OPEN PLATFORMS ON THE WAY TO AUTOMOTIVE PRACTICE

Journal:	*8th ITS European Congress*
Paper ID:	Draft
Paper Type:	Scientific
Date Submitted by the Author:	n/a
Complete List of Authors:	Holle, Jan; University of Siegen, Institute for Data Communications Systems Groll, André; University of Siegen, Institute for Data Communications Systems Ruland, Christoph; University of Siegen, Institute for Data Communications Systems Cankaya, Hakan; escrypt GmbH Wolf, Marko; escrypt GmbH
Topics:	4. From smart concepts to successful implementation.....A. Research agenda (EU and national level)

OPEN PLATFORMS ON THE WAY TO AUTOMOTIVE PRACTICE

Jan Holle, André Groll, Christoph Ruland
University of Siegen, Institute for Data Communications Systems
Hölderlinstraße 3, 57076 Siegen, Germany
Phone: +49 271 740-3325, Fax: +49 271 740-2536
{jan.holle, andre.groll, christoph.ruland}@uni-siegen.de

Hakan Cankaya, Marko Wolf
escrypt GmbH – Embedded Security
Leopoldstraße 244, 80807 München, Germany
Phone: +49 89 208039-117, Fax: +49 89 208039-127
{hakan.cankaya, marko.wolf}@escrypt.com

ABSTRACT

Two clicks on the vehicle console touchscreen and the new parking lot finder application is downloaded online from the OEM application market directly into the car. The introduction of standardized open in-vehicle IT platforms hence not only provides more functionality, more flexibility and a more efficient sharing of vehicle resources, it also creates new perspectives onto the interaction between people and their cars. Thus, open platforms will provide various new possibilities for OEMs, third parties and users by enabling individual application developments together with an easy application deployment process that will yield to a new vivid automotive application development community.

In this work, we will try to establish a common understanding of open automotive platforms. We will discuss the different challenges on their way to automotive practice and will give a brief survey together with first evaluations about existing or ongoing open automotive platform projects.

INTRODUCTION AND MOTIVATION

Until a few decades ago (virtually until the 1990s) cars were closed, electro-mechanical systems with only a few, isolated, and mainly uncritical IT systems. However, this setting has changed dramatically. Today even compact vehicles have already multiples of the computing power of an Apollo spacecraft from the 1970s while having a few tens of interconnected microprocessors with up to several hundred megabyte of software installed together with first integrated communication capabilities to the outside world. Moreover, this very impressive and dynamic development is still at its very beginning. With the constantly increasing number of vehicular applications, functions, and various (new) stakeholders involved, the requirements for in-vehicle IT performance, flexibility, and complexity will further notably increase (2).

On the other hand, the so far usual approach of simply adding a new Electronic Control Unit (ECU) for every new or third party vehicle function has already been maxed out. Today, with almost 100 individual ECUs in a car, the in-vehicle network and maintenance complexity has already reached its technical and economical limits for a safe and efficient handling (1). Consequently, future vehicular IT architectures try to merge several individual control units into a few powerful ones. This enables modern vehicles to adapt their functionality in a dynamic and very flexible manner. Instead of replacing a wired ECU, now adding, updating or removing a vehicular function simply means adding, updating or removing the corresponding application software on the shared platform. Applications further can then access shared resources and services that would be virtually unaffordable applying the hitherto existing dedicated ECU per application approach.

Beyond that the application development process and the IT user profile is changing. User oriented IT services are nowadays an integral part of a vehicle adding value to the product by enhancing functionality and providing a different and new connection between driver and car. Especially the interaction between vehicle and user provides an emotional binding to the product (3) (4). Unfortunately, the static and slow IT development of the automobile branch is unable to catch the dynamic lifecycle and variety of today's IT world whereas an open platform with a strong user community could provide different approaches and perspectives in this area. Having standardized runtime environments and interfaces publicly available would enable users and third parties to develop vehicular functionality at their own enabling a vivid, open market for new vehicular functionality. Furthermore, it could enable manifold vehicle individualization resulting in a strong emotional binding of users to their vehicle and brand. This binding could be similar to what have been observed with Apple's AppStore for iPhones in the smart phone market or personalization of software (e.g., by use of "personas" for the Firefox web browser). Such an approach would need strong runtime isolation and a clever deployment process since insecure or untrusted applications would share resources with safety relevant software. An interesting example in this context would be smart phones creating an extreme enhanced product from a mobile phone by changing the approach to its application development and deployment process, focusing on the interaction design and creating a mutual relation between platform and user community.

