

Automotive Software Engineering.

Zentralübung Infotainment.



BMW Car IT GmbH



Zentralübung Infotainment. Überblick.

1. Infotainment Hardwarearchitektur
2. Software-Infrastruktur des AS L
3. Eine Software-Architektur für Infotainment-Lösungen
4. CDC hanger - Ein Beispiel

Zentralübung Infotainment.

Technische Umsetzung: Headunit.

Zentrales Steuergerät für

- Mensch-Maschine-Interface → MMI
- Entertainment
- Fahrerinformationen
- Kommunikation



Zentralübung Infotainment. Hardwarekomponenten.

ECUs (Steuergeräte)

- Head Unit
- SingleCDPlayer
- CDChanger
- Amplifier
- TCU (Telefon)
- CAN-Steuergeräte
 - Fahrerassistenzsysteme
 - iDrive

Headunit

- Realisierung von Anzeige- und Bedienkonzept
- Größter Anteil der Infotainment-Software
- Most-Master
- Gateway-Funktionalität

Automotive Systems Lab.

Plattform für Infotainment Prototypen.

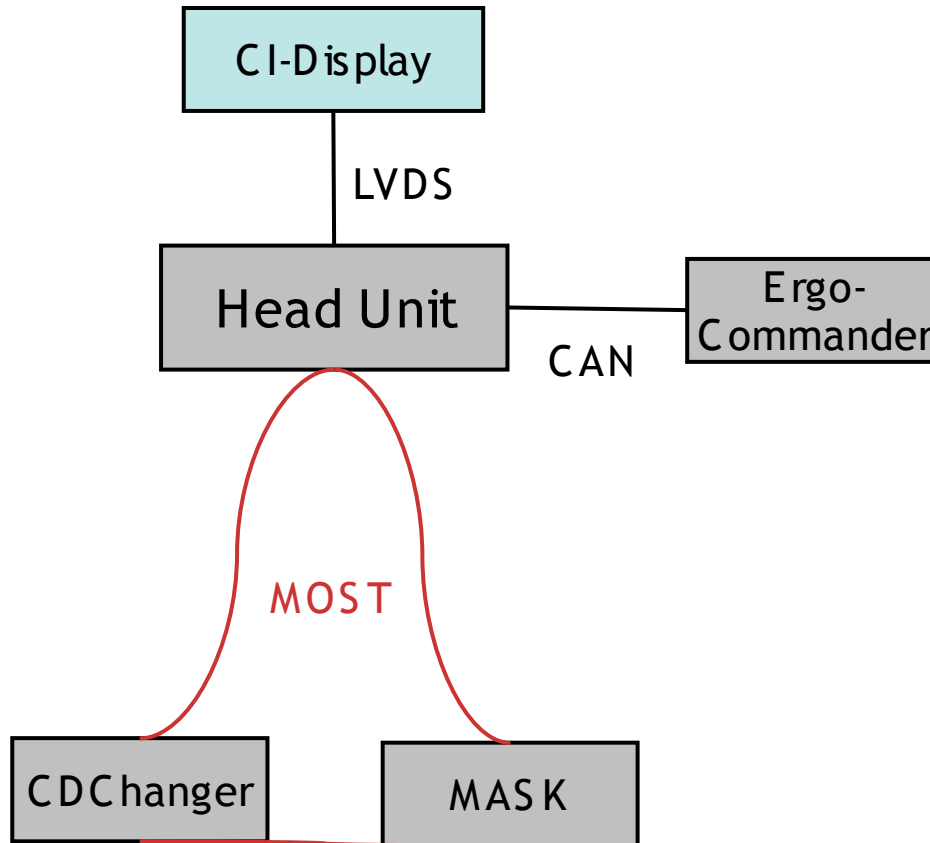
Ursprung des ASL (Automotive Systems Lab)

- Projekt MEWADIS (<http://www4.in.tum.de/~mewadis>)
- Basis für Automotive Prototypen
- Verwendung von Serien-Steuergeräten

Offene Architektur

- Schnittstellen zu Display und Steuergeräten
- Basierend auf Java
- Verwendung von OSGi

Automotive Systems Lab. Hardwarearchitektur.



Automotive Systems Lab.

Infotainment Software-Infrastruktur.

Embedded RTOS

- QNX
- Photon (Graphic Library)

Embedded Java

- J9 (IBM)

Mostzugang

- J2Most (BMW Car IT)
- ca. 90% generierter Code

OSGi (Open Services Gateway Initiative)

Widget Toolkit

- MMI-Framework Kylie (BMW Car IT)
- E60 "Look and Feel"

Applikationen (CDC hanger, Navigation, Telefon, etc.)

Automotive Systems Lab.

