

Software Integration - Erfahrungen und Perspektive.



Dr. Christian Salzmann, BMW AG

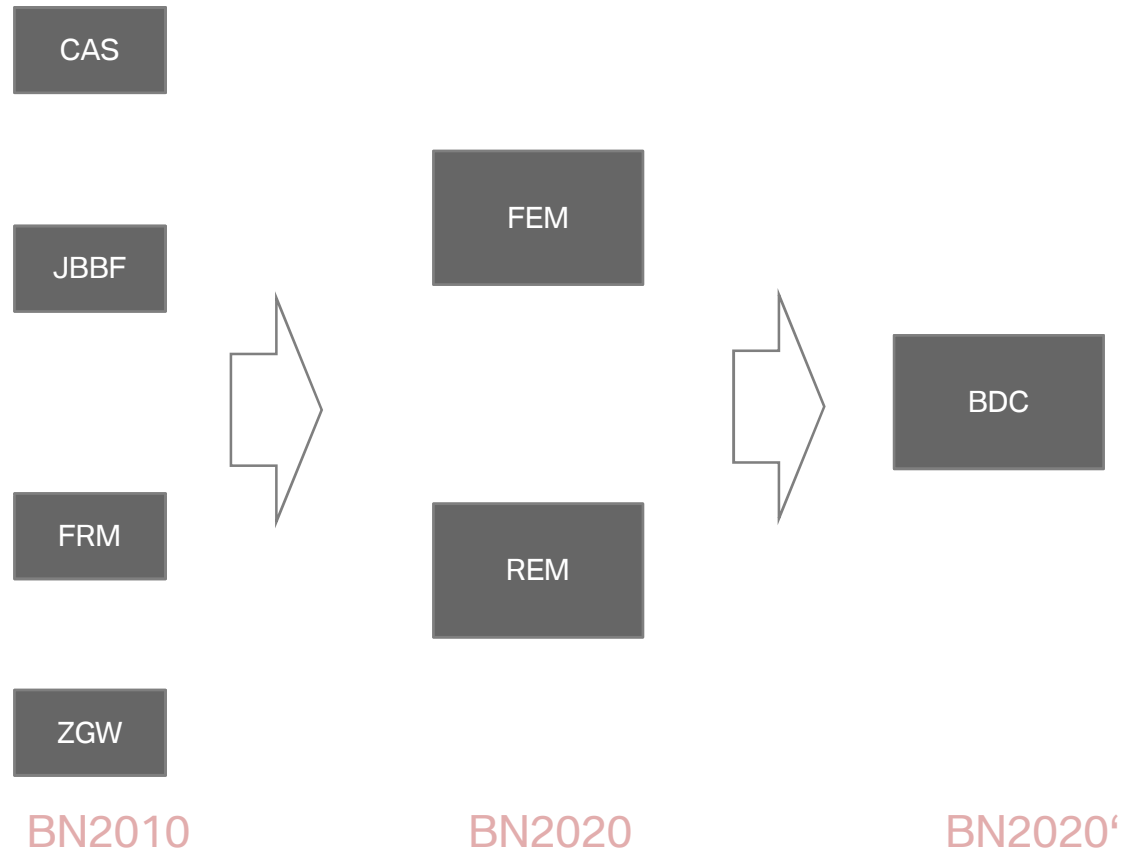
Software Integration.

Inhalt.

- **Warum wird SW Integration so wichtig?**
- **Wie kann SW Integration angegangen werden?**
- **Was sind die Erfahrungen von BMW ?**

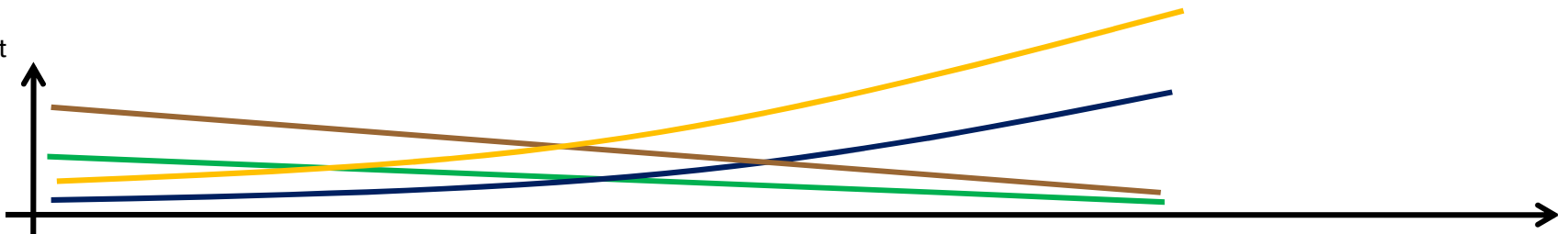
Software Integration.

Beispiel.



SW Komplexität

- Kosten
- Varianten
- Gewicht



Software Integration.

Eintrittsbarrieren erhöhen sich.