Thus, the parallel execution of individual ECU applications on a standardized, open, multi-purpose vehicle IT platform allows for a noticeable more efficient, more flexible, more secure and easier utilization of the always scantily hardware resources. It decreases complexity and costs during production, operation, and maintenance for (redundant) automobile IT hardware and necessary wiring and increases value by providing new interaction possibilities with the user.

OPEN IN-VEHICLE PLATFORMS

An open platform (as considered here) is a specified in-vehicle IT environment to manage and execute several software applications in parallel. Therefore, the platform offers software services (file system, software timer, etc.) and provides access to hardware components like the human machine interface or to the on-board network. One of the main benefits of this approach is that applications developed for this environment are independent from underlying implementation details of the platform. Thus, these platforms act as an abstraction layer. In the automotive domain, platforms are a well-known concept to reduce complexity and allow the development of reusable applications and components. Hence, the main new aspect is the openness of platforms. Concerning this feature, the following questions have to be answered.

Open to whom: By defining and publishing a platform specification, everyone becomes able to develop software applications for the platform. Nevertheless, this does not necessary imply that everyone is allowed to install software on a particular platform, so the platform can be controlled for deployment. Openness in our definition means every person being able to develop and distribute software to potential customers, who are allowed to install software on their platforms. However, this of course does not inherently mean that software is not checked or certified, for instance, by the original platform developer (e.g., AppStore by Apple).

Open for what: Especially in the automotive sector, today available platforms are typically dedicated to one domain, such as control of the comfort functions or infotainment. Due to the increasing demand for interactions between applications, also from different domains, the demands for platforms being able to serve different kinds of applications in parallel is rising. Therefore, open means, being able to operate applications from different domains, for instance infotainment, comfort functions and ITS in parallel.

Technical implementation of openness: There are different possibilities for the technical realization of open platforms, ranging from pure hardware platforms up to rich application runtime environments (e.g., Java Virtual Machine). While high-level platform implementations usually offer easy and fast application development, they are usually inherently dedicated to a specific application domain and therefore provide less flexibility and performance than platform implementations that are close to hardware. Open means therefore that a platform is able to serve different software execution environments for different purposes in parallel.

Shared (hardware) resources: Efficiency is one of the main goals in automotive development and production, especially in terms of money. For automotive electronics, this means that as less and as cheap as possible ECUs should be integrated into the vehicle. This will save both, costs and weight, which improves the efficiency in general, that means, also in terms of fuel consumption or maintenance efforts. Therefore, open in-vehicle platforms should be able to share existing resources as much as possible.

Figure 1 shows a generic view on an open in-vehicle platform approach to highlight the underlying architecture and components that could realize the features of open in-vehicle platforms as mentioned within this section.

