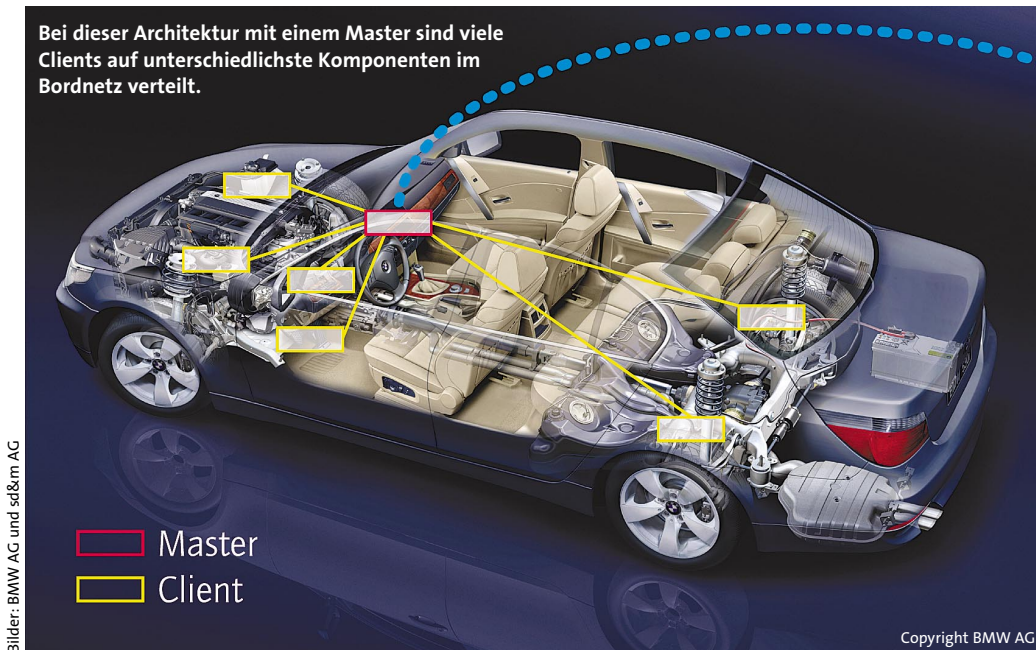


Bei dieser Architektur mit einem Master sind viele Clients auf unterschiedlichste Komponenten im Bordnetz verteilt.



Backend

Master
 Client

Bilder: BMW AG und sd&sm AG

Copyright BMW AG

Verteiltes System

Die Architektur eines verteilten Softwaresystems im Automobil spielt eine entscheidende Rolle – denn nur mit einer klaren Trennung der Zuständigkeiten und sauberen Schnittstellen lassen sich Komplexität und Aufwand beherrschen. **SCHLÜSSELELEMENTE** in einer solchen Architektur sind die Vereinheitlichung und zentrale Implementierung von Client-Anteilen, lose Kopplung zwischen Master und Clients, Trennung von Bordnetzkommunikation und Ablauflogik sowie Kapselung der Backend-Integration.

Am Anfang jeder Entwicklung steht die Produktdefinition. Sie wird u. a. bestimmt von der Unternehmensstrategie, Markenidentität und -weiterentwicklung. Bei Anbindung von Backend-Teilfunktionen sollte mit der Definition auch eine fundierte Prognose des Technologie- und Marktumfeldes relevanter Schnittstellen, Applikationen und Geräte einhergehen. So werden Rahmenbedingungen festgelegt, an denen die Architekturgestaltung auszurichten ist. Die Herausforderung ist u. a., den Zielkonflikt zwischen Innovationsfähigkeit (Produkt) und Standardisierung (Kosten) bestmöglich aufzulösen.

Summary

Software im Automobil wird traditionell von dem Zulieferer entwickelt, der das jeweilige Steuergerät liefert. Die zunehmende Vernetzung der Steuergeräte erhöht allerdings den Bedarf an komplexen Softwarediensten enorm. Dies stellt die Entwickler verteilter Softwaresysteme vor neue Herausforderungen.