Entwicklungskosten



Skills

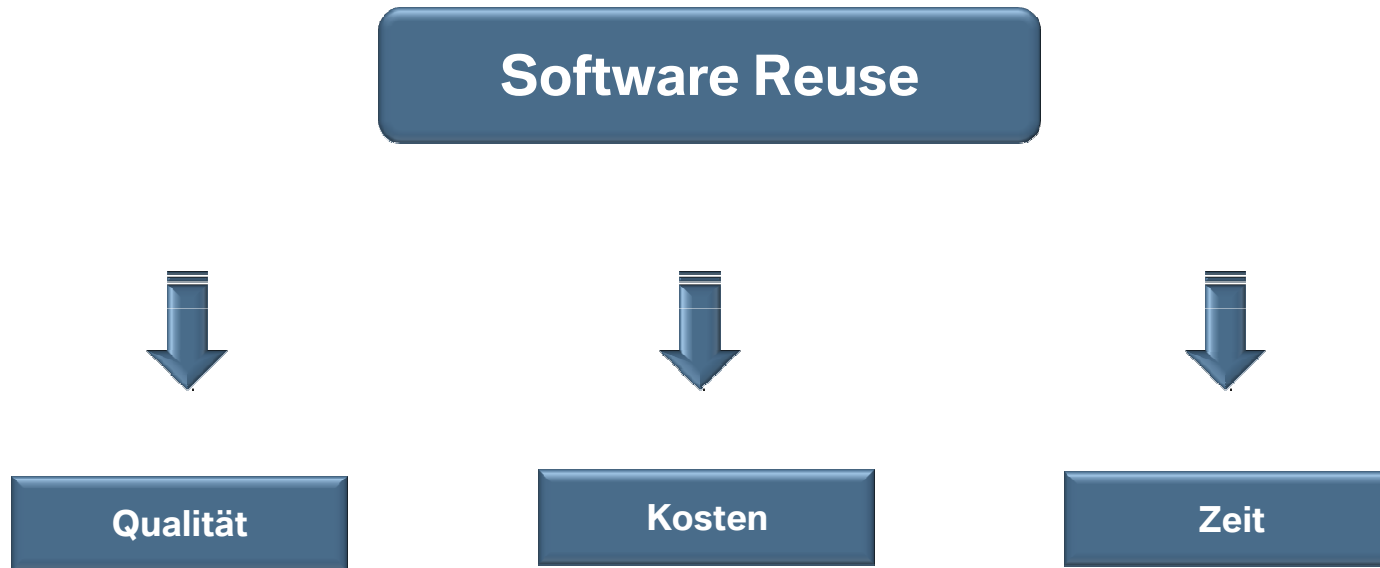


Geschwindigkeit



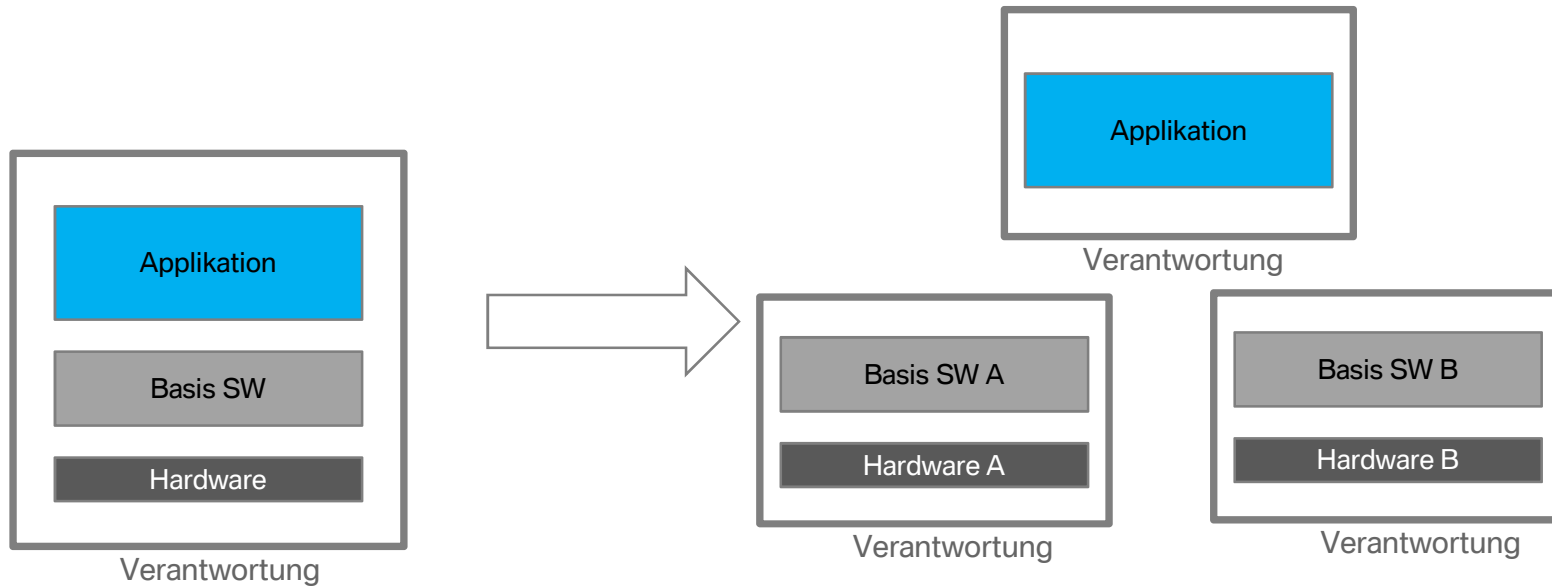
Software Integration.

SW Reuse: Kosten, Qualität, Zeit.



Software Integration.

SW Reuse: Trennung Applikation – Basis SW – HW.



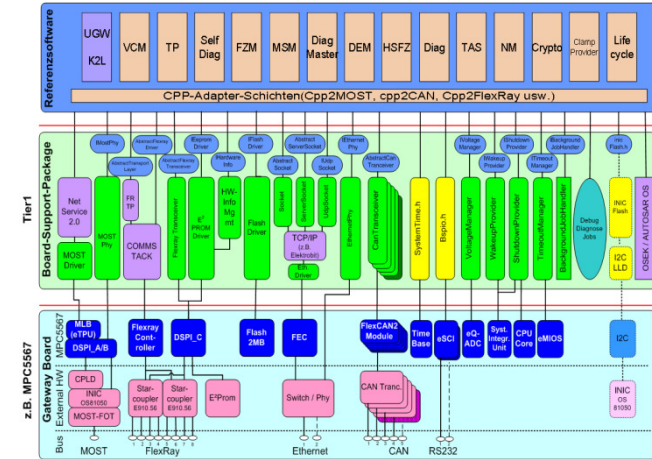
SW Reuse ist Benefit für OEM und Tier 1.

Software Integration.

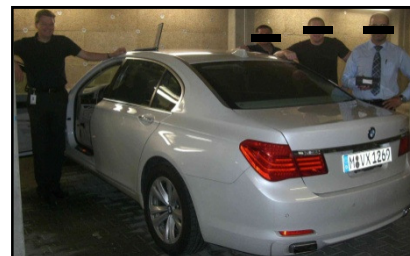
SW Reuse: Zentrales Gateway.

Bereitstellung der ZGW Referenz SW durch BMW an Lieferanten:

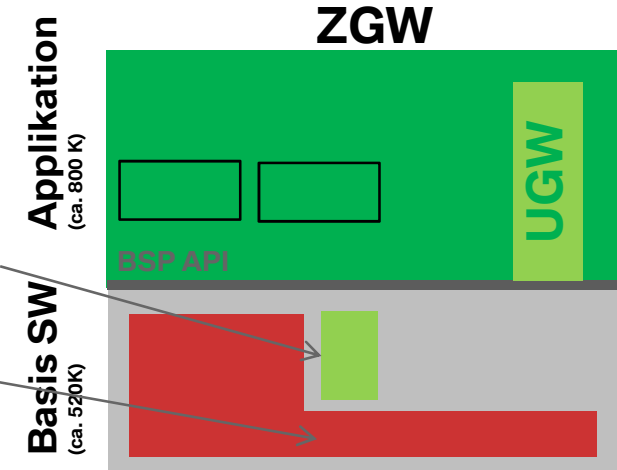
- Im BMW Prozess („Life Software“) parallel zum Serieneinsatz portiert (alle Bugfixes/ Change Requests wurden mitgezogen)
- Eigenständige Integration durch Lieferanten auf Basis der existierenden Dokumentation.