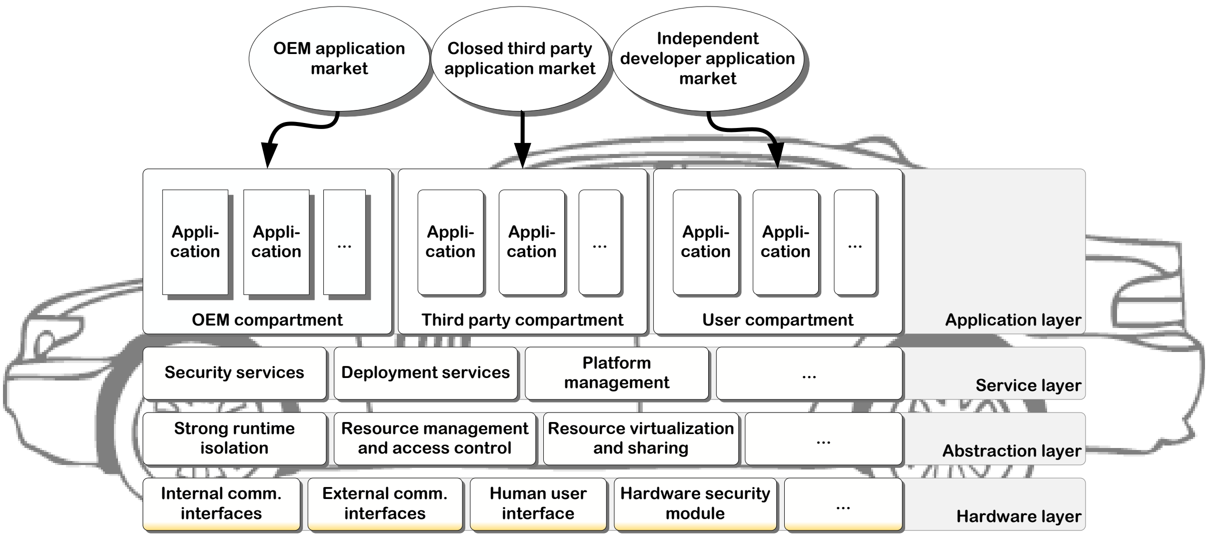

Figure 1: Generic view on an open in-vehicle platform architecture

CHALLENGES OF OPEN IN-VEHICLE PLATFORMS

In order to realize a reliable and trustworthy open automotive software platform, we have to face several challenges. The most important challenges for open in-vehicle platforms are shortly described below.

Strong runtime isolation: Executing several ECU applications in parallel on a shared IT platform requires reliable measures to protect them against each other and especially to protect the underlying software layer that realizes the platform resource sharing and management. That means, neither an accidental malfunction nor a systematic manipulation of one application should affect or compromise the underlying management layer nor any other application executed in parallel.

Dependable resource management and quality of service enforcement: Applying the IT security measures for reliable authorization verification, a shared automotive software execution platform requires a dependable management at least for processing power, memories and communication bandwidth. Further a proper QoS enforcement is required that is able to always assure the minimum resources required for each ECU application executed especially regarding potential real-time demands of safety relevant applications, not only in normal concurrency but also in cases such as manipulative attacks of untrusted Software.

Effective IT security and trustworthiness: Effective security mechanism are required to prevent or at least to reliably detect any kind of malicious manipulations on stored or communicated data, to protect critical secrets (e.g., IP, privacy data) and reliably enforce access control based on proper entity authentication (e.g., humans, components, services) together with an effective verification and enforcement of access authorizations. Being able to prove the assurance of the effectiveness of the implemented IT security measures, OEMs, suppliers, third parties and especially vehicle users are willed to put necessary trust into the protection of their critical data and functionality (7).

Seamless HMI integration: A particular challenge is the seamless fusion of our generic, standardized, and open IT platform with the manifold of the vehicles individual HMI peripheries that are seen as distinctive feature for most OEMs. On the other hand, as learned from the mobile phone market, IT platforms with extensive user interactions stand or fall with its usability capabilities. Thus, the open platform has to feature flexible mechanisms to integrate different types of interaction methods providing the OEM and vehicle user a variety of capabilities to integrate, enhance and customize the interaction architecture.

Efficient and open deployment process: A comprehensive infrastructure is needed dealing with evaluation, certification, distribution and deployment of the vehicle applications. Especially for applications from third-party companies and independent community members, the platform should provide an efficient, flexible and as open as possible deployment process. Hence, effective solutions are necessary for independent/third-party software developers to easily deploy their software and for OEMs as well for being able to guarantee a safe and secure platform operation by evaluating the applications and provide an easy to use and trustworthy end-user channel for downloading applications. Furthermore, the open platform has to provide trustworthy mechanisms for application updates and upgrades and potential billing processes in an automated and user-friendly manner as possible.

