

TOP-THEMEN: #Multicore #Open Source #Safety & Security #IoT & Smart Edge #C und C++ #KI #Agility

Eine offene Plattform für vernetzte Fahrzeuge – Einführung in APPSTACLE und Eclipse Kuksa

27.02.19 | Autor / Redakteur: Joerg Tessmer und Robert Höttger / [Sebastian Gerstl](#)



Die Open-Source-Plattform Eclipse Kuksa und das europäische Forschungsprojekt APPSTACLE streben an, offene, abgesicherte Standards, Schnittstellen und Plattformen für vernetzte Fahrzeuge zu schaffen.

(Bild: Clipdealer)

Mit Eclipse Kuksa und dem europäischen Forschungsprojekt APPSTACLE entsteht derzeit eine leistungsstarke und zugleich offene Plattform für Connected Cars, welche auch die Eingliederung älterer Fahrzeuge in ein Ökosystem für vernetzte Fahrzeuge einfach gestalten will.

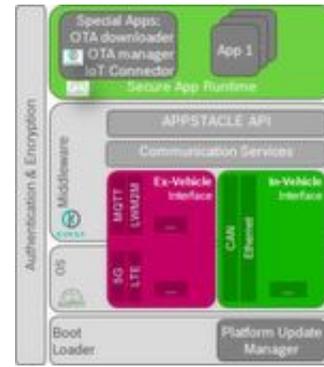


Die Automobildomäne ist von einem intensiven Zusammenspiel zwischen Mechanik, Elektronik und Software geprägt, wobei die Software den wesentlichen Treiber bei der Realisierung von neuen Funktionalitäten darstellt. Während Innovationen innerhalb eines Fahrzeuges weitestgehend durch intransparente Entwicklungsprozesse entstehen, erfordern Entwicklungen im Bereich der vernetzten Fahrzeuge neue und offenere Modelle der Kooperation. Vernetzte

Fahrzeuge setzen die Verbindung unterschiedlicher Fahrzeuge untereinander oder mit weiteren Akteuren wie Herstellern oder Drittanbietern mittels des Internet-of-Things (IoT) zum Zwecke des Datenaustausches voraus. Sie haben dabei das Potential, den Automobilbereich sowohl aus technischer als auch wirtschaftlicher Sicht nachhaltig zu verändern.

Aktuelle Marktstudien prognostizieren dem Bereich des vernetzten Fahrens und den damit verbundenen Daten-basierten Dienstleistungen ein starkes Wachstum und enormes Marktpotential. So prognostiziert beispielsweise der Connected Car Report 2018 von Statista ein Wachstum von der derzeit 107 Millionen vernetzten Fahrzeugen auf bis zu 358 Millionen vernetzten Fahrzeugen im Jahr 2022 [1]. Andere Studien beziffern den potentiellen Markt rund um das vernetzte Fahrzeug und die entstehenden Daten auf 219 Milliarden US Dollar in 2025 (Research and Markets [2]) bzw. auf 450-750 Milliarden US Dollar in 2030 (McKinsey [3]).

Bildergalerie



Fotostrecke starten: Klicken Sie auf ein Bild (7 Bilder)

Bisweilen fehlt es jedoch an offenen, abgesicherten Standards, Schnittstellen und Plattformen für Szenarien vernetzter Fahrzeuge. Diese sind jedoch eine Voraussetzung für die Integration von branchenfremden Akteuren und deren Geschäftsmodellen in Form von innovativen, digitalen Services, wie etwa eine intelligente Verkehrsführung auf Basis großer Datenmengen.

Anzeige

Fahrt aufnimmt

[mehr...](#)

An dieser Stelle positioniert sich das Open-Source-Projekt **Eclipse Kuksa** sowie das damit verbundene **europäische Forschungsprojekt APPSTACLE**. Im Rahmen von Eclipse Kuksa entsteht ein

„Open Source Connected Vehicle Ecosystem“, welches sich zum Ziel gesetzt hat eine technologische Basis anzubieten.

Diese beinhaltet im Wesentlichen drei Elemente: Zum einen entsteht eine **In-Vehicle Plattform**, welche es ermöglichen soll Apps (Anwendungen) nachzuladen die sowohl auf Fahrzeugressourcen zugreifen können als auch eine Kommunikationsverbindung mit beliebigen Cloud Plattformen aufbauen.

