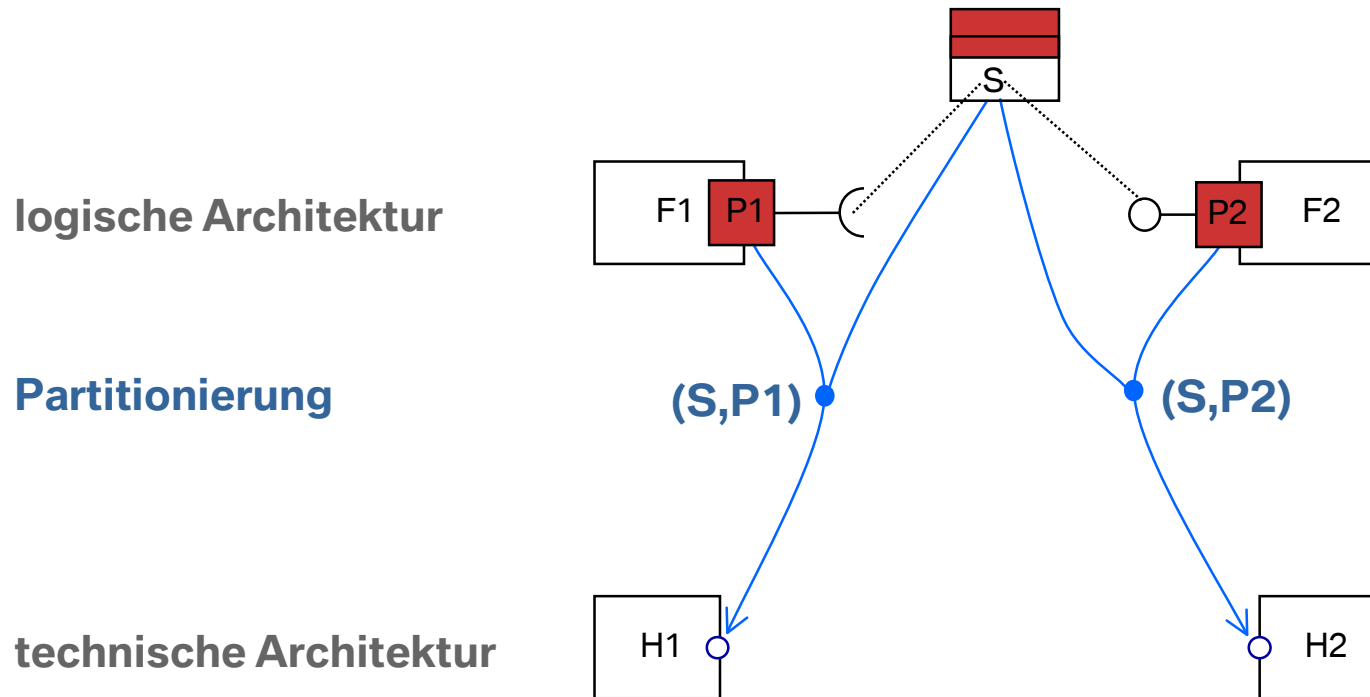


logische Architektur

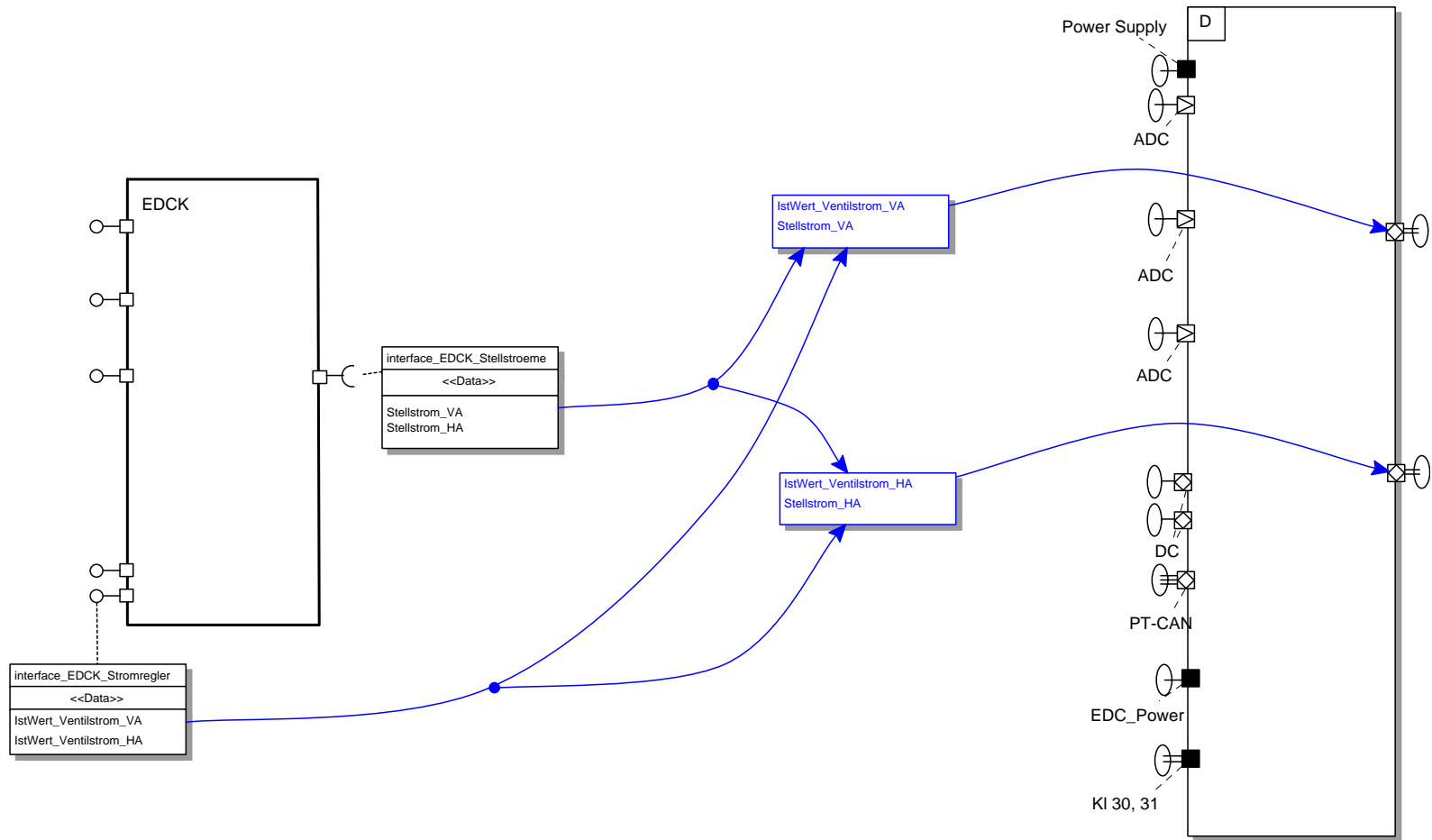
Partitionierung

