

# Herausforderung pilotiertes Fahren

Der Erfolgsfaktor in der Softwareentwicklung, um diese meistern zu können

Florian Netter, Audi Electronics Venture GmbH

**Das pilotierte Fahren – eine technologische und sogleich funktionale Herausforderung. Hieraus lassen sich aktuelle Aufgabenstellungen für die Softwareentwicklung aus den Bereichen Embedded Entwicklung, Mobile Computing und cloudbasierter Dienste ableiten und damit gezielt bearbeiten. Flankiert durch ein virtuelles Entwicklungsframework und adaptiven Prozessen ergibt sich das Potential, Softwareprojekte optimal unterstützen und so die Herausforderungen im pilotierten Fahren bewältigen zu können.**

## 1. Einleitung

In den letzten Jahren wurden viele Innovationen, unter anderem im Bereich der Fahrerassistenz, konzipiert und zur Serienreife entwickelt. Ein Beispiel dafür ist der *Audi active lane assist*: Beim Verlassen der Fahrspur wird der Fahrer nicht nur gewarnt, sondern auch im Rahmen der Systemgrenzen aktiv bei der Spurhaltung unterstützt. Ermöglicht wird dies durch eine stetig steigende Anzahl elektrischer/elektronischer Systeme und Software im Automobil. Die logische Fortsetzung führt zu dem Thema pilotiertes Fahren (und Parken), welches bereits heute eine der Schlüsseltechnologien der Zukunft ist. Anfang des Jahres hat die AUDI AG mit dem *Audi A7 Sportback piloted driving concept* in den USA einen Langstreckentest von Stanford über 550 Meilen nach Las Vegas erfolgreich beendet. Diese neuen Konzepte sind eine funktionale und sogleich technologische Aufgabenstellung. Daraus ergeben sich neue Herausforderungen für die automotive Softwareentwicklung, zusätzlich muss diese auf aktuelle Trends der Elektrik-/Elektronik-Entwicklung reagieren.

Die Aufgabe der Audi Electronics Venture GmbH als 100-prozentige Audi Tochter ist, bei der Bewältigung der anstehenden Herausforderungen als kompetenter Partner zu unterstützen.

## 2. Einführung pilotiertes Fahren

Die derzeitige Landschaft von Fahrerassistenzfunktionen ist sehr heterogen in den verschiedensten Domänen verteilt, wie z.B. beim Parken der *Parklenkassistent* oder im Bereich Licht der *High-Beam Assist*. Zwischen diesen besteht ein Informationsaustausch, jedoch sind sie untereinander nicht hoch vernetzt und nutzen nur sehr eingeschränkt gemeinsame Basiskomponenten, wie z.B. eine zentrale Sensordatenfusion. Um neuartige Ansätze wie einen Stauassistenten umsetzen zu können, muss eine Konsolidierung auf funktionaler Ebene erfolgen, und die Synergien zwischen allen Applikationen müssen genutzt werden. Dies stellt den ersten Schritt auf einer Entwicklungsroadmap dar, danach können in einer zweiten Stufe assistierende zu pilotierten Applikationen weiterentwickelt werden. Dieser Meilenstein ist wiederum die Basis für alle weiteren Entwicklungen – *Piloted Lane*

*Changing* oder *Piloted City Driving* – welche dann schrittweise den Weg in die Serie finden.

Um die Konzepte umsetzen zu können, benötigt das Fahrzeug eine vollständige 360-Grad-Sicht auf das unmittelbare und mittelbare Umfeld. Hierzu kommen verschiedenste Sensoren mit unterschiedlichen Wahrnehmungsbereichen zum Einsatz, siehe Abbildung 1. Auch muss der Umkreis redundant erfasst werden, um Plausibilisierungen durchführen, Störungen bzw. Fehlfunktionen erkennen und ggf. Sensorersatzwerte bestimmen zu können.