Efficient business models and clear liability: Important financial and legal challenges that have to be resolved between all involved parties (e.g., OEM, vehicle user, third party provider) before implementing such a shared open platform: sharing of platform costs (acquisition, operation, maintenance), sharing of liabilities and warranties, or sharing of potential revenues. Especially legal issues regarding third party software are closely coupled with the certification and deployment process.

Open standardization: An open platform has to provide and to enforce obligatory open standards as far as possible since this would be the key factor for successful marketing and successful long-term business models. The simplest part of this standardization probably is the publication of the software developer kit (SDK). Furthermore, the OEM (consortium) should standardize the hardware platform itself to enable long-term development and marketing of modules by many (independent) suppliers. Thus, the standardization has to find a fruitful balance between the requirements from individual developers and suppliers and the overall requirements for maintaining a holistic common platform. On the other hand, the upcoming platform has to consider already existing or already emerging standards as an integral part of its design. An example would be the integration of the emerging ETSI ITS standard (9) that standardizes architectures, protocols and functional blocks for intelligent transportation communications.

Market launch: The first important step for the realization of an open vehicle platform is the decision of the OEMs to allow the integration of such an open platform into their vehicles. Especially since this step not only means many practical changes to their current hardware, software and communication interfaces, but also means a change in philosophy for opening their vehicles to independent developers. The availability of smart phones with their tremendous application pools may create another psychological barrier that an additional dedicated in-vehicle platform would not be accepted. The success of the platform and the attractiveness of the corresponding markets moreover highly depend on the platform acceptance by third party developers and the existence of a strong independent user/developer community. Especially in the first phases of the market launch, this will be a quite difficult challenge as the number of users and developers will be quite small.

STATUS QUO AND FUTURE OF PLATFORM CHALLENGES

In the following chapter, we will give a short state-of-the-art evaluation and a short outlook into the future (if possible) for the challenges identified for the open in-vehicle platforms.

Strong runtime isolation: Possible realizations that allow for efficient and flexible resource sharing while reliably enforcing strong isolation are hardware-based approaches such as ARM’s TrustZone technology or mainly software-driven virtualization approaches (e.g., hypervisors or microkernels) (5) (6).

Dependable resource management and quality of service enforcement: Effective measures for enforcing QoS guarantees and fine-grained access control is usually an integral component of most approaches for resource sharing enabling strong runtime isolation (cf. paragraph above). Thus, the Xen virtualization approach, for instance, already has an efficient and flexible resource management (19), while the INTEGRITY resources sharing with QoS approach is even real-time capable (20).

Effective IT security and trustworthiness: Most existing resource sharing approaches based on virtualization can be easy extended by effective and efficient security mechanisms for protection of software, data, settings, communications, or access control against manipulations or unauthorized access (19). Together with well-established security evaluation methods, for instance, based on the Common Criteria approach (21), corresponding evaluation efforts can comprehensibly assure the effectiveness of the implemented IT security measures, and hence create necessary trust into the protection of their critical data and functionality for all parties involved.

Seamless HMI integration: Assuming a standardized hardware platform as base for our open platform approach, a seamless HMI integration should be possible. Even with several classes for integrated hardware features and hardware performance, standardization reduces complexity considerably and it will enable a tight connection between HMI and platform while allowing a distinctive look for ensuring the individual branding.

Efficient and open deployment process: Mechanisms and infrastructures for efficient and flexible application deployment can be adapted from the already existing smartphone application markets. There, the end-user gets access to the applications via dedicated public interfaces providing easy search, payment and installation facilities. Software will be evaluated and certified either by the entity responsible for the platform (e.g., the OEM) or the entity responsible for a certain platform domain (e.g., an independent third-party developer). The open platform concept would also include an isolated user-controlled domain where the user could install applications independently from most restrictions that includes also uncertified software without being able to endanger any other domains executed in parallel.

Efficient business models and clear liability: The success of most smartphone application markets demonstrates the practical feasibility of a fruitful synergy between platform providers and independent applications developers. Creating similar successful legal agreements and business models hence seems feasible also for open in-vehicle platforms, even though these non-technical issues cannot be tackled in detail by this article.