- Während Angebotsphase vollständige Funktionalität in A-Muster
- Unterschiedliche HW
- Applikations-SW Wiederverwendet - Basis SW von Lieferant

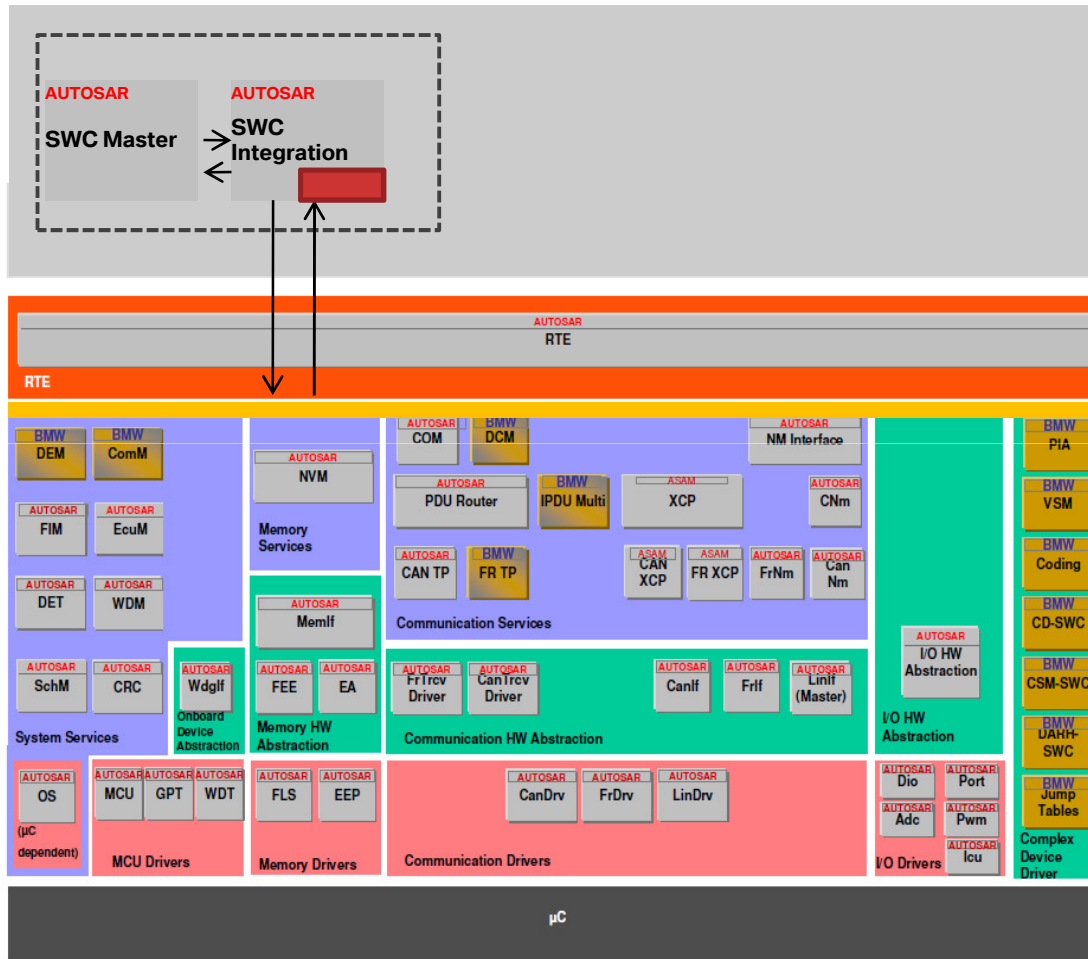


Applikation 800K
 AUTOSAR OS 20K
 Board support Package 500K



Software Integration.

Beispiel: BDC Software.



- > 55 Funktionen
- > 120 SWC
- > 500.000 LoC
- > 1500 Autosar Interfaces
- > 6 Parteien

Application Mode ■ BMW Systemfunktion ■ BMW Kompatibilitätsmodul ■ AUTOSAR / ASAM Standard

Software Integration.

Voraussetzungen.

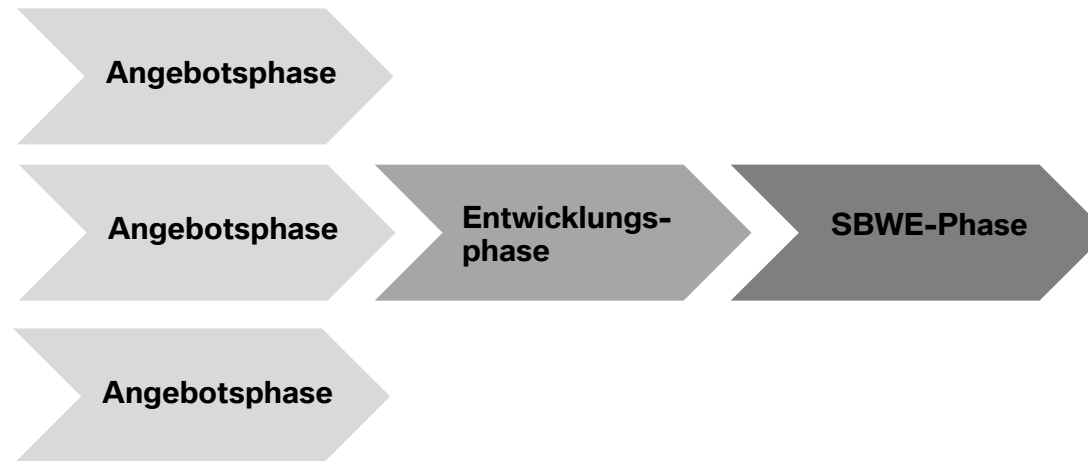
Was

- Zusammenarbeitsmodelle
- SW Technologie
- Tooling
- Methodik

Software Integration.

Voraussetzung: Zusammenarbeitsmodelle.

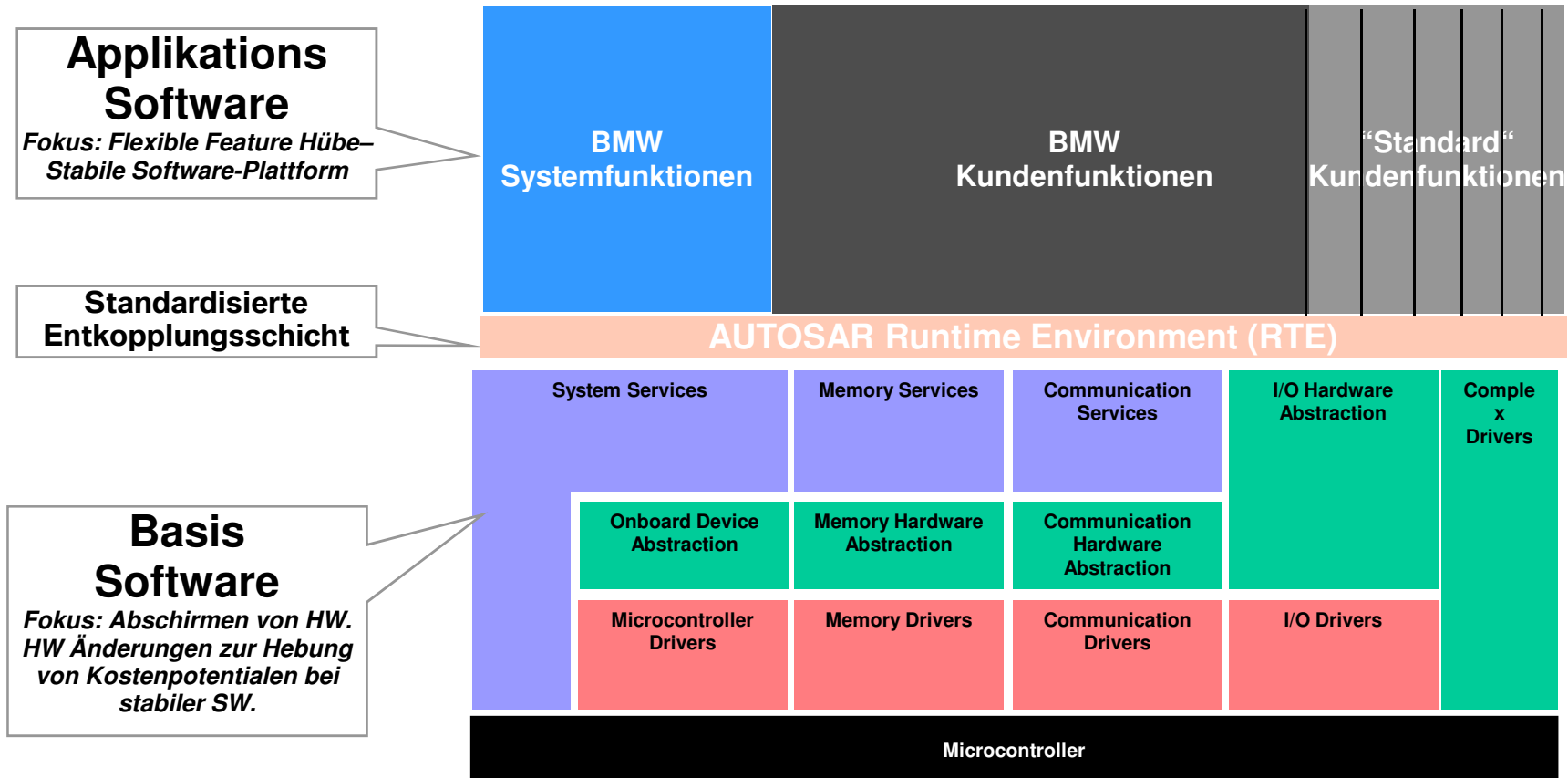
- Softwareanteile von vielen Parteien
- Software auf unterschiedlichen Basis SW
- Software auf unterschiedlicher Hardware