technische Architektur

Partitionierung von Ein- und Ausgangssignalen



Ein- und Ausgangssignalpartitionierung (EDCK-Beispiel)*



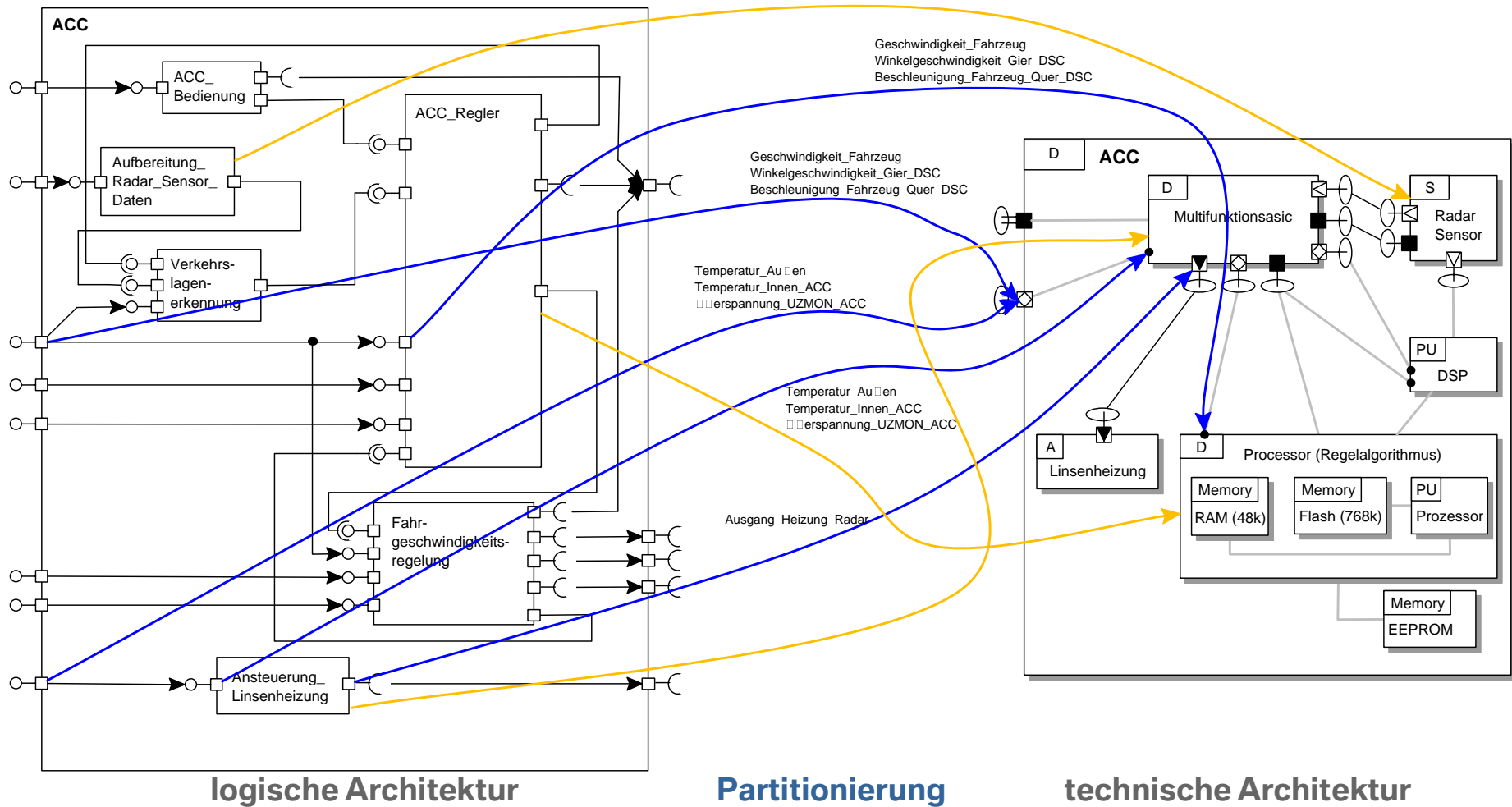
logische Architektur

Partitionierung

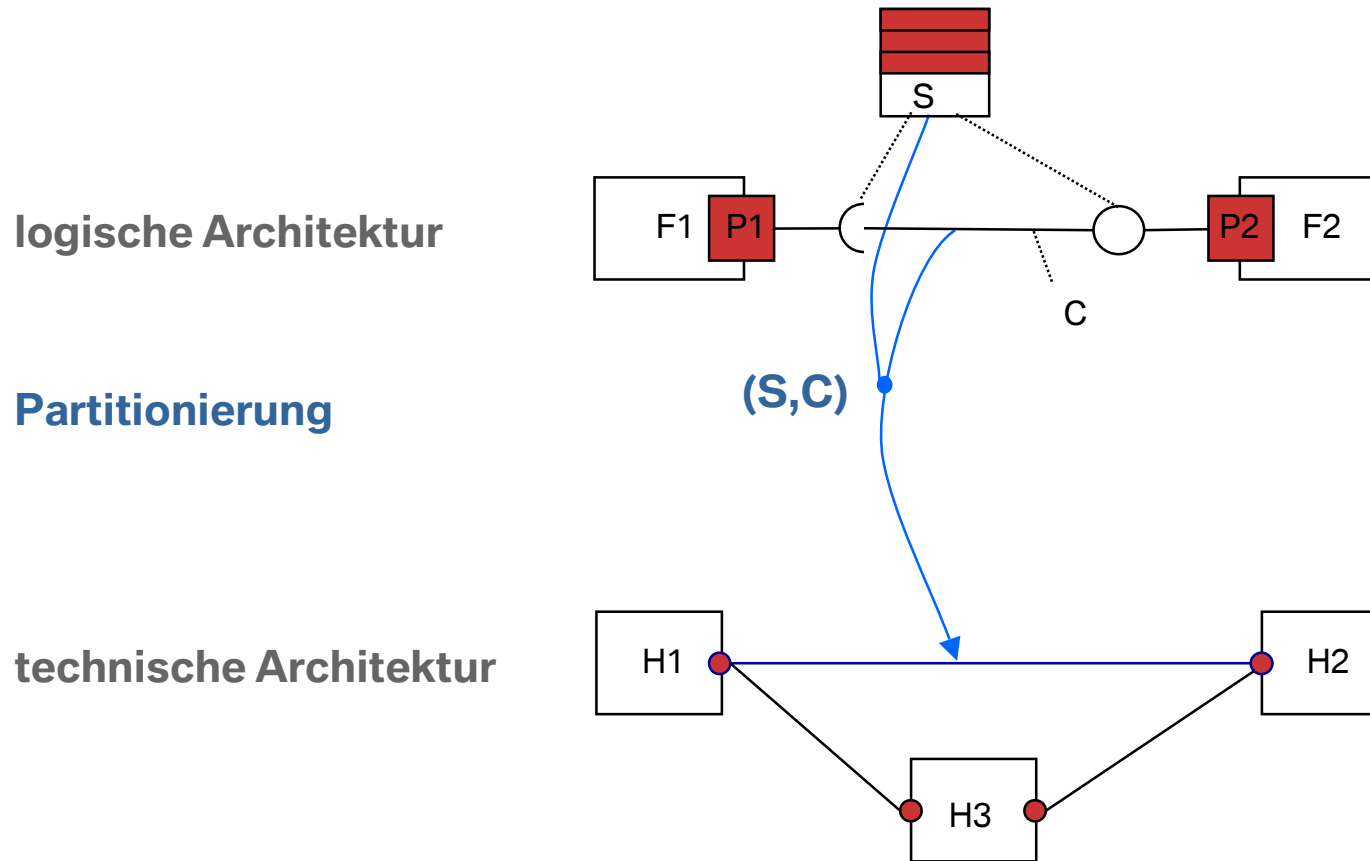
technische Architektur