Open standardization: In order to promote the development of open standards, OEMs should provide detailed guidelines, comprehensive libraries and/or a full SDK for easing the software development for their vehicular platforms as much as possible. The provision of service oriented APIs could further simplify the use of all available platform facilities. However, for long-term success, we would suggest the standardization of the whole platform done within a well-balanced alliance of automotive OEMs, automotive suppliers and third party vendors that would automatically create a strong acceptance in the industry together with a large user basis. Having a widely accepted standardized, easy-to-extend kind of plug-in architecture would enable the manifold development of individual hardware and software components while ensuring the compatibility with virtually all existing platforms. The necessary standardization processes could be integrated within existing standardization bodies, which have the necessary automotive expertise such as ETSI (8) and/or IEEE (10).

Market launch: The open platform philosophy together with a strong user community is the base for the success of today's smartphones. Once OEMs have decided to integrate open vehicle platforms in their cars, again a well-balanced alliance of automotive OEMs, automotive suppliers and third party vendors could create the necessary strong acceptance in industry together with a large user basis. To increase the value of the platform attractive applications have to be realized that cannot realized or used similar with smartphones (22).

This could also demonstrate the special potentials of an open vehicular platform and stimulate the creation of a community. Furthermore, the open platform should be marketed as a whole including the necessary infrastructure.

SURVEY OF CURRENT OPEN IN-VEHICLE PLATFORMS

The current section describes some state-of-the-art projects and products, started to develop or provide an open in-vehicle platform. All platforms are at least suited (except Ford SYNC®) to install applications, which could be chosen by the driver freely. The compared products and projects are: AutoLinQ™ from Continental (13), the GENIVI platform proposed by the GENIVI Alliance (12), an OEM concept exemplarily represented by Ford SYNC® (14) and OVERSEE (1), a European research project on open platforms. Finally, mobile smartphones and AUTOSAR, which are often discussed as a possible open in-vehicle platform, were considered.

AutoLinQ™: AutoLinQ™ is a platform that is currently in development stage. The company Continental and other partners plan to build a platform aiming to enable easy development and integration of infotainment applications. Doing so, the focus of AutoLinQ™ is to offer a holistic user experience integrating home, mobile and in-vehicle view of applications. The platform is based on the open source project Google Android™ (23); a SDK including an emulator are already available. Security and safety issues were already considered in the project and the solutions will be based on an application certification process involving the OEMs. Beyond this, legacy Android™ applications, which could be executed on the platform, will not able to be executed while one is driving. Anyway, the project is still in first stage and a real world implementation in vehicles will not be expected before the next vehicle product generation.

GENIVI: The reference platform will be developed by the GENIVI Alliance, involving a lot of OEMs and automotive suppliers. The GENIVI (Geneva In-Vehicle Infotainment) platform focuses on infotainment applications and it consists of Linux based core services, middleware and an application layer interface. Currently, the GENIVI core code is combined with MeeGo an open source Linux based operating system (16). By now, an integration of GENVI in real world vehicles is not foreseeable.

Ford SYNC®: Ford SYNC is an already available product offered in many brands of the FORD group. Until now, the platform was only used for applications offered by this OEM, so it was a closed platform. Anyway, since the platform is under the control of the OEM, a good connectivity to the vehicle electronic is implemented (e.g., to support vehicle diagnosis services). Currently, Ford aims to offer a programming interface for SYNC that provides applications running on mobile devices an interface to the vehicle electronics. Thus, the application still runs on the mobile device but could be accessed from the vehicle display as well as control units and vice versa. However, Ford SYNC® is therefore a special case, not fulfilling all requirements of an open platform. The platform uses Microsoft Auto (15) as software base and Bluetooth as wireless interface.

OVERSEE: OVERSEE (Open Vehicular Secure Platform) is a European research project aiming at the development of a reference specification and implementation of an open platform and showcasing its feasibility. Since OVERSEE is based on software virtualization technologies, software components for this platform are not limited to a special operating system. Indeed, the software components could consist of an application specific part and an adapted operating system. The project mainly focuses on security issues raised by open platforms and the secure integration of a wide range of communication means (e.g., WiFi, ITS communication, CAN). Results of the project will be available at the end of 2011. By the middle of 2012, a demonstrator will be presented, too.