Wie kann der Lieferant ein System inkl. HW Auslegung anbieten,
wenn er die SW nicht kennt?

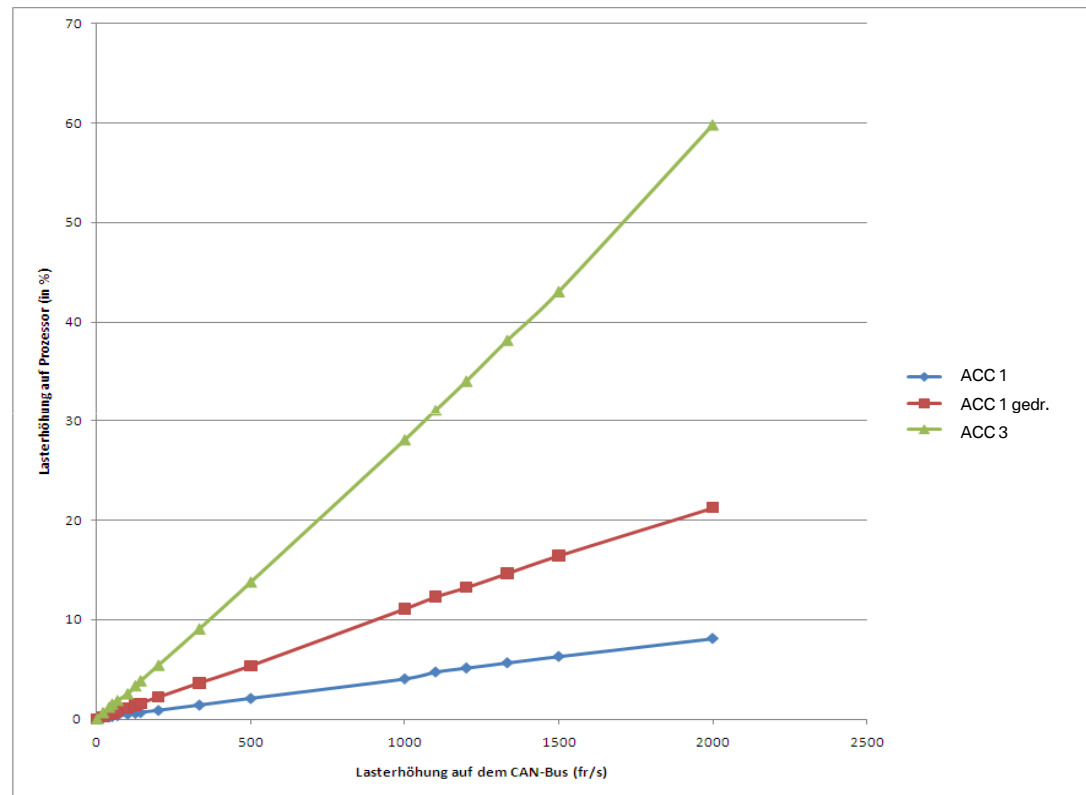
Software Integration.

Voraussetzung: Standardisierte SW Plattform.



Software Integration.

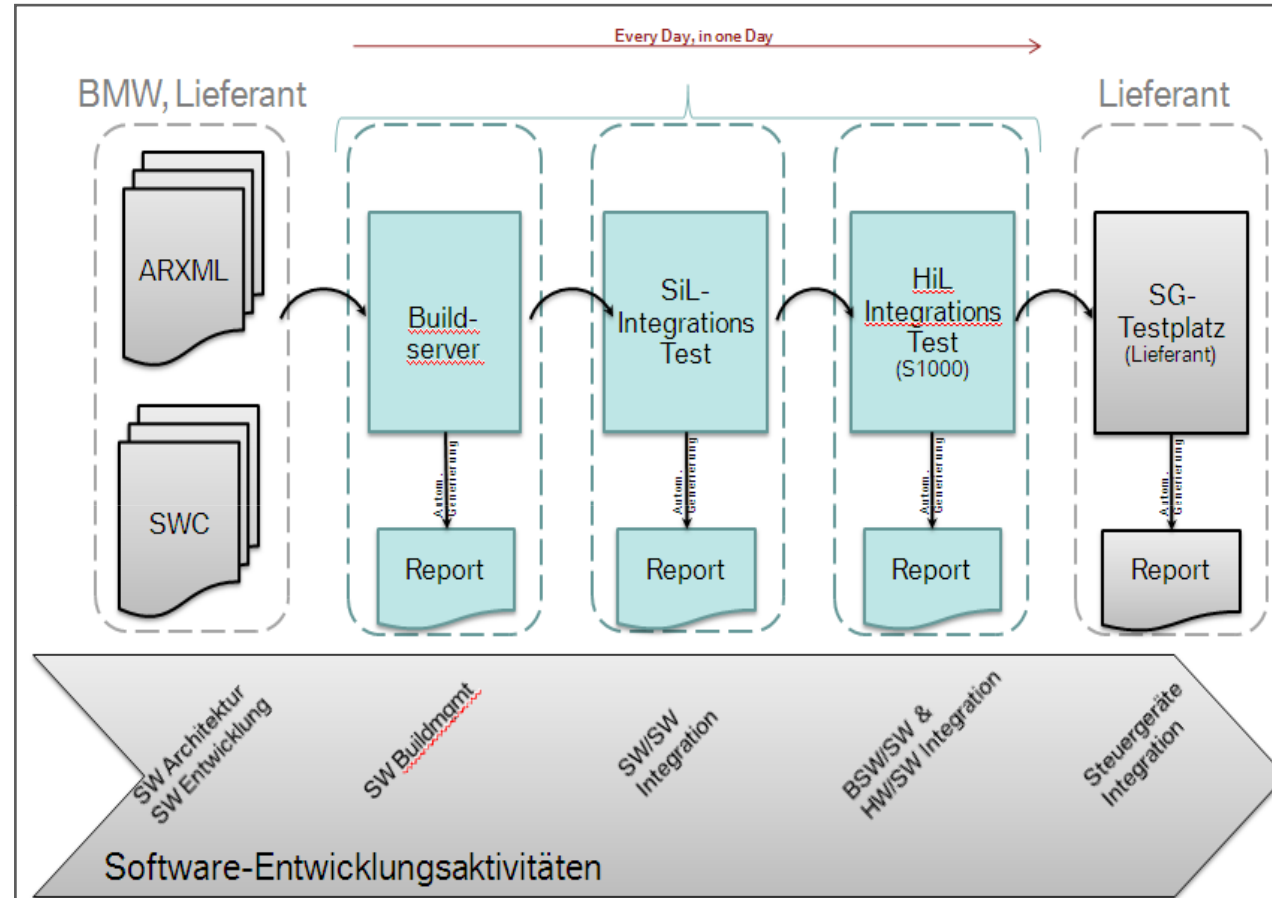
Voraussetzung: Skalierbare Architekturen.