* Elektronische Dämpfer Kontrolle Kontinuierlich

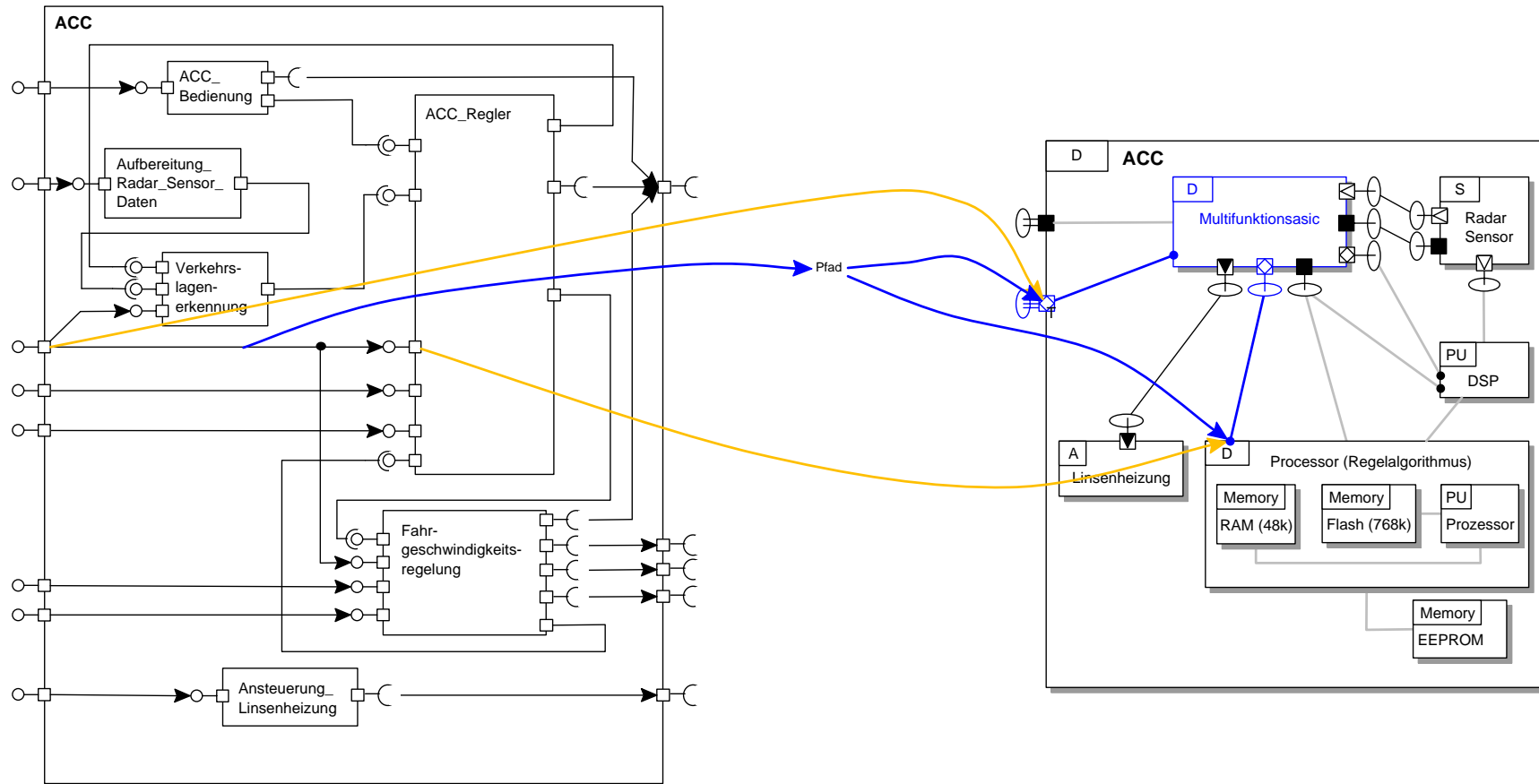
Ein- und Ausgangssignalpartitionierung (ACC-Beispiel)



Partitionierung von logischen Verbindungen



Partitionierung von logischen Verbindungen (ACC-Beispiel)