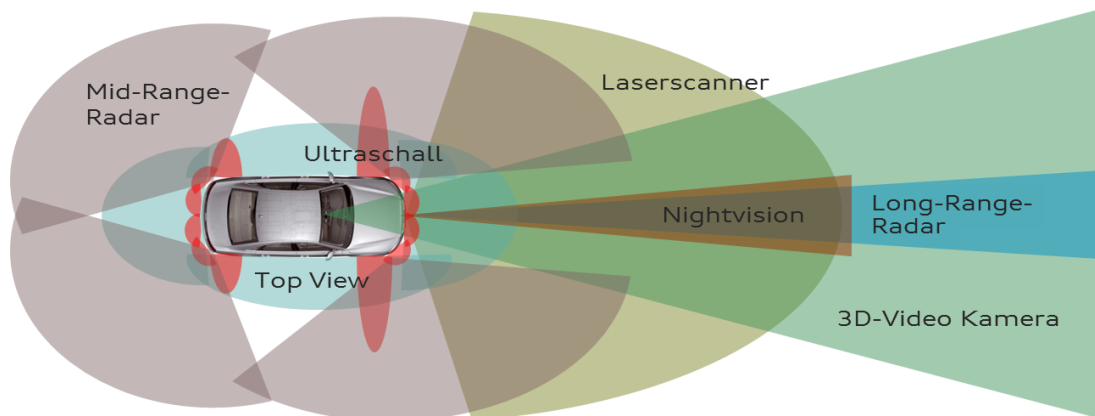


Abbildung 1: Fahrzeugsensorik für Umfelderfassung

Alle Messgrößen werden dazu in der zentralen Sensordatenfusion verarbeitet, welche ein vollständiges Modell der Umgebung berechnet. Zusätzlich können über Car2x weitere Informationen gesammelt und mit verarbeitet werden. Dieses Modell wird nun in der Trajektorienplanung weiter verwendet, um Planungen der Fahrbahn auf Basis des Umfelds durchführen zu können. Eine weitere Softwarekomponente zur *Entscheidungsfindung* berechnet die internen Zustände und aktiviert die entsprechenden Softwaremodule, z.B. die Berechnung einer neuen Trajektorie. Eine *Aktoransteuerung* steuert nun das Getriebe, das elektronische Gaspedal und die Bremsen an, und gibt den berechneten Lenkwinkel vor.

Mittlerweile existiert für die seriennahe Entwicklung das sogenannte zFAS (zentrales Fahrerassistenzsteuergerät) mit einer sehr leistungsfähigen Hardwarearchitektur in der Größe eines Laptops, siehe Abbildung 2, während noch im Jahre 2012 die komplette Technik im ganzen Kofferraum eines Prototyps verbaut wurde.



Abbildung 2: Zentrales Fahrerassistenzsteuergerät (zFAS)

### 3. Herausforderungen für die Softwareentwicklung

Durch die bereits angesprochene funktionale Konsolidierung ergibt sich ein starker Einfluss auf die beiden Themen E/E-Architektur und hochintegrierte Steuergeräte, Ethernet steht als breitbandiger Datenbackbone zur Verfügung. Die Vernetzung ist längst nicht mehr nur innerhalb des Fahrzeugs zu sehen, sondern hat sich in den letzten Jahren auch außerhalb des Automobils stark entwickelt, welches sich mehr und mehr mit seiner Umwelt vernetzt und mit dieser interagiert. Die Car2x-Kommunikation ist ein Beispiel, jedoch auch die Nutzung von aggregierten Flottendaten über cloudbasierte Dienste sei hier erwähnt, Stichwort Big Data. Auch die Vernetzung mit Mobile Devices wird eine sehr große Rolle spielen, um z.B. den Trigger für den pilotierten Ausparkvorgang an das Fahrzeug zu übermitteln. Eingebettet in ein virtuelles Entwicklungsframework können Algorithmen, Architekturen, Steuergeräte, Hardwaremuster, etc. bereits sehr früh geprüft und getestet werden. Zusätzlich müssen all diese Aspekte in Prozessen abgebildet werden. Zusammenfassend lassen sich die Herausforderungen daher wie folgt extrahieren:

- Zukunftsweisende E/E-Architekturen und Ethernet
- Hochintegrierte Steuergeräte
- Car2x/Cloudbasierte Dienste/Big Data
- Mobile Computing
- Virtuelle Entwicklungsmethoden
- Entwicklungsprozesse

In den nächsten Unterkapiteln wird nun auf diese Punkte kurz eingegangen.

#### 3.1 Zukunftsweisende E/E-Architekturen und Ethernet

Die Komplexität der fahrzeuginternen Vernetzung hat in den letzten 15 Jahren stark zugenommen. Dies begründet sich durch eine Vielzahl an Innovationen, die durch elektrische/elektronische Systeme und Software ermöglicht werden, weshalb die Anzahl der Electronic Control Units (ECUs) im Fahrzeug massiv angestiegen ist.