OSGi - Deployment.

OSGi-Framework

- Flexibilität im laufenden System
- smf (OSGi-Implementierung von IBM)

Bundles

- Basisklassen für OSGi (http etc.)
- J2Most und generierte Klassen
- MMI-Framework Kylie (ewt, kylie, appManager)
- Services (CDC hanger, Amplifier etc.)
- iDrive (Control Elements)
- Applikationen (Navigation, Telefon, CDC hanger, etc.)

Deployment-Tool

- "OSGiTool"
- Remote Deployment
- Kommunikation mit OSGi-FW über HTTP-Protokoll
- Nicht notwendig, aber bequem

Automotive Systems Lab.

MMI Framework Kylie - Widget Toolkit.

Bedeutung

- Schnittstelle zum User (Fahrer)
- Qualität, Usability, Distraction etc.
- Andere Modalitäten denkbar (Sprache, Gestik etc.)
- Hoher Implementierungsaufwand

Funktionen

- Registrierung von "MMIs" (Applikationen)
- Festlegung von "Widget-Verhalten" (z.B. Scrolling)
- Darstellung der Widgets (dynamisch)
- Möglichkeiten zur Strukturierung von Widgets (Parent-Child Beziehungen)
- Rendering-Strategie (Rendering-Loop)

Schnittstellen

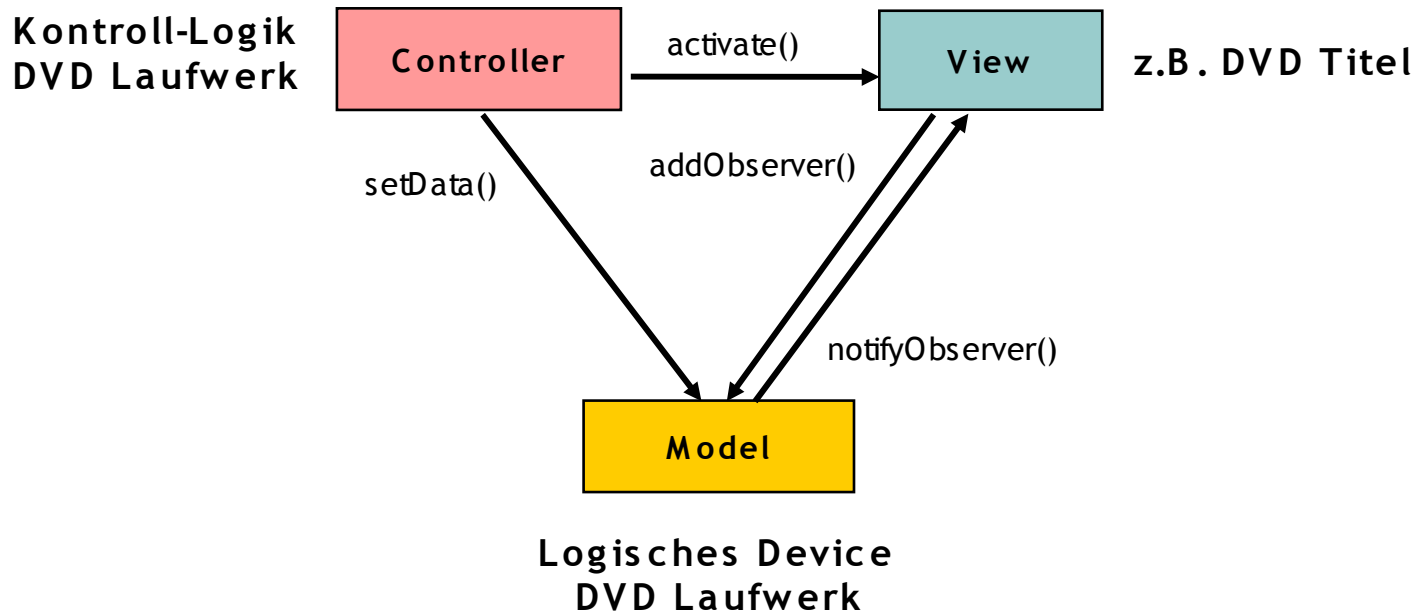
- Graphic Library (Java-Abstraktion zu Photon → ewt)
- Applikationsprogrammierer (Widgets, Focus, etc.)

Automotive Systems Lab.