logische Architektur

Partitionierung

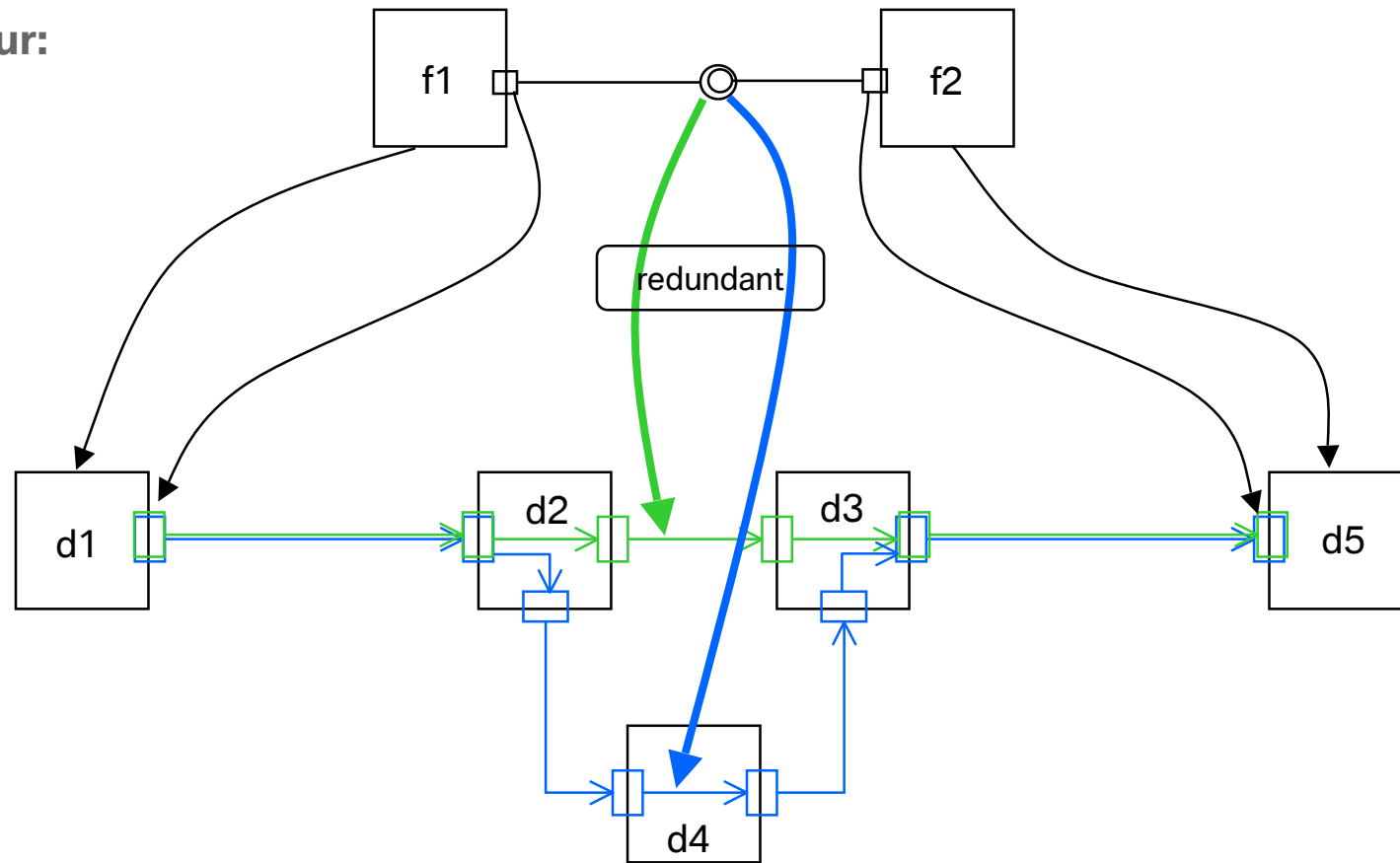
technische Architektur

Beispiel für eine redundante Verbindungspartitionierung

logische Architektur:

Partitionierung:

technische
Architektur:

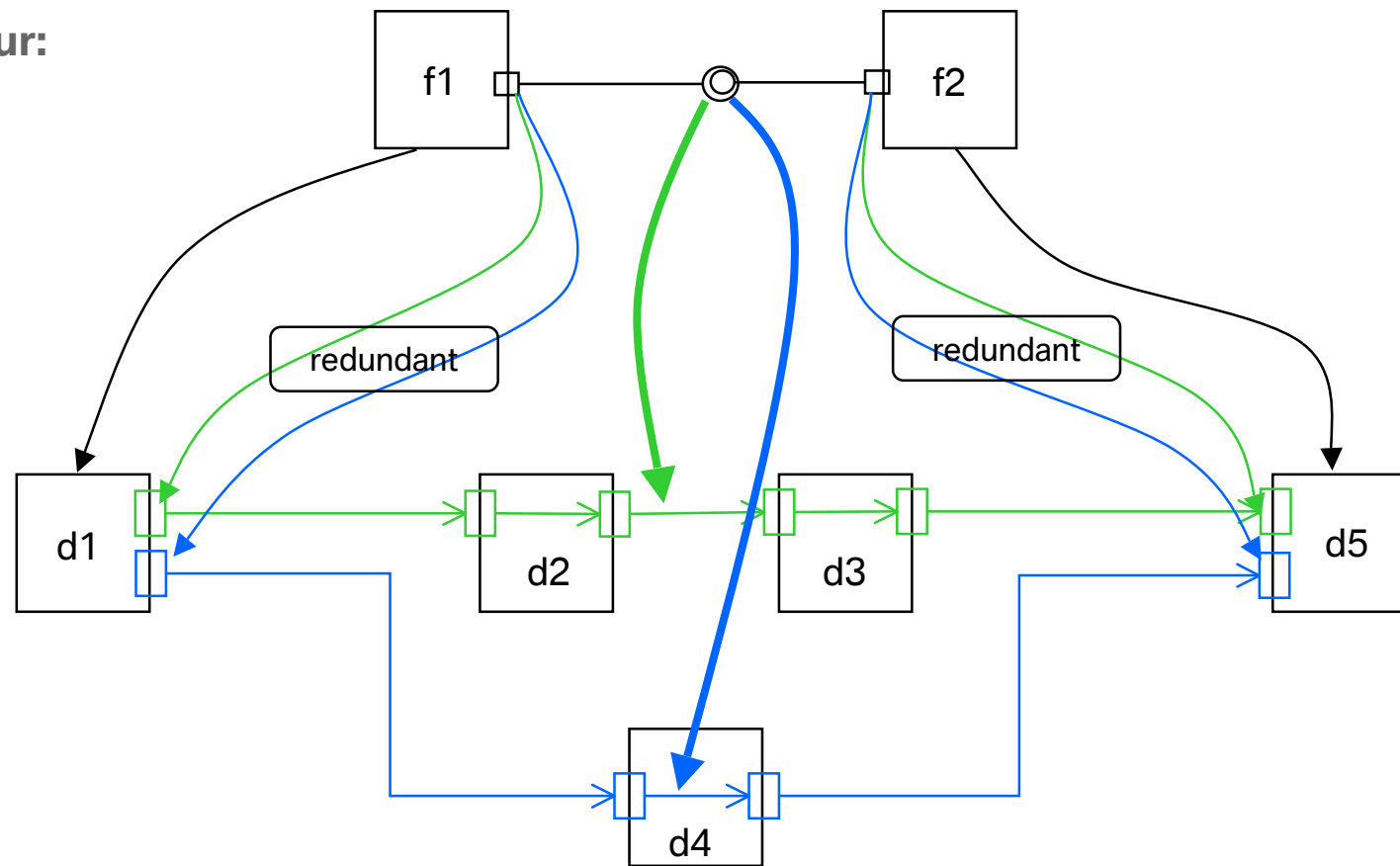


Beispiel für eine implizit redundante Verbindungspartitionierung auf Grund von redundanten Signalabbildungen