Table 1 provides an overview on the presented platforms and their assessment concerning some important aspects of open platforms as well as a status of the development process.

	Open for development	**Open for deployment**	**Software environment for applications**	**Development status**
AutoLinQ™	Yes, SDK available	Application certification process	Android plus AutoLinQ™ API	Prototypes shown
GENIVI	Yes	Application certification process will be considered	Linux likewise environment	Reference platform available
Ford SYNC®	Restricted API for applications on connected mobile devices	Application store for mobile device applications with SYNC capabilities, controlled by the OEM.	Depends on the individual device	In the field (except openness for third party applications)
OVERSEE	Yes	Application certification process will be considered	OSEK, Linux, (further possible)	Design specification available, implementation in progress

Table 1: Overview of open in-vehicle platforms

CONCURRENT OPEN VEHICLE PLATFORM APPROACHES

Often smartphones or AUTOSAR (17) are regarded as open in-vehicle platforms as well. The following two sections discuss why these approaches are incomparable to open platforms as they are introduced in this paper.

MOBILE SMARTPHONES AS OPEN IN-VEHICLE PLATFORMS

Within the automotive and ITS community, smartphones are regularly discussed as an implementation of an open in-vehicle platform. Indeed, they are open platforms with imposing flexibility and functionality. Since smartphones are also becoming more and more the key device in people's life, replacing a broad range of classically discrete devices, for instance media players, navigation systems, etc., it seems to be natural to bring this approach to the automotive domain, too. Nevertheless, especially concerning the challenges to open in-vehicle platforms, mentioned in the sections above, there are some strong contradictions.

HMI integration: The main job of the driver of a vehicle is to steer his vehicle in a safe manner. Therefore, one of the main goals of application development in automotive is to avoid any unnecessary distraction of the driver. Integrating mobile phones with their complex handling is problematic and, additionally, touching it while driving is thus prohibited in many countries. Furthermore, as described in the section above, the seamless integration of HMI would be a huge task depending on the amount of implementations on both sides – OEM or brand specific user interfaces versus smartphone specific interfaces.

Onboard integration: Most of the vehicle-based functions are controlled via the vehicle internal networks. Mobile phones are typically not equipped with appropriate interfaces, for instance CAN. And also if it would be feasible to add these interfaces there would be still the question if drivers and/or the OEM would allow to connect a smartphone – with its typically untrusted applications – to their safety relevant vehicle networks.

Runtime environments: Solving the issues of a secure integration of smartphones in the vehicles communication system would be only one part of the integration. A further question is how to cope with the real-time requirements of automotive applications and their inherent needs of dependability. Current smartphones are not able to guarantee spatial and timely separation of applications and also the assurance of dedicated computing power and time slots for execution of applications is not provided. Therefore, it seems to be unreasonable to serve any safety relevant application on top of smartphones.

Physical constraints: Smartphones are designed for execution in well-tempered, dry environments with well-standardized equipment. They are not constructed to withstand the strong physical constraints if left in a car during severe frost or strong heat or to withstand voltage peaks from a typical vehicle power supply for instance.

While this reflection on smartphones as open in-vehicle platforms is still not exhaustive, it seems that smartphones are not really a reasonable alternative to dedicated open in-vehicle platforms. Nevertheless, smartphones would for sure become connected with open platforms to allow seamless integration of user data and additional connectivity.

AUTOSAR AND OPEN IN-VEHICLE PLATFORMS

AUTOSAR (Automotive Open System Architecture) is a standard, which is continuously developed by a consortium of OEMs and ECU suppliers. The current published version is 4.0. The group aims to define a common software architecture for embedded software in vehicles. This standard would allow defining software components independently from a particular ECU in order to improve, among others, the reusability of software components (17).