MMI-Framework Kylie - Model View Controller

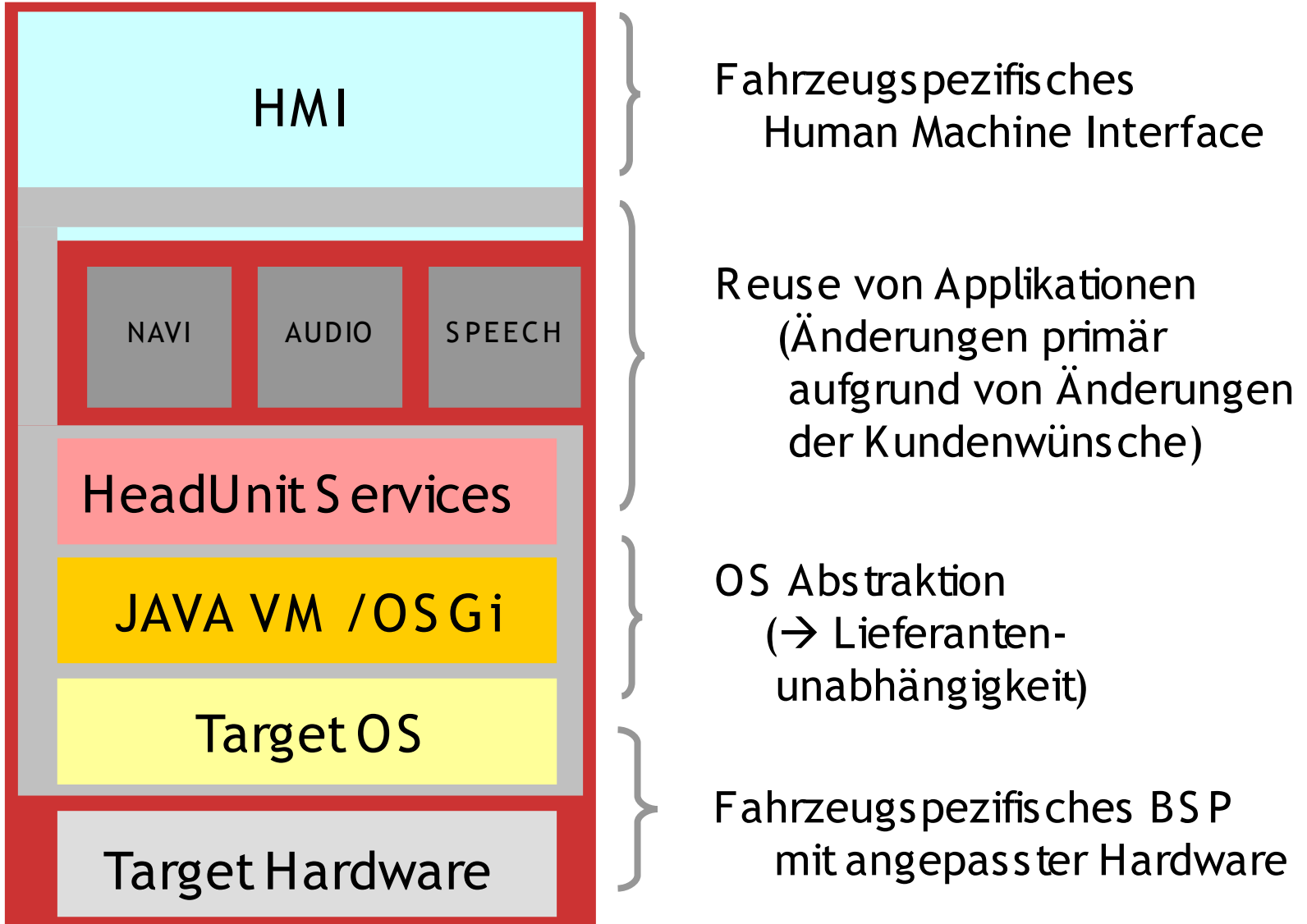
Separierung von Daten, Präsentation und Kontroll-Logik

Prinzip: Model-View-Controller



Automotive Systems Lab.

HeadUnit: Software-Architektur.



Beispiel.

CDChanger Applikation - Deployment.

Deployment der CDChanger Applikation

- Starten des Aufbaus
- Starten von SMF
- Starten des OSGiTools
- Deployment der Basis-Bundles
- Deployment der CDChanger-Applikation (Binary)

Bundle für CDChanger Applikation bauen (Eclipse)

- Export Jar
- Eigenes Manifest verwenden
(/extendedcdchanger/src/etc/meta-inf/Manifest.mf)
- Create Jar
- Deploy ExtendedCDC-Jar

Beispiel.

CDChanger Applikation - Änderungsszenarien.

Einfache Änderungsszenarien

- Darstellung von Widgets (Farbe, Position etc.)
- Position der Applikation
- Track-Daten (Darstellung)

Komplexeres Szenario - Amplifier

- Zugang zu Amplifier-Service
- Widgets für Lautstärke anlegen (Buttons)
- Button-Listener
- Data-Listener