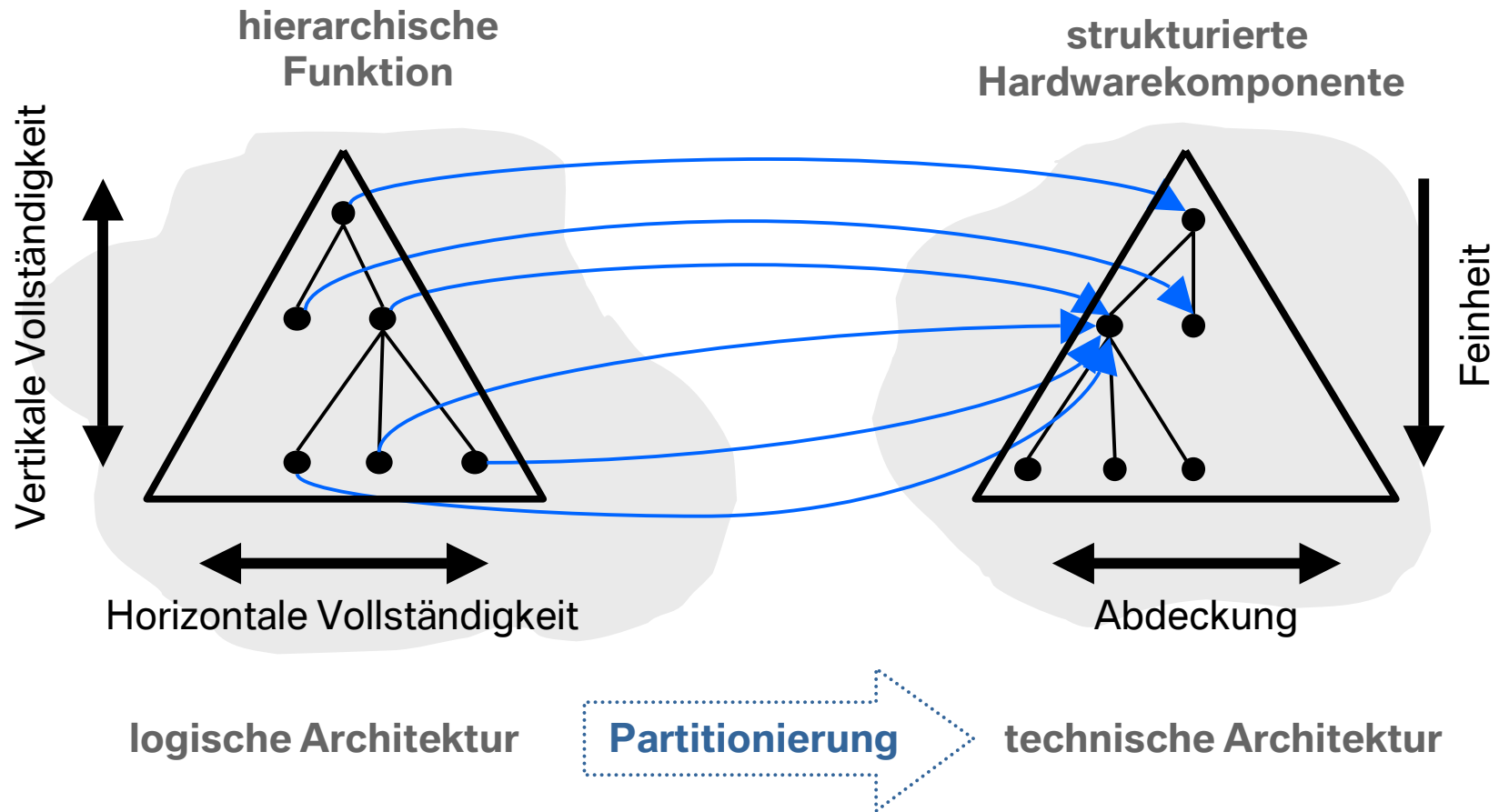
logische Architektur:

Partitionierung:

technische
Architektur:



Bewertung von Partitionierungen



Partitionierungskriterien

Parameter der Bordnetz-Entwicklung

Produktphilosophie

- Kundenzufriedenheit
- Zuverlässigkeit
- Qualität

Produktausführung

- Gleichteile
- Kosten
- Funktionsinhalte
- Modularität

Materialfluss

- Logistik
- Just in time
- Produktionsdesign
- Montage
- Fertigung
- Stückzahl
- Teileanlieferung

Variante, Stückzahlvolumen

- Reparaturfreundlichkeit
- Diagnose
- Gewicht

Geometrische Vernetzung

- Belastung durch
- Einbauraum
- Crashfestigkeit
- Gestaltungsfreiheit
- Stil
- Package

Vernetzung mit Zulieferern

- Lieferantenbewertung
- Entwicklungsrisiko
- extern / intern
- Aufteilung

Signalvernetzung

- technisches Risiko
- Bordnetzarchitektur
- Signalfluß
- Busdatenrate
- Bordnetzkommunikation
- Kabelbaum
- Steuergeräte

Energievernetzung

- Powermanagement
- Energieversorgung
- Steuergeräte
- Kabelbaum
- Sicherungskonzept
- Massekonzept
- EMV

Sonstige Restriktionen

- gesetzliche Vorschriften
- Rahmenbedingungen
- Normen
- BMW Vorschriften