Im ersten Schritt der Partitionierung wird entschieden, welche Anteile der

Funktion im Backend, welche im Fahrzeug realisiert werden sollen. Entscheidungskriterien sind hier Kundenwert, Umsetzungs-, Prozess- und Betriebsaufwand, Flexibilität, Standardisierung und Risiken. Wichtig ist es dabei, innerhalb der Gesamtfunktion System- und Kundenfunktionsanteile zu identifizieren und bei der Partitionierung zu berücksichtigen: Systemfunktionsanteile sind infrastruktureller Natur und müssen langfristig mit klar definierten Schnittstellen ausgelegt werden. Kundenfunktionsanteile wiederum können über Fahrzeugprojekte, aber auch innerhalb eines Fahrzeugprojektes auf Grund verschiedener Konfigurationen und Ausstattungsvarianten variieren; sie sind typischerweise in Clients anzusiedeln, die funktional voneinander möglichst unabhängig sein sollten.

Schon bei der Architekturgestaltung muss darüber hinaus ein Projekt aufgesetzt werden, das von Struktur und Durchführung her den Maßgaben konsequenten Systems Engineering ent-

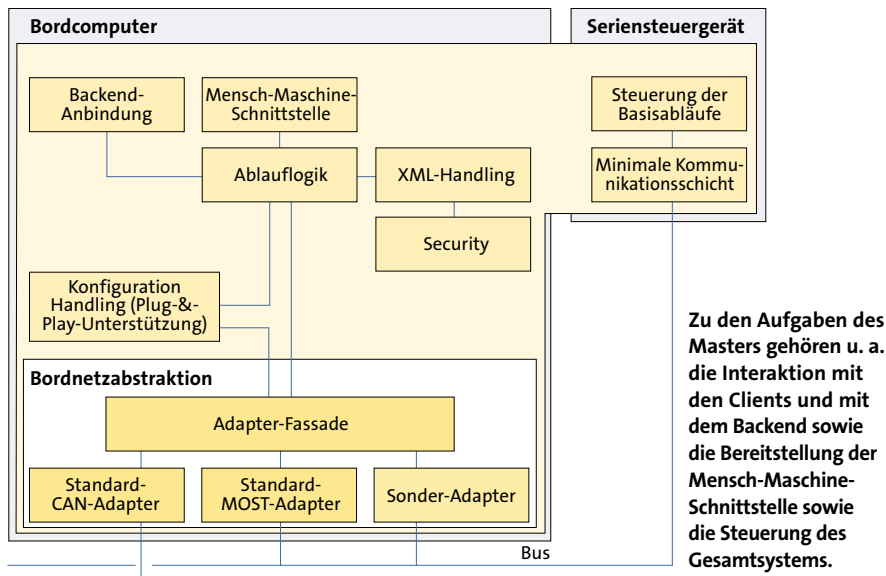
spricht. Dies wurde hier durch die Etablierung eines eigenständigen Projektes in einem übergeordneten Change Programm [1] realisiert. Ein positiver Nebeneffekt ist der Erkenntnistransfer auf andere, prinzipiell gleich geartete Projekte.

Worauf es ankommt

Gerade im Produktentstehungsprozess mit vielen unterschiedlichen Entwicklungspartnern muss das System skalierbar und in seiner Komplexität beherrschbar bleiben. Die Komponenten des Systems dürfen nicht eng aneinander gekoppelt werden, damit vermieden wird, dass Änderungen in einer Komponente Änderungen in anderen Komponenten nach sich ziehen.

Die Entwicklung erfolgt heute nicht mehr dediziert für einzelne Fahrzeugprojekte, sondern im Rahmen eines Baukastens für verschiedenste – auch bereits in Serie befindliche – Fahrzeuge. Daraus resultieren baureihenübergreifende Kompatibilitätsanforderungen, die im Kon-

Architektur des Masters



zept berücksichtigt werden müssen. Fahrzeuge der BMW Group bieten heute immense Kombinationsmöglichkeiten unterschiedlicher Sonderausstattungen [2]. Je nach verbauten Komponenten sind verschiedene Anteile des Systems nicht verfügbar. Das System muss so konzipiert sein, dass in diesen Fällen eine zwar eingeschränkte aber dennoch sinnvolle Kundenfunktion darstellbar ist.

Bei der Architekturgestaltung muss des weiteren die Optimierung des Ressourceneinsatzes für Entwicklung, Betrieb und Service besonders betrachtet werden, insbesondere auch die Minimierung des Konfigurationsaufwands.

Ebenfalls wichtige Ziele sind Rückfallebenen zur Eigensicherheit der Komponenten, Diagnosemöglichkeiten, Minimierung des Einflusses auf andere Fahrzeugfunktionen sowie Restriktionen der Bordnetzkommunikation (z. B. Bandbreite, Topologie).