Je grösser ein zu integrierender SW Anteil ist, desto wichtiger ist die Skalierbarkeit der Plattform.

Software Integration.

Voraussetzung: Methodik zur Integration.

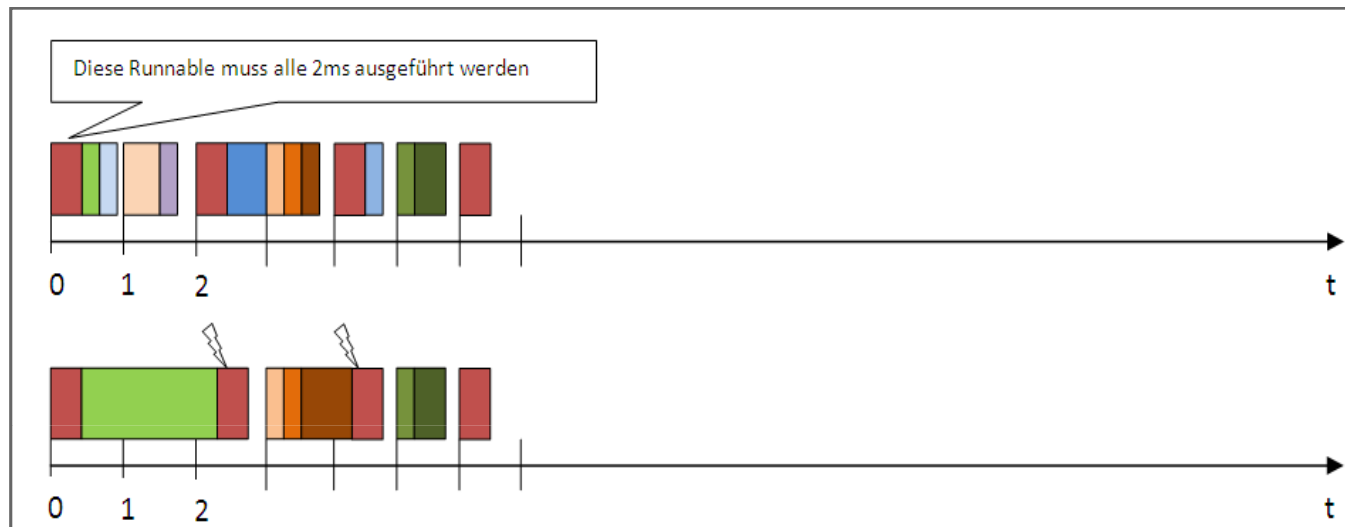


- Statische Prüfung der Signaturen
- Statische Prüfung der Architektur

Dynamische Integrierbarkeit?

Software Integration.

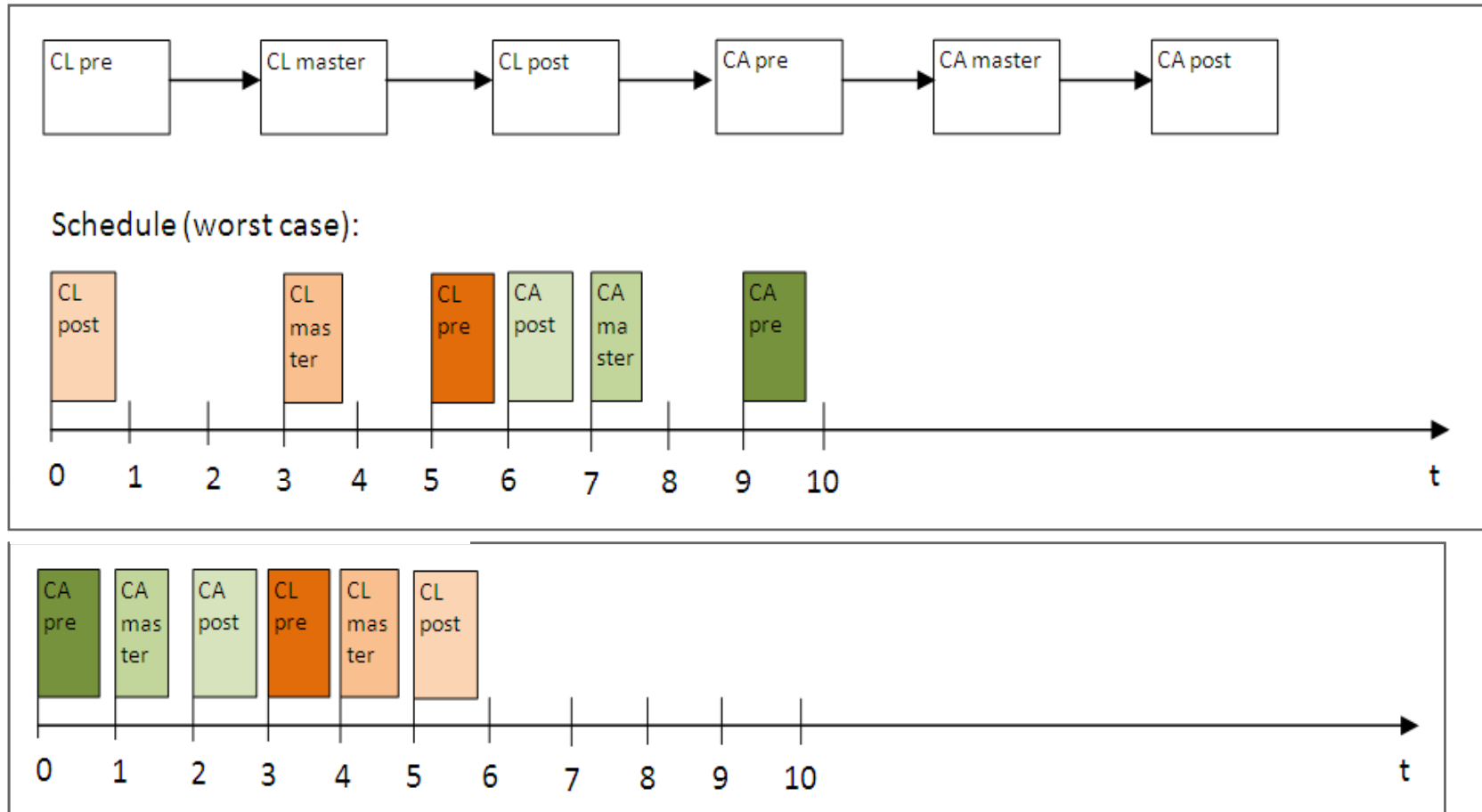
Problemstellung 1: Verletzung von Zykluszeiten / Jitter-Anforderungen