Zum anderen wird in Eclipse Kuksa eine Connected Vehicle **Cloud Plattform** entwickelt, welche Dienste im Bereich des Datenmanagements und der Datenanalyse sowie des Gerätemanagements anbietet. Darüber hinaus ermöglicht eine **integrierte Entwicklungsumgebung (IDE)** die einfache Entwicklung von Anwendungen für das Kuksa In-Vehicle- und Cloud Ökosystem.

In den folgenden Abschnitten werden zunächst das Forschungsprojekt APPSTACLE sowie das Open Source Projekt Eclipse Kuksa vorgestellt. Außerdem werden die einzelnen Elemente des Open Source Ökosystems genauer beleuchtet und erklärt. Ein weiteres Kapitel beschäftigt sich mit den möglichen Einsatzszenarien und Systemarchitekturen bevor abschließend ein Ausblick auf die weitere Entwicklungs-Roadmap gegeben wird.

Ein Überblick über die Projekte APPSTACLE und Eclipse Kuksa

Das unter dem ITEA Call 3 geförderte, europäische Forschungsprojekt APPSTACLE befasst sich mit der Etablierung einer offenen und gleichzeitig sicheren Plattform für die Verbindung zwischen Fahrzeugen und der Cloud [4]. Es setzt sich aus einem Konsortium zusammen, dem insgesamt 21 Mitglieder aus vier Nationen angehören (siehe Bild 1). Dazu gehören starke Partner, die Expertise bezüglich Fahrzeugtechnik und -vernetzung beisteuern wie beispielsweise die Robert Bosch GmbH, BHTC, Kocsistem oder auch Otokar. Dazu kommen Wissensträger aus dem Bereich der Software- und Hardwareplattformen wie zum Beispiel Link Motion, Haltian, Technolution, taskit und das Fraunhofer IEM.

Außerdem tragen mehrere Beteiligte ihr Knowhow im Bereich des Cloud-Computings bei (unter anderem die Universität Oulo, die Fachhochschule Dortmund sowie die Bosch Software Innovations GmbH). Ergänzt wird der Zusammenschluss durch Expertise im Bereich der Netzwerktechnik (NXP, Ericsson und Universität Oulo), Security (TU Eindhoven und SecurityMatters) sowie Open Source Organisation durch die Mitgliedschaft bei der Eclipse Foundation. Neben Themenstellungen wie beispielsweise Kommunikationstechnologien für die nächste Generation des Mobilfunknetzes fokussiert sich APPSTACLE sehr stark auf die Entwicklung und Verbreitung des Open Source Connected Vehicle Software Ökosystems.

Um die Entwicklung dieses Ökosystems zu institutionalisieren und zu fördern wurde aus APPSTACLE heraus das Open Source Projekt Kuksa unter dem Dach der Eclipse Foundation gegründet [5]. Im Projekt Eclipse Kuksa sollen nicht nur die implementierten Ergebnisse von APPSTACLE zusammenfinden, sondern es ist auch geplant hieraus ein nachhaltiges offenes Ökosystem entstehen zu lassen.

Die technischen Elemente von Eclipse Kuksa

Das Open Source Ökosystem Eclipse Kuksa setzt sich drei einzelnen Plattformen zusammen:

- Einer In-Vehicle Plattform, welche eine Softwareumgebung bereitstellt, die in ein Fahrzeug integriert werden kann
- Einer Cloud-Plattform, welche Dienste im Rahmen der Datenverarbeitung sowie des Device Managements zur Verfügung stellt

Anzeige



Data
M
ADRESS
DIREKT

**WIR FINDEN
IHRE ADRESSEN**

KLASSISCH - DIGITAL - QUALITATIV

www.datam-adressdirekt.de

Ein Unternehmen der **VOGEL** GROUP

- Einer Integrierten Entwicklungsumgebung (IDE), welche das Implementieren von Apps erleichtert.

Bild 2 gibt einen Überblick über die drei Plattformen von Eclipse Kuksa. Im Folgenden werden die einzelnen Plattformen im Detail vorgestellt und technische Einblicke in deren Struktur gegeben.

