Designing Open Computing Platforms with Security in Mind

Hakan Cankaya(2) Jan Holle(1)

(1) University of Siegen, Germany
jan.holle@uni-siegen.de

(2) escrypt GmbH - Embedded Security, Germany
hakan.cankaya@escrypt.com

oversee
Open Vehicular Secure Platform

4th ETSI TC ITS Workshop
07 - 09 February 2012 - Doha, Qatar
Motivation – Why Open and Secure In-Vehicle Platforms for ITS

Challenges and Requirements

OVERSEE as an Open and Secure In-Vehicle Platform for ITS

Standardization and Consensus building

Conclusion & Outlook
Motivation – Why Open and Secure In-Vehicle Platforms for ITS

- Efficient sharing of (HW) resources
- New business models (e.g., application store)
- Development of vehicle independent applications
- Easy deployment of ITS Services (or any other application)
- Overcoming the chicken-and-egg problem of ITS deployment
Motivation – Why Open and Secure In-Vehicle Platforms for ITS

- No safety without security
  - No ITS deployment without a secure foundation!
- Dependability and Security Assurances provided by platform
  - Reduction of certification and validation efforts
- Separation of Concerns
Challenges and Requirements (Economical)

- Change from one application leads to one chip/device approach
- Adapting of the development process in automotive
- Develop efficient business models and define clear liability
  - Sharing of platform costs, liabilities, warranty, etc.
- Define, manage and certify policies for applications
- Identify minimum set of needed standards
Challenges and Requirements (Technical)

🌳 Different vehicle applications need different dependability and security measures
  - Protect applications from external influences
  - Protect vehicle from destructive or malfunctioning applications
  - Assure dependability and timing constraints for applications

🌳 Applications access many communication networks in and outside of the vehicle
  - Providing dependable enforcement of access policies

🌳 Dependable and Secure by design
  - Provision of security and dependability without loosing flexibility and openness
OVERSEE FACTS

Open Vehicular Secure Platform

Europe Commission FP7 Project
- 8 Partners
- Granted appr.3 Million Euros
- 30 Months – Ending October 2012

Jan Holle
OVERSEE as an Open and Secure In-Vehicle Platform for ITS (Objectives)

- Standardized interfaces/runtime environments (partitions) for application development
- Runtime environments for applications with different security/dependability/timing assurances
- Policy based access control to resources
  - Secure single point of access to communication networks
- Security by design
  - Minimize attack surface
  - Attack/Error Resilience
  - Separated domains for policy enforcement and applications
  - Secure integration of a Hardware Security Module (HSM)
  - Security anchor is an essential part of the platform
OVERSEE as an Open and Secure In-Vehicle Platform for ITS (Resource Access)

- The Secure I/O Partition
  - Interfaces the real I/O devices
  - Enforces access policies for every I/O device (or address, signal etc.)
  - Provides virtualized I/O devices to the user partitions

- Device drivers and policy enforcement are executed in a system partition isolated from the other runtime environments.
  - Applications and guest partitions don’t have direct access to I/O devices
OVERSEE as an Open and Secure In-Vehicle Platform for ITS (Architecture)

Secure I/O Partition
- Policy
  - HW Driver
  - OS

User Partition
- Virtual Driver
- OS

User Partition
- OVERSEE API
- RTOS

Security Services Partition
- Security Services

Communication Channels / Shared Memory

Virtualization Layer (XtratuM)

HW Devices
- I/O Interfaces
- CPU
- Memory
- HSM

Hardware Platform
Standardization and Consensus Building

Need for an consensus building process
- Among all stakeholders (OEMs, public authorities, users, application developers, service providers)
- Agreement on an understanding and the requirements of open platforms

Need for standardization
- ITS Action plan: “Open in-vehicle platform architecture”

“Minimum” set of “Standards”
- Interfaces for resources and services on runtime environment level
- Runtime environments for application development
- Application certification process
- Performance requirements
- Dependability requirements
Conclusion & Outlook

Open platforms could be a solution for some current and upcoming challenges in the (automotive) industry.

Open platforms offer the potential to build a new market for software and services based on standardized RTEs.

Open platforms offer a lot of potentials while there are also a lot of challenges that have to be solved, before real world deployment could take place.

The architecture of OVERSEE could help to define a minimum level of requirements for open and secure in-vehicle platforms, to assure security, dependability without losing flexibility.

A consensus building process and a standardization process should be started immediately.
Thank you for your attention!

For further information visit our project homepage

www.oversee-project.com

Acknowledgment: OVERSEE is a research project funded by the European Union within the FP7.
Virtualization

- The hypervisor provides an additional abstraction layer between HW and RTE’s
- Multiple isolated runtime environments (partitions) on same platform
  - Time-shared CPU resources
  - Assigned memory areas
  - Assigned I/O Interfaces
- Different OS can run parallel
  - Linux, MeeGo, OSEK OS, Bare (RTE without OS, executing e.g. native C code)
- Timing constraints at the RTE level can be assured by the static scheduler of the hypervisor
General Structure – Security

Ĥ HSM (Hardware Security Module)
  ▪ Provides a trust anchor for Secure/Authenticated Boot
  ▪ Secure storage for keys
  ▪ Accelerated cryptographic services

Ĥ Similar as with the I/O devices no direct access to HSM from RTE’s possible
  ▪ Services / keys can be shared according to access policies

Ĥ The hypervisor can take actions depending on secure boot result
  ▪ Stop tampered partition only
  ▪ Run partition but restrict access to networks
  ▪ Attest external entities or the partitions the result
  ▪ Etc...
Design of an Open Platform: Information Flow

Secure I/O Partition
- Management Module
  - Driver
  - Service Binding

Virtualization Solution

Service Binding API or virtual Driver

Hardware Platform

Communication Module

Driver

OS

RTOS

Service Binding API