Um all diese Punkte optimal betrachten, entwickeln und nutzen zu können, bedarf es zentraler Systemarchitekturen. Diese müssen neben der Sicht auf das Fahrzeug auch die Innovationen anderer Domänen mit einschließen: Car2x, mobiler Geräte und Cloud-basierter Dienste/Big Data. Zusätzlich müssen die Aspekte hinsichtlich Safety, Security und Privacy während der Entwicklung nach der Methodik „by-Design“ mit konzipiert und implementiert werden. Abbildung 4 stellt diesen Zusammenhang am Beispiel einer Kommunikationswirkkette nochmals grafisch dar.

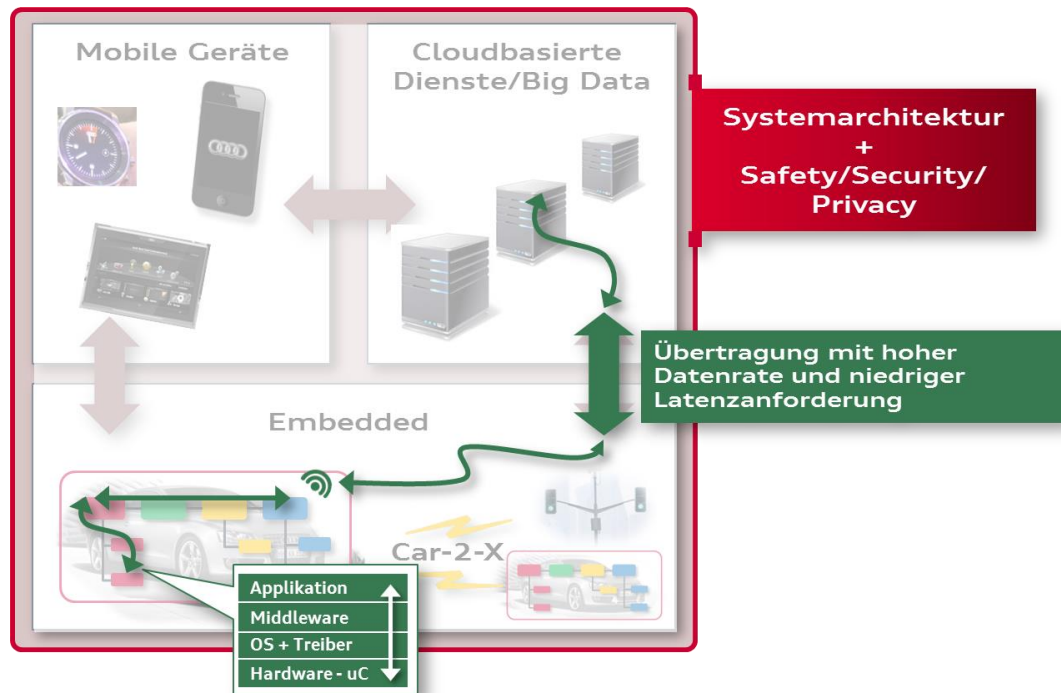


Abbildung 4: Domänen in einer Systemarchitektur mit Beispiel einer Wirkkette

### 3.2 Plattformen

Die Schnittstellen zwischen den vier definierten Domänen in einer Systemarchitektur werden über entsprechende Plattformen und die darin enthaltenen Schnittstellenadapter realisiert. Dadurch ist sichergestellt, dass festgelegte und standardisierte Interfaces verwendet werden, und sich nahtlos weitere Entwicklungsschritte aufsetzen lassen. Außerdem steigert dieser Ansatz die Wiederverwendbarkeit von Softwarekomponenten und senkt so Entwicklungskosten. Als konkrete Beispiele werden drei Plattformen aus den jeweiligen Domänen aufgelistet:

- Embedded: ECU Hochintegrationsplattform
- Cloudbasierte Dienste/Big Data: Framework für Datenanalyse und –verarbeitung
- Mobile Geräte: Mobile Computing Framework

### 3.3 Virtuelle Entwicklungsmethoden

Virtuelle Entwicklungsmethoden sind neben den Plattformen und der Systemarchitektur eine weitere wichtige Komponente in der automobilen Softwareentwicklung. Damit lassen sich Fahrzeugfunktionen frühzeitig in den verschiedensten Situationen und Szenarien verifizieren und validieren.