- Bei hochintegrierten Steuergeräten sehr heterogene Anforderungen an Zykluszeit und Jitter.
- Um diesen Schedule zu finden und zu pflegen sind Informationen über die (netto-)Laufzeit erforderlich.
- Die Anforderung der kleinsten Zykluszeit im System hat außerdem Einfluss auf die Anforderung an die Laufzeit aller Runnables!

Software Integration.

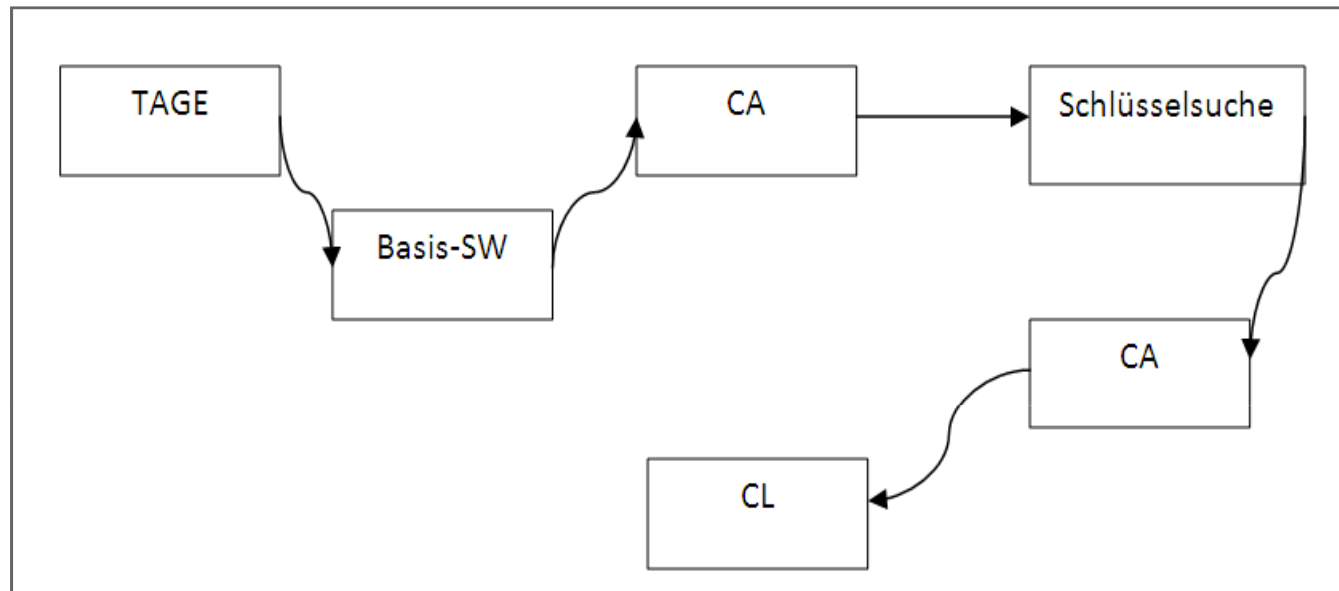
Problemstellung 2: Interaktions-Optimierung.



- Kausalitätsbeziehung zwischen Runnables.
- Zyklizität kann bei ungünstiger Reihenfolge zu sehr viel „Verschnitt“ führen:
Ein Signaldurlauf benötigt hier $8+9+9+8+7=41\text{ms}$
- Optimierungspotential!

Software Integration.

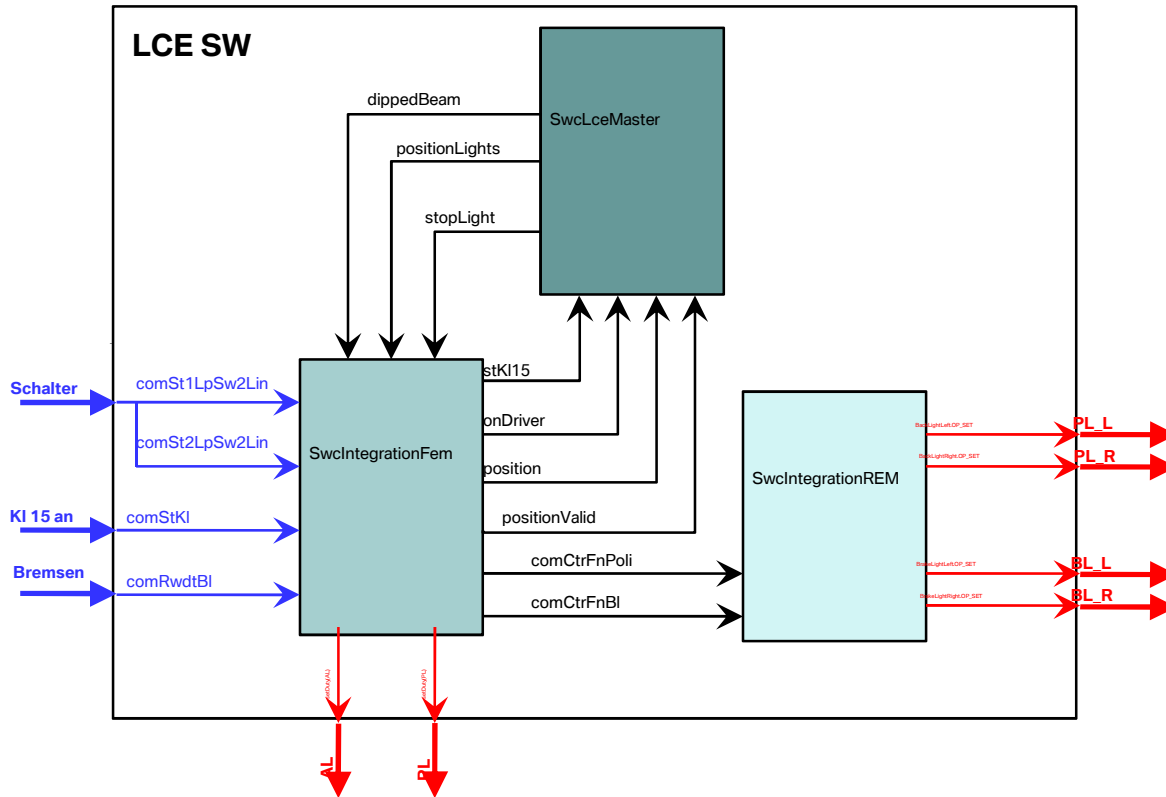
Problemstellung 3: Vergabe von Zeit-Budgets.