Die Eclipse Kuksa In-Vehicle Plattform

Die Eclipse Kuksa In-Vehicle Plattform (siehe Bild 3) baut auf Automotive Grade Linux (AGL) auf, einem offenen Linux-Derivat für die Automobilwelt. Auf Basis dieses Betriebssystems entsteht eine Laufzeitumgebung in der Kommunikationsdienste die Verbindung zur Cloud bereitstellen aber auch Zugriff zu den Fahrzeug-internen Systemen und Daten auf einem hohen Abstraktionsniveau ermöglicht wird. So dient beispielsweise eine Implementierung der Vehicle Information Service Specification [6] – einer Datenschnittstelle die in Kooperation des W3C sowie des GENIVI Konsortiums entwickelt wird – zur fahrzeugunabhängigen Abfrage von Werten und Zuständen aus dem Fahrzeug heraus.

Darüber hinaus erweitert Eclipse Kuksa die von AGL bereitgestellte Applikations-Laufzeitumgebung. Zu diesen Erweiterungen gehören beispielsweise maßgeschneiderte Apps für die Kuksa-Umgebung, aber auch Elemente wie Intrusion-Detection-, oder Device-Management-Komponenten zum Herunterladen, Installieren und Verwalten von Anwendungen (Apps).

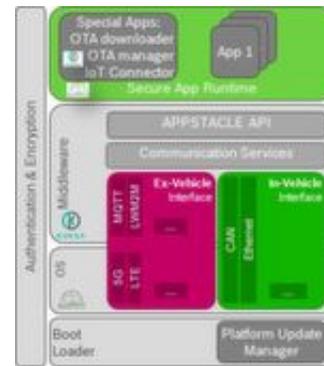
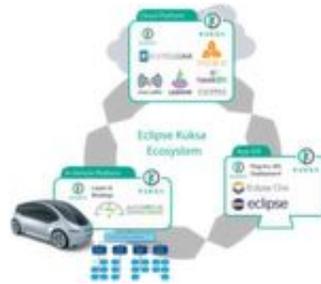
Die Eclipse Kuksa Cloud-Plattform

Die Eclipse Kuksa Cloud-Plattform hat sich zum Ziel gesetzt eine möglichst flexible und leistungsfähige Laufzeitumgebung für die Cloud zu schaffen. Hierzu nutzt und integriert Eclipse Kuksa zahlreiche existierende Open Source Komponenten, vorzugsweise aus der Eclipse IoT Welt. Hierzu gehört beispielsweise Eclipse HONO, welches den einheitlichen Zugang zu allen angebotenen Diensten und Ressourcen ermöglicht. Dabei können sowohl große Datenmengen effizient verwaltet und verarbeitet sowie angekoppelte In-Vehicle Plattformen über eine Device Management Komponente produktiv administriert werden. Bild 4 zeigt eine Übersicht über die Komponenten der Eclipse Cloud-Plattform.

Im Bereich der Datenverarbeitung sorgt die Integration von Eclipse Ditto dafür, dass Daten und Ressourcen möglichst abstrakt in Form eines digitalen Zwillings bereitgestellt werden. Außerdem werden Datenbank-Komponenten (z.B. influxdb oder mongoDB) sowie Datenvisualisierungs-Komponenten (Grafana) zur Verfügung gestellt. Diese Komponenten werden durch die in Eclipse Kuksa entwickelten Services und Schnittstellen zu externen Komponenten ergänzt.

Im Bereich des Device Managements setzt Eclipse Kuksa auf das Open Source Software Roll-Out System Eclipse hawkBit. Dieses ermöglicht die Verwaltung von Softwareständen der In-Vehicle-Plattform selbst sowie das Installations- und Updatemanagement von Apps. Daran angeschlossen ist ein in Eclipse Kuksa entwickelter App-Store (siehe Bild 5), welcher dem Besitzer eines Fahrzeugs ermöglicht Apps auszuwählen und seinem Fahrzeug zuzuweisen.