While the AUTOSAR standard is a real valuable tool to design embedded systems in automotive, the standard is currently not aiming at infotainment systems (18). Since an open in-vehicle platform should at least be able to execute applications from the infotainment domain, AUTOSAR is not a holistic and suitable solution. However, developers of open in-vehicle platforms should consider parts of the AUTOSAR standard as an option for the definition of the communication to the vehicle's electronics to achieve easy integration with upcoming ECUs using this standard.

CONCLUSION AND OUTLOOK

The current work shows that open platforms are an upcoming and promising trend in the automotive domain as well, which seems especially meaningful for the deployment of upcoming ITS applications, where adding further extra devices solely for ITS purposes seems not reasonable for us. By providing our definition of openness for an in-vehicle software application platform and by summarizing the challenges and state-of-the-art approaches for realizing such a platform, we gather a holistic view on the topic. The survey of a selection of current products, projects and initiatives in this area shows the diverse approaches and understandings around this issue.

The ongoing project OVERSEE, where the authors are involved in, will contribute to the development of such a open in-vehicle platform especially by providing solutions for the often disregarded questions regarding security, privacy and resources sharing, all with respect to vehicular safety and flexibility. We believe that a common and therefore considerably larger market for automotive applications could have the same success as we have observed in the mobile device market and will change the interaction between vehicles, users, and environment in a radical way.

ACKNOWLEDGEMENT

The presented paper is part of the ongoing work in the OVERSEE project, a European research project funded within the 7th Framework Programme of the European Union (Project ID FP7-ICT-248333). Therefore, we would like to thank the European Union for funding and all other partners of the project for their jointly contribution.

References

(1) E. Coelingh, P. Chaumette, M. Andersson, Open-Interface Definitions for Automotive Systems – Application to a Brake-By-Wire System, Technical Paper 2002-01-0267, SAE International, March 2002

(2) H.-G. Frischkorn, Automotive Software – The Silent Revolution, In Workshop on Future Generation Software Architectures in the Automotive Domain, San Diego, USA, 2004

(3) Dan Saffer, Designing for Interaction: Creating Innovative Applications and Devices, New Riders, Berkeley, Calif, 2009

(4) D. A. Norman, Emotional Design: Why We Love (or Hate) Everyday Things, Basic Books, New York, 2005

(5) P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the Art of Virtualization, In Proceedings of the 19th ACM Symposium on Op-erating Systems Principles, New York, NY, USA, 2003

(6) K. Elphinstone, G. Heiser, R. Huuck, S. M. Petters, S. Ruocco, L4Cars, In 3rd Work-shop on Embedded Security in Cars (escar), November 2005

(7) M. Wolf, Security Engineering for Vehicular IT Systems — Improving Trustworthiness and Dependability of Automotive IT Applications, Dissertation, Vieweg+Teubner-Verlag, 2009

(8) European Telecommunications Standards Institute (ETSI), www.etsi.org

(9) ETSI, EN 302 665 V1.1.1. Intelligent Transport Systems (ITS); Communications Architecture

(10) Institute of Electrical and Electronics Engineers (IEEE), www.ieee.org

(11) OVERSEE project, www.oversee-project.com

(12) GENIVI Alliance, www.genivi.org

(13) AutoLinQ™, www.autolinq.de

(14) Ford SYNC®, www.ford.com/technology/sync/

(15) Microsoft Auto, www.microsoft.com/auto

(16) MeeGo, www.meego.com

(17) The Automotive Open System Architecture (AUTOSAR)., www.autosar.org

(18) O. Kindel, M. Friedrich, Softwareentwicklung mit AUTOSAR, dpunkt.Verlag, Heidelberg, 2009

(19) Sailer, R. and Jaeger, T. and Valdez, E. and Caceres, R. and Perez, R. and Berger, S. and Griffin, JL and var Doorn, L., sHype: Mandatory Access Control For XEN, 2006.

(20) Green Hills Software, The INTEGRITY Real-Time Operating System, http://www.ghs.com/products/rtos/integrity.html, 2010.

(21) Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007, Continually maintained as ISO 18045.

(22) T-Systems International GmbH. Telekom App-Award, www.appfahren.de

(23) Android™ , www.android.com