- Zeitliche Anforderungen werden oft auf Funktions/Usecaseebene gestellt.
- Unabhängig von Software Design und -Partitionierung
- Notwendig ist ein Herunterbrechen auf SW Komponenten bzw. Runnable Ebene.

Software Integration.

Verwendung: Funktion - SWC

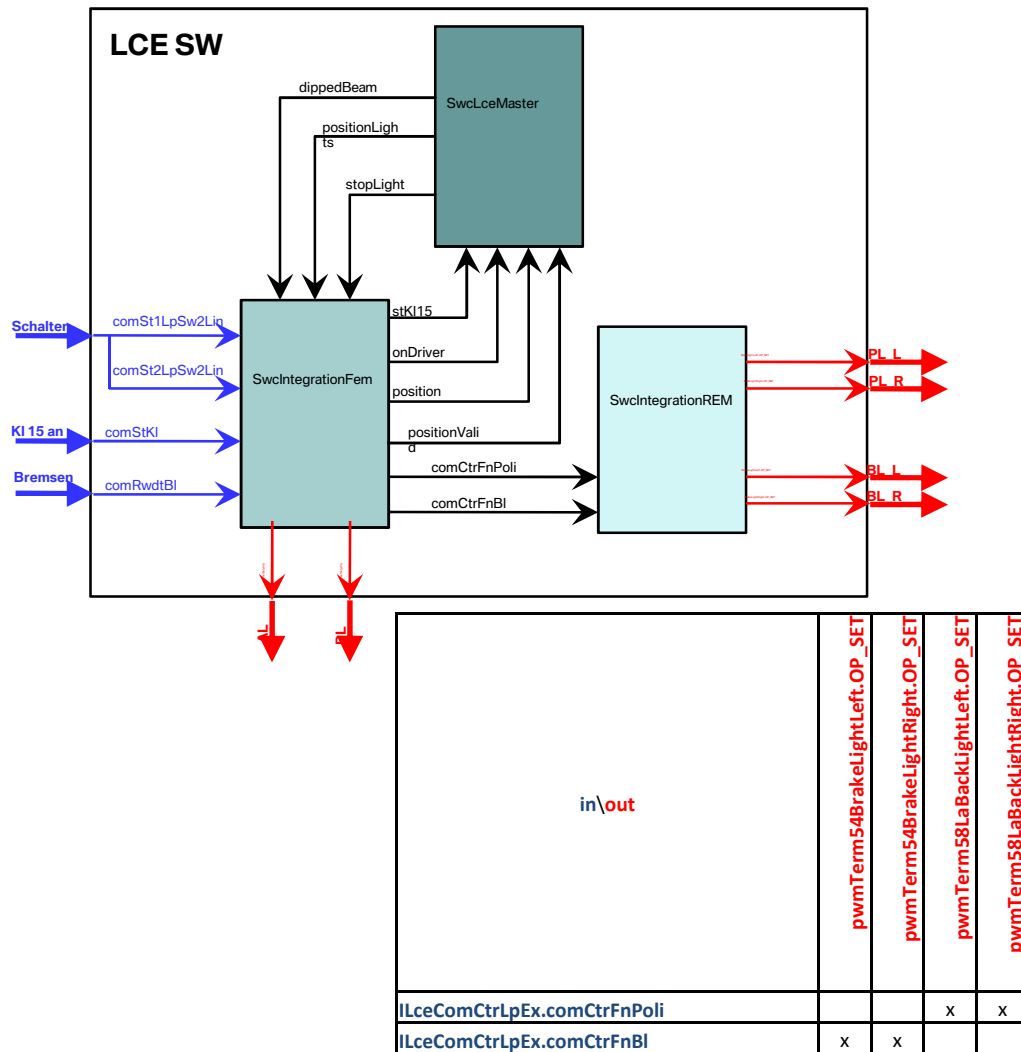


in\out	Ablendlicht ansteuern	Positionlicht ansteuern	Positionlicht hinten links	Positionlicht hinten rechts	Bremslicht links	Bremslicht rechts
Schalter auf Position AL	X	X	X	X		
Klemmen 15 an	X	X	X	X	X	X
Bremse betätigt					X	X

in\out	ILceLampCtrl.setDuty(AL)	ILceLampCtrl.setDuty(PL)	pwmTerm58LaBackLightLeft.OP_SET	pwmTerm58LaBackLightRight.OP_SET	pwmTerm54BrakeLightLeft.OP_SET	pwmTerm54BrakeLightRight.OP_SET
ILceComStLpSw2Lin.comSt1LpSw2Lin	X	X	X	X		
ILceComStLpSw2Lin.comSt2LpSw2Lin	X	X	X	X		
ILceComKlemmen.comStKl	X	X	X	X	X	X
ILceComKlemmen.comRwdtBl					X	X

Software Integration.

Verwendung: Decomposition SWC.



in\out	<code>ILceStateLightSwitch.positionValid</code>	<code>ILceStateLightSwitch.positionValid</code>	<code>ILceStateBrake.onDriver</code>	<code>ILceKlemmen.stK115</code>	<code>ILceComCtrlPEx.comCtrlFnPoli</code>	<code>ILceComCtrlPEx.comCtrlFnBl</code>	<code>ILceLampCtrl.setDuty(AL)</code>	<code>ILceLampCtrl.setDuty(PU)</code>
<code>ILceLightsCtrl.(lightsCtrl).dippedBeam</code>							x	
<code>ILceLightsCtrl.(lightsCtrl).positionLights</code>					x			x
<code>ILceLightsCtrl.(lightsCtrl).stopLight</code>						x		
<code>ILceComStLpSw2Lin.(comStLpSw2Lin).comSt1LpSw2Lin</code>	x							
<code>ILceComStLpSw2Lin.(comStLpSw2Lin).comSt2LpSw2Lin</code>		x						
<code>ILceComKlemmen.(comKlemmen).comStK1</code>				x				
<code>ILceComKlemmen.(comKlemmen).comRwdtBl</code>			x					

