HoliSec

Automotive Security and Privacy

Holistic Approach to Improve Data Security

Report type	**Deliverable D1.1**
Report name	**Security & privacy for connected & automated vehicles**
Dissemination level	**Public**
Status	**Released**
Version number	**1.0**
Date of preparation	**2016-12-15**

Authors

Authors	E-mail
Henrik Broberg	henrik.broberg@volvocars.com
Lars-Olof Berntsson	lars-olof.berntsson@volvo.com

Contributors	E-mail
Andreas Bokesand	andreas.bokesand@volvo.com
Atul Yadav	atul.yadav@volvo.com
Boel Nelson	boel.nelson@chalmers.se
Christian Johansson	christian.johansson.13@consultant.volvo.com
Christian Sandberg	christian.sandberg@volvo.com
Eva Carlsson	eva.carlsson.2@volvo.com
Henrik Wärmlind	henrik.warmlind@volvo.com
Joakim Pauli	joakim.pauli@volvo.com
Lars Magnusson	lars.magnusson@volvocars.com
Malte Isacsson	malte.isacsson@volvocars.com
Mathias Westlund	mathias.westlund@volvocars.com
Mathias Widman	mathias.widman@wirelesscar.com
Mattias Jidhage	mattias.jidhage@assured.se
Riccardo Scandariato	riccardo.scandariato@cse.gu.se
Tomas Olovsson	tomas.olovsson@chalmers.se

TC Review	E-mail
Jörgen Borg	jorgen.borg@volvocars.com
Peter Folkesson	peter.folkesson@sp.se

Revision chart and history log

Version	Date	Reason
0.1	2016-10-10	1st draft version
0.2	2016-10-14	Updated after WP1 meeting
0.3	2016-10-21	Updated after HoliSec internal review
0.4	2016-10-28	Draft version sent out for review among WP1 partners
0.5	2016-11-11	Updated after second review among WP1 partners
1.0	2016-12-15	First release

Table of contents

1 Introduction

Modern and future vehicles not only communicate with each other to solve critical traffic situations, they also provide many convenient functions to the car owners, drivers, and passengers and are part of the Internet of Things (IoT) community.

It is essential that security and privacy problems do not hinder the development of new functionality or cause harm. With the advent of connected and autonomous driving, data security is of increasing importance to the automotive industry; however data security is very hard to address despite being directly linked to some of the most important attributes of the automotive industry, quality and safety.

Best practices for security includes functions like analyzing risk at the different life cycles, security by design, monitoring or threats and vulnerabilities, respond and recover as incidents response and other core capabilities that supports the organization (awareness training, governance etc.).

As with system functional safety, it is beneficial to apply a rigorous, risk driven system security engineering process to the development of automotive EE systems, reducing the likelihood of successful incidents.

Besides a rigorous system security engineering process, there are various classes of countermeasures and mechanisms to tackle threats associated with security attributes:

Forecasting: Countermeasures or security measures/mechanisms that aim to take into account current situations and experiences to estimate future incidents and possible consequences.

Prevention: Defence approach for prevention i.e. countermeasures or mechanisms that aim at preventing attacks. The approach ensures that unwanted modifications, reads, and injections of messages are prevented. No matter how invincible a system may seem, it's vulnerable. This is the first rule that any professional in contact with data security should assume. While the risk may be very low, there's no product or practice that can completely eliminate the chance of a data security-related incident. Prevention software, hardware, and processes are typically thought of as the core of data security.

Detection: Defence approach for detection i.e. countermeasures or mechanisms that aim at detecting and alerting on potential attacks. The approach ensures that detective measures are taken to reveal the presence of attacks and intrusions that have compromised or circumvented preventive mechanisms. While prevention is undeniably important, a system is still apt to fall prey to an attack. If an attack occurs, it's of utmost importance that the presence and activities of the attacker can be detected and recorded. With proper detection technologies in place, a system can react automatically to the existence of a malicious presence.

Isolation: Countermeasures or security measures/mechanisms that aim to identify the root cause of an attack and limit the consequences to a specific set of assets (origin of the attack) by protecting other assets to be affected by the same attack. As a result, isolation refers to identification of root cause as well as containment similar to the notion of “fault isolation and containment”.

Recovery: Defence approach for recovery i.e. countermeasures or mechanisms that aim at transferring the system into a controlled state (can be a completely recovered state where nominal service is provided; or degraded state where a limited or no service is provided). It is recommended to incorporate the recovery plan, including both the backup procedures and the recovery procedures, into the initial risk-avoidance planning. This should form the foundation on which everything else is built, and brings things full circle. When all else fails, a solid recovery is extremely important.