Bei der Audi Electronics Venture GmbH wird dazu die sogenannte *Functional Engineering Platform* (FEP) entwickelt. Es besteht die Anforderung, nicht nur Embedded Softwaremodule in eine virtuelle Simulationsplattform einbinden zu können, sondern Komponenten aus der gesamten Systemarchitektur, wie z.B. aus den Bereichen mobile Geräte und cloudbasierte Dienste/Car2x. Eine Anbindung an reale Steuergeräte, HIL-Prüfstände, virtuelle ECUs oder Kombinationen aus diesen ergibt vielfältige Setupmöglichkeiten, welche genau an die Testanforderungen angepasst werden können. Die Integration von Sensormodellen, wie z.B. einem Radar, und die Kopplung mit Fahrermodellen und einer Verkehrssimulation ergeben flexible Einsatzmöglichkeiten während der Entwicklungsphasen. Ein mögliches Testsetup stellt Abbildung 5 dar.

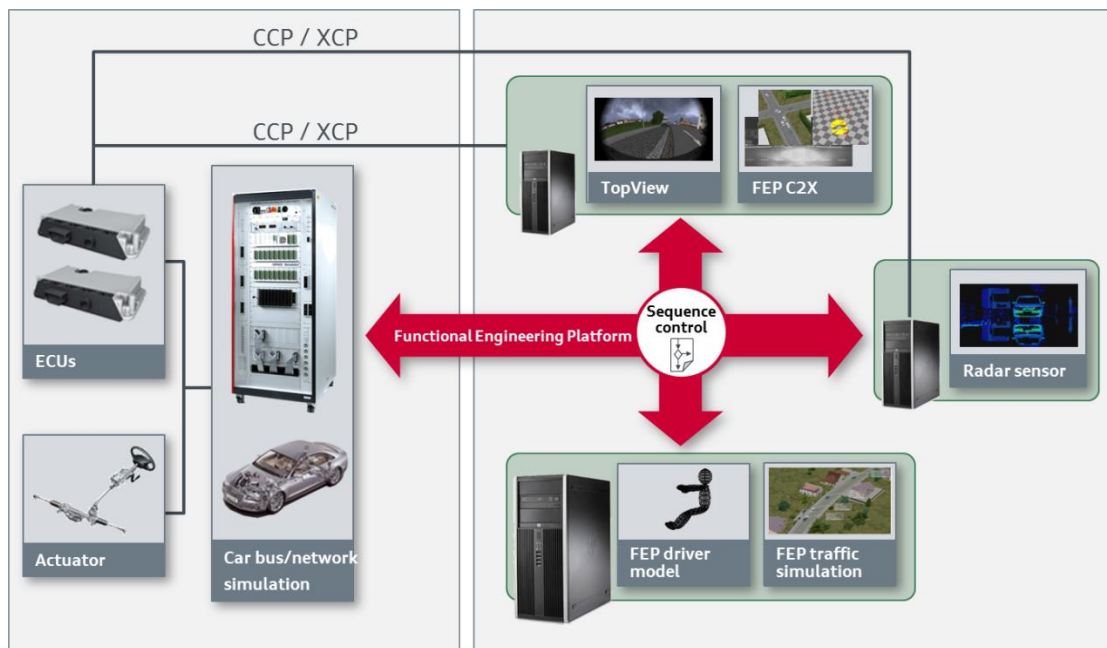


Abbildung 5: Mögliches Testsetup mit der *Functional Engineering Platform*

### 3.4 Entwicklungsprozesse

Mit einer einheitlichen Systemarchitektur, dem Einsatz von Plattformen und der virtuellen Entwicklung sind alle technischen Voraussetzungen erfüllt, um softwareintensive Innovationen hervorbringen zu können. Diese drei Aspekte müssen jedoch sinnvoll eingesetzt werden, um die Innovationen erfolgreich zu entwickeln und nachhaltig im Unternehmen nutzen zu können.

Die Audi Electronics Venture GmbH setzt deshalb auf die Einbettung dieser Aspekte in adaptive Prozesse. So ist z.B. *EnProVe* ein Projekt zur Organisation des kontinuierlichen Verbesserungsprozesses mit dem Ziel, ein einheitliches, modulares Entwicklungsframework für die Embedded Softwareentwicklung bereitzustellen. Es umfasst ebenfalls gemeinsame Tools, geeignete Entwicklungsmethoden, performante Infrastruktur und stellt darüber hinaus Support-Services für die Entwickler zur Verfügung. Dieser EnProVe-Prozess ist nach Automotive SPICE® Level 3 zertifiziert und kann an die spezifischen Rahmenbedingungen eines gewünschten Projektes angepasst werden.