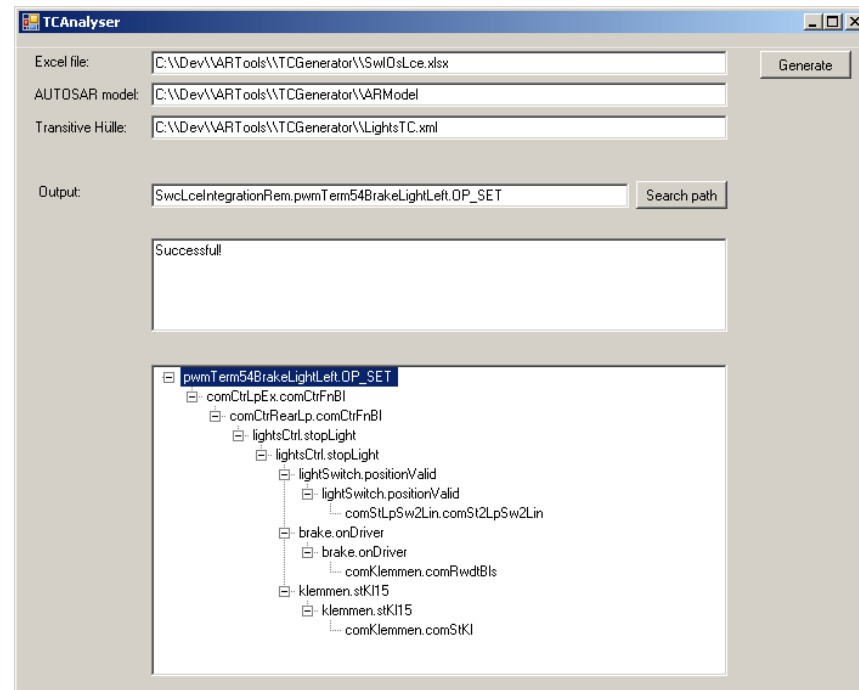
in\out	<code>ILceLightsCtrl.dippedBeam</code>	<code>ILceLightsCtrl.stopLight</code>	<code>ILceLightsCtrl.positionLights</code>
<code>ILceStateLightSwitch.position</code>	x		x
<code>ILceStateLightSwitch.positionValid</code>	x	x	x
<code>ILceStateBrake.onDriver</code>		x	
<code>ILceKlemmen.stK115</code>	x	x	x

Beipackzettel. Verwendung.

Beipackzettel bietet auf einer Stelle eine komplette Beschreibung einer oder mehreren Softwarekomponenten. Außer der Dokumentationszwecken kann diese semiformale Beschreibung auch zur weiteren automatisierten Analyse der SWCs verwendet werden. Beispiele:

- Darstellung des dynamischen Verhaltens von SWCs
- Simulation des Verhaltens von SWCs
- Nachverfolgbarkeit der Kommunikation zwischen SWCs
- Analyse des Impacts von SW-Änderungen
- usw.

Beispiel der Verwendung:
Tool für die Nachverfolgbarkeit der
Signalflüsse innerhalb eines Verbunds
von Softwarekomponenten

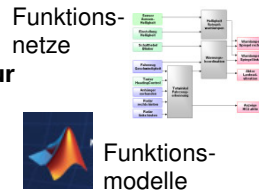


Software Integration. Zielbild AIDA.

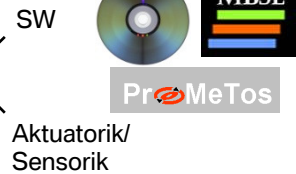
Produktdefinition



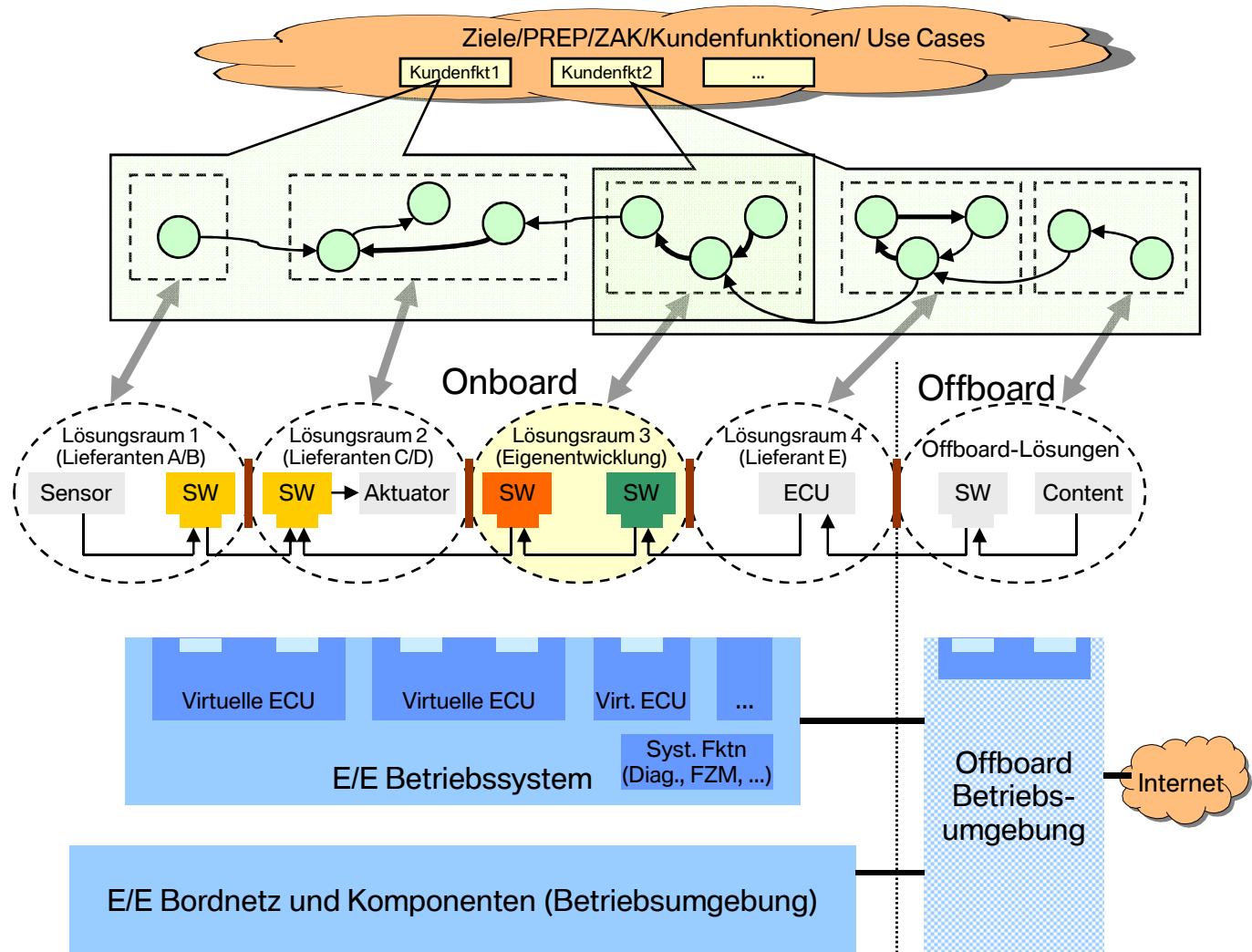
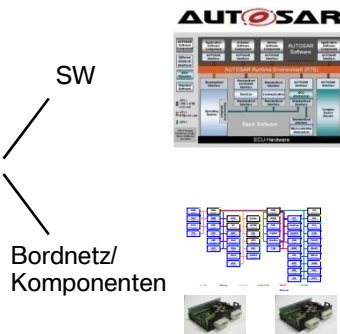
Funktionsarchitektur und -modellierung



Entwicklung



Integration



Software Integration. Summary.

- SW Umfang in Steuergeräten nimmt aus verschiedenen Gründen zu (Funktionszuwachs, Hochintegration, Bauraum ..)
- Die SW wird dadurch zunehmend komplexer, aufwändiger und teurer.
- Zielführend ist eine Trennung von Hard- und Software (Kosten, Qualität, Zeit) um SW Wiederverwendbar zu gestalten.
- Die Anforderungen an das Know How zur SW Integration steigt!

Die Integrationsschnittstelle des OEM erweitert sich von Hardware hin zu Software.

Backup