Bildergalerie



Fotostrecke starten: Klicken Sie auf ein Bild (7 Bilder)

Die Eclipse Kuksa IDE

Die Eclipse Kuksa IDE dient der einfacheren Entwicklung von Apps für die In-Vehicle sowie Cloud-Plattform. Hierzu beschreibt die Eclipse Che Cloud IDE die Basis und ermöglicht die Bereitstellung der IDE ohne dass der Nutzer diese auf einem lokalen Rechner installieren muss.

Die Kuksa IDE ist Docker-basiert und erlaubt sowohl private als auch geteilte Workspaces. Verschiedene neue Implementierungen erleichtern die Anbindung zu zusätzlichen Diensten sowie automatisierte Build Services oder auch die Verbindung zum Kuksa App Store.

Deployment-Varianten von Eclipse Kuksa

Während einzelne Elemente aus dem Eclipse Kuksa Ökosystems sehr flexibel und aufgrund der verwendeten Eclipse Public License (EPL) auch sehr einfach in Produktentwicklungen im Bereich der vernetzten Fahrzeuge übernommen werden können, gibt es auch für das Ökosystem als Ganzes zahlreiche Anwendungsfelder.

Bild 7 zeigt exemplarisch wie Eclipse Kuksa in die Erstausrüstung eines Fahrzeugs integriert werden könnte. Hierbei wird die In-Vehicle Plattform fest im Fahrzeug verbaut und in die E/E-Architektur integriert. Mit Zugriff auf das Fahrzeug-integrierte HMI kann Kontakt mit dem Fahrer hergestellt und über die In-Vehicle Plattform können Daten mit einer oder mehreren Kuksa Cloud-Plattform(en) ausgetauscht werden. Ein Fahrzeugbesitzer könnte seinem Fahrzeug über die Weboberfläche des Eclipse Kuksa App Stores Apps zuweisen bzw. diese auch wieder deinstallieren. App Entwickler könnten die Eclipse Kuksa IDE nutzen um neue Anwendungen zu entwickeln und in den Store einzustellen. Außerdem könnten die entstehenden Daten je nach Nutzungsszenario mit dritten geteilt werden um weitere Services zu ermöglichen.

Eclipse Kuksa ist insbesondere auch für Aftermarket- bzw. Retrofit-Lösungen geeignet. Durch ein Zusatzgerät könnten mit Hilfe des OBD-II Konnektors Fahrzeugdaten gesammelt und in das Eclipse Kuksa Ökosystem übertragen werden. In Verbindung mit beispielsweise

einem Mirroring-System (z.B. Android Auto, Apple CarPlay, MirrorLink) oder anderen HMIs könnten dann ebenfalls neue Eclipse Kuksa Applikationen genutzt werden.

Zusammenfassung und Ausblick

Mit Eclipse Kuksa entsteht derzeit ein leistungsstarkes und zugleich offenes Ökosystem für vernetzte Fahrzeuge. Durch die Bereitstellung von drei aufeinander abgestimmte Plattformen - der In-Vehicle Plattform, der Cloud-Plattform sowie der IDE – wird der Einstieg in den Markt rund um die Daten von vernetzten Fahrzeugen erleichtert. Durch den Einsatz von offenen Standards wird zudem die Interoperabilität des Frameworks mit anderen Systemen sicherstellt.

Die zukünftige Entwicklung wird sich auf den die Anreicherung des Frameworks mit Features sowie die Realisierung von Security-Eigenschaften konzentrieren. Auf Seiten der In-Vehicle Plattform wird der Fokus auf neuen Bestandteile im Bereich des Device und App Managements sowie der Schnittstellen zu Fahrzeug- und Cloud-Ressourcen liegen. Außerdem wird die Eclipse Kuksa Cloud-Plattform mit erweiterten Fähigkeiten bezüglich der Datenverarbeitung und des Datenmanagements bereichert.

(Dieser Beitrag wurde mit freundlicher Genehmigung der Autoren dem Tagungsband des Embedded Software Engineering Kongress 2018 entnommen.)



API-Management für mehr Fahrzeugsicherheit in Connected Cars

15.03.18 - Geht es um das Thema autonomes Fahren, stehen meist Aspekte wie hochpräzise Sensoren und KI im Vordergrund. Doch die zahlreichen Services, die autonomes Fahren ermöglichen, bergen auch Gefahren, denn mit jeder neuen Funktion kommen auch neue Sicherheitslücken und Schwachstellen.