Flexible Architektur

Diese Anforderungen führen zu einer Architektur mit einem Master und im Prinzip beliebig vielen Clients, die auf unterschiedlichsten Komponenten im Bordnetz verteilt sind.

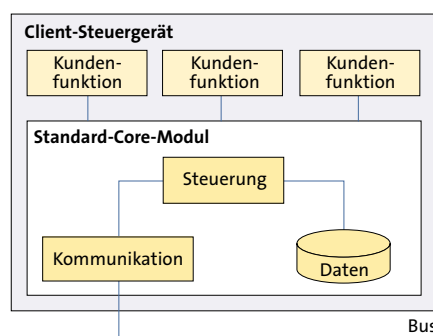
Der Master ist verteilt auf zwei Komponenten. Die Basisfunktionalität, die auch in Fahrzeugen ohne umfassende Sonderausstattungen zur Verfügung steht, wird durch ein Seriensteuergerät gewährleistet. Weiterführende Abläufe, die nur mit Display intuitiv bedienbar sind, werden auf dem Bordcomputer umgesetzt. Mit einem geeigneten Protokoll er-

mittelt der Master dynamisch die im Fahrzeug verbauten Clients und deren Eigenschaften. Neue Clients können im „Plug & Play“-Verfahren in den Systemverbund aufgenommen werden, ohne dass der Master neu konfiguriert (also neu codiert oder programmiert) werden muss.

Vielfältig und doch standardisiert

Die Client-Steuergeräte nehmen separat voneinander Funktionsaufgaben wahr, beispielsweise Displaysteuerung oder Einstellung der Sitze. Darüber hinaus müssen sie aber auch im Gesamtsystem agieren können, u. a.: Interaktion mit dem Master, Empfangen und Senden entsprechender Telegramme, Wiederholung von Telegrammen bei ausbleibender Bestätigung, Auflösung logischer Adressen, Verwaltung der Da-

Architektur eines Clients



Die Client-Steuergeräte nehmen unabhängig voneinander Funktionsaufgaben wahr, z. B. Displaysteuerung oder Einstellung der Sitze.

ten, die mit dem Master ausgetauscht werden, Ablaufsteuerung.

Zu den Aufgaben des Masters gehören u. a. die Interaktion mit den Clients und mit dem Backend sowie die Bereitstellung der Mensch-Maschine-Schnittstelle sowie die Steuerung des Gesamtsystems.

Für eine saubere Architektur ist es wichtig, einzelne Teilfunktionen systematisch zu trennen und in einzelnen Software-Komponenten zu verwirklichen [3], [4]. Beispielsweise bei Änderung der Backend-Schnittstelle muss so nur die betroffene Komponente geändert werden; für den Rest des Systems ist die Änderung transparent.

Integration des Backend

Der Datenaustausch mit dem Backend wird mit Verschlüsselung und beidseitiger Authentisierung geschützt. Der Backend-Anteil des Systems ist im Wesentlichen eine Standard-Web-Applikation. Die Interaktion zwischen Fahrzeug und Backend ist jedoch komplexer als das reine Pull-Prinzip zwischen einem Browser und einem Webserver.

Die Benutzerführung erfolgt teilweise durch die normale Mensch-Maschine-Schnittstelle im Fahrzeug, teilweise durch Seiten, die vom Backend generiert und im Fahrzeug angezeigt werden. Die Konsistenz der Benutzerführung muss dabei sichergestellt werden. ←

Dr. Andreas Heider, Leiter Systementwicklung verteilter kundenorientierter Funktionen, BMW Group.
Bernhard Staudacher, BMW Group;
Dr. Zoltán Mann, Senior-Softwareingenieur, sdem AG

www.all-electronics.de

infoDIRECT Direktlink Code: 209AEL0306

Literaturhinweise:

- [1] J. Hauser et al.: Einführung eines Entwicklungsprozesses in der BMW Elektrik-/Elektronik-Entwicklung, Objektspektrum, 2/2005.
- [2] U. Heiden, A. Wagner, A. Heider, F. Schulte et al.: Funktionalität des neuen BMW 3er – Die perfekte Freude am Fahren, ATZ/MTZ Sonderausgabe – Der neue BMW 3er, 5/2005.
- [3] G. Reichart et al.: Systemarchitektur im Kraftfahrzeug – der Schlüssel für Innovation, Systemintegration und Qualität, 8. Jahrestagung EUROFORUM – Elektronik-Systeme im Automobil, 2/2004.
- [4] J. Siedersleben: Moderne Softwarearchitektur, dpunkt Verlag, 2004.