#### 4. Zusammenfassung

Um die Herausforderungen in der Softwareentwicklung im Bereich des pilotierten Fahrens bewältigen zu können, ist es notwendig, von Anfang an gemeinsame Systemarchitekturen zu entwickeln und einzusetzen. Durch einen frühen Einsatz von Plattformen ergibt sich die Möglichkeit, auf standardisierte Schnittstellen zu setzen, um so schnell und nahtlos weitere Entwicklungsschritte aufsetzen zu können. Eingebettet in ein virtuelles Entwicklungsframework können Softwareprojekte sehr früh geprüft und getestet werden. Um all dies in Prozessen abbilden zu können, werden adaptive Prozesse benötigt, um sämtlichen Aspekten aus den Bereichen Embedded Entwicklung, Mobile Computing und Big Data gerecht zu werden.

Durch dieses Vorgehen ergibt sich eine enge Verzahnung von Fahrzeugfunktionen mit der entsprechenden Softwaretechnologie. Zusätzlich dazu stellt eine kontinuierliche und durchgängige Anwendung der genannten Aspekte von der Konzeptphase bis hin zur Maintenance den entscheidenden Erfolgsfaktor bei der Audi Electronics Venture GmbH dar, siehe Abbildung 6. Demzufolge ergibt sich die Möglichkeit, Artefakte aus den vorherigen Prozessschritten zu übernehmen, um so die Entwicklungszeit zu verkürzen, Kosten zu senken und Entwicklungsprojekte optimal zu unterstützen.

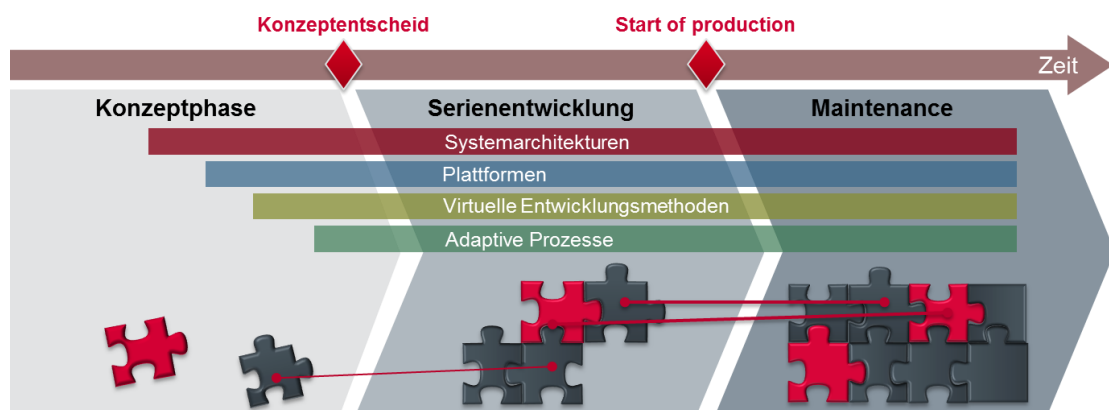


Abbildung 6: Erfolgsfaktor – durchgängige und kontinuierliche Anwendung im Vorgehensmodell

**Autor**

Florian Netter stieg 2003 nach seinem Studium der Elektro- und Informationstechnik an der FH Ingolstadt bei der Audi Electronics Venture GmbH ein. Dort war er zunächst als Entwicklungsingenieur in der Embedded Softwareentwicklung tätig, später auch als Projektleiter und übernahm Tätigkeiten im Bereich „Automotive Ethernet“. Nun ist er zuständig für das Software Competence Center. Die Hauptaufgaben dort sind das Finden von neuen Softwaretechnologien und der Aufbau von entsprechenden Technologieprototypen für Automotive. Seine Interessen reichen dabei von verteilten Embedded-/IT-Systemen und Architekturen über Mobile Devices bis hin zu Themen aus dem Umfeld der Künstlichen Intelligenz.

**Kontakt**

Email: [florian.netter@audi.de](mailto:florian.netter@audi.de)