[lesen](#)



(R)Evolution der Automotive-Software-Architekturen

17.10.18 - Mit dem Einzug von prozessorbasierten Plattformen im Fahrzeug findet eine Leistungsexplosion hinsichtlich Speicher, Rechenleistung und Konnektivität statt. Dies lässt aber auch die Software-Komplexität gewaltig ansteigen. Um dem Herr zu werden, müssen sich Entwickler stark auf spezielle Automotive-Software-Architekturen fokussieren.

[lesen](#)

Literaturnachweis

- [1] Statista, „Connected Car Report 2018“ [Online], aufgerufen 10.2018.
- [2] Research and Markets, „Connected Car Market - Global Forecast to 2025“ Research and Markets, Dublin, 2018.
- [3] McKinsey&Company, „Monetizing car data,“ McKinsey&Company, 2016.
- [4] ITEA, „APPSTACLE,“ [Online]. [Zugriff am 20 08 2018].
- [5] Eclipse Foundation, „Eclipse Kuksa,“ [Online]. [Zugriff am 16 08 2018].

[6] [W3C, „Vehicle Information Service Specification,“](#) [Online]. [Zugriff am 16 08 2018].

[7] [The Linux Foundation, „Automotive Grade Linux,“](#) [Online]. [Zugriff am 20 08 2018].

* Joerg Tessmer arbeitet als Konsortiallieter der Robert Bosch GmbH für das ITEA3 Projekt APPASTACLE. Er hat langjährige Erfahrung in den Bereichen der automotiv Toolentwicklung, Entwicklungsumgebungen und der umgebenden Werkzeuge. Neben der externen Koordination bereitet er den Bereich übergreifenden Rollout, die Verwendung der Ergebnisse und die Anpassungen der Ergebnisse und Tools innerhalb der Robert Bosch GmbH vor.

* Robert Höttger forscht im Rahmen unterschiedlicher Industrieprojekte im Bereich der Automobilen Softwareentwicklung an der Fachhochschule Dortmund. Derzeit lehrt er das Modul `Verteilte und parallele Systeme`, arbeitet mit Studierenden an verschiedenen Aktivitäten und ist seit 2011 an mehreren internationalen ITEA-Projekten beteiligt. Seit 2015 ist er Committer für Eclipse APP4MC und seit 2016 Software-Co-Projektleiter für Eclipse Kuxsa.

Kommentar zu diesem Artikel abgeben

Schreiben Sie uns hier Ihre Meinung ...

(nicht registrierter User)

Zur Wahrung unserer Interessen speichern wir zusätzlich zu den o.g. Informationen die IP-Adresse. Dies dient ausschließlich dem Zweck, dass Sie als Urheber des Kommentars identifiziert werden können. Rechtliche Grundlage ist die Wahrung berechtigter Interessen gem. Art 6 Abs 1 lit. f) DSGVO.



I'm not a robot



reCAPTCHA
Privacy - Terms

[Kommentar abschicken](#)

Download-Angebote:



QA Systems GmbH



MicroConsult Microelectronics
Consulting & Training GmbH

Arduino Look-and-
Feel für Multicore -



Parasoft® Deutschland GmbH

Parasoft –
Quality@Speed with
DevOps

mehr...

Seminarbeschreibung "Testen von Software Eingebetteter Systeme" Durchbruch in der Multicore-Anwendung?

mehr...

mehr...

Aktuelle Jobs



Lauterbach GmbH

Ingenieur für das Technische Marketing

mehr...



MicroConsult Microelectronics Consulting & Training GmbH

Trainer & Berater (w/m) für Mikrocontroller und Embedded-Software

mehr...

Embedded Software Engineering ist eine Marke der Vogel Communications Group. Unser gesamtes Angebot finden Sie [hier](#)
AGB | EWG | Hilfe | Kundencenter | Media | Datenschutz | Impressum

Copyright © 2020 Vogel Communications Group

Elektronikpraxis; Clipdealer; Eclipse.org; Robert Bosch GmbH;