Forensics: Countermeasures or security measures/mechanisms that allow investigating teams to learn from successful attacks: what has happened and how it happened. This can help to reconstruct the events and revisit existing countermeasures.

1.1 Scope and purpose

Security engineering is an engineering discipline concerned with securing a system and it encompasses the entire process from system design to deployment and maintenance; however the influence of data security mechanisms on safety still needs research and investigations.

To address this, the HoliSec project was launched in April 2016. The project targets a holistic approach to improve data security of safety-related embedded systems.

The objective of the HoliSec project is to design, to verify, and to prototype modular, (cost-) efficient security solutions targeting the automotive context, aiming at enabling development of (cost-) efficient security functions that copes with the special automotive requirements and restraints e.g.

- limited bandwidth of the on-board networks,
- limited microcontroller resources (processing performance) for security algorithms,
- real-time capabilities,
- scalability.

The security solutions are anticipated to be based on hardware-based security anchors and software security layer. The solutions may apply e.g.

- secure boot,
- secure memory,
- secure security artefacts processing,
- run-time environment partitioning, and
- secure communication.

The purpose of this document is to explore the security and privacy needs and requirements in order to serve as a base for the other work packages within the HoliSec project.

The scope of this deliverable is to investigate the needs and requirements for: (1) increased levels of connectivity and automation in vehicles, (2) access methods with both permanent and temporary access to the vehicle and its parts, and (3) data sharing and big data according to user consent in collaborative ITS, predictive maintenance and product improvement [1].

The scope is not limited to the security products, but also aims to cover processes and people that can counter risks.

This document will give a short introduction to the subject of security and privacy in vehicles (chapter 3) and define some of the questions that need to be addressed in developing, manufacturing, service and operations of vehicles. A selection of use cases is explored (chapter 4) to elaborate needs and requirements.

The main method for capturing needs and requirements was a workshop with cross domain representation, which is reflected in this document.

This document does not have the ambition to cover state of the art since this is covered by deliverable D1.2

Requisite document is deliverable D1.1: Needs and requirements; Version 2.0 from the former FFI HEAVENS project – HEAling Vulnerabilities to ENhance Software Security and Safety; Vinnova Dnr 2012-04625 [2].

The Consortium

ArcCore AB	Assured AB	Chalmers
SP Technical Research Institute of Sweden	Viktoria Swedish ICT	Volvo Car Corporation
Volvo Technology		

2 List of industrial security needs

3 Basic security attributes and threats

This chapter identifies some general research topics (RT) related to security and is also intended to be an introduction to the subject of managing security and privacy risks. For more information refer to HEAVENS Security models, deliverable D2 [9].

The question “is the vehicle secure” is common but both naïve and dangerous since it is indeterminate. The answer can be both yes (yes, given we made no mistakes) or no (no, we cannot prove we made no mistakes) for any given system. There is an obvious risk that the binary nature of the question will lead to misunderstanding.

A better determined question to serve as “go/no go” criteria for a product launch and to address in the HoliSec project is:

RT-SE-1 ***Are the risks with the product or service manageable?***

Fundamental concepts and taxonomy of information and computer security can be helpful to understand risks and how to counter them. As described in HEAVENS Security models, deliverable D2 [9] risk is a function of assets and threats. The harm to stakeholders when assets are compromised will set the impact level. The threat level is a function of properties of the system such as exposure and the presence of vulnerabilities and properties of the threat such as window of opportunity, capabilities and motivation.

The term security level is defined as a metric of risk in order to give a sense of priority and provide a basis for setting the required countermeasures and assurance level. Per definition risk can only be avoided by removing the feature (thereby the opportunity with the function is also lost). There are still little guidance how to assess what an acceptable residual risk is from legal, company or moral perspective.

RT-SE-2 ***What is an acceptable residual risk to the stakeholders (i.e. user, owner, company, society)?***

HEAVENS Security models, deliverable D2 [9] also provide definitions for security attributes associated with the assets. When the security properties of the asset are understood, security goals can be elaborated.

RT-SE-3 ***What are a set of complete and correct security attributes for the assets at risk?***

Given the constraints of available resources (money, competence, time etc.) how can risk identification and treatment be incorporated in the development, manufacturing, service and operations when it is a fact that mistakes happens and security is only as good as the weakest link. No pressure, just make no mistakes.

What are efficient quality controls for avoiding and detecting vulnerabilities (internal faults) in design, implementation and configuration of a system?

RT-SE-4 ***What controls (process, products & people) are efficient (cost, completeness, correctness) to identify and control risks?***

One fact in automotive EE system development is that the lead times are long for some things. Changes to the topology (ECU deployment) might be infeasible and deployment decisions may take decades to phase out. It is then extremely valuable to filter out the critical design decisions like providing isolation between critical assets and threats by a proper domain separation. System engineering efforts aims to efficiently and reproducible seek out critical decision point in product development and operations and applying methods and effort to avoid vulnerabilities in design, implementation or configuration.

In an operational phase there is a need for reliable indicators of emerging risks or actual incidents to enable an appropriate response. How can a defender differentiate natural noise factors from intelligent noise created by a threat actor?

RT-SE-5 ***How can risks be identified as early as possible with high reliability?***

In information and computer security it is important with administrative controls but the emphasis in the HoliSec project will be on EE system development. In HEAVENS the threat model of choice was STRIDE since it was deemed suitable for analysis of EE systems in vehicles and the associated IT systems. Below follows a general application STRIDE to a vehicle EE system to identify question around what is an efficient set of security functional requirements in the vehicle EE system and associated IT systems to counter risks.

3.1 Preventing violation of authenticity (spoofing and elevation of privilege)

Authentication is the process or activity that increases confidence that someone or something is genuine. For example, entering a password increases the system´s confidence that the person behind the keyboard really is authorized to use the system.

The threat definition, from [9], of *spoofing* is:

- **attackers pretend to be someone or something else**

The threat definition of *elevation of privilege* is:

- **attackers perform actions they are not authorized to perform**

The spoofing and elevation of privilege threats violate the authenticity property.

Countermeasure areas for spoofing/elevation of privilege threats are: 1/ Use strong authentication 2/ Do not store secrets (for example passwords) in plain text. 3/ Do not pass credentials in plain text over wire.

Signing of messages is an alternative to encryption to counter spoofing. Concerning elevation of privilege, it is also up to the whole system to require authentication for some operations.

RT-SE-10 ***How to efficiently restrict services to authorized users?***

RT-SE-11 ***How to make it impossible to enable more features in the system than what were bought?***

RT-SE-12 ***How to protect remote software download from unauthorized usage, replay, modification, spoofing?***

RT-SE-13 ***How to provide authorization for determining tool privileges – fleet owner/dealer/workshop, mitigating threats of exploiting vulnerability or possible implementation errors to elevate access privileges to the assets?***

RT-SE-14 ***How to prevent unauthorized usage of aftermarket tools?***

RT-SE-15 ***How to enable the possibility to share/sell repair and maintenance information RMI to independent parties?***

RT-SE-16 ***How to provide the sender ECU/sensor authentication, mitigating the threat with impersonating sender of message?***

RT-SE-17 ***How to make Applications (Apps) authenticate themselves before getting access to vehicle APIs?***

3.2 Preventing unauthorized modification (tampering)

The threat definition, from [9],of tampering is:

- **attackers change data in transit or in a data store**

The tampering threat violates the integrity property. There are integrity threat risks on the complete system (on-board – off-board).

There are basically three main countermeasure areas for tampering threats:

1/ relying on system defenses such as permissions, 2/ use of cryptographic mechanisms, 3/ and use of logging technology and audit activities as a deterrent.

Permission mechanisms can protect things that are within their scope of control, such as files on disk or data in a database. Cryptographic primitives for integrity often use hashes and signatures. A hash takes an input of some arbitrary length, and produces a fixed length digest or hash of the input. Ideally, any change to the input completely transforms the output. A signature is a cryptographic operation with a private key and a hash.

Logging technology and audit shows how files change, possible being able to recover from integrity failures.

RT-TN-10 ***How to protect information held in the vehicle from unauthorized modification and deletion (data at rest)?***

RT-TN-11 ***How to prevent tampering of service history?***

RT-TN-12 ***How to prevent tampering of crash information?***

RT-TN-13 ***How to prevent tampering of driver reports?***

RT-TN-14 ***How to protect information against unauthorized or malicious modification or manipulation during transmission i.e. integrity shall be secured of data in transit?***

3.3 Preventing violation of accountability (repudiation)

The threat definition, from [9], of *repudiation* is:

- **attackers perform actions that cannot be traced back to them**

The repudiation threat violates the accountability property.

The technical elements of addressing repudiation are fraud prevention, logs, and cryptography. Fraud prevention is sometimes considered out of the scope of repudiation.

The key tools for non-repudiation are logging and digital signatures.

There are basically two main countermeasure areas for repudiation threats: 1/ Create secure audit trails. 2/ Use digital signatures.

RT-NR-11 ***What data security mechanisms need to be implemented to prevent manipulation of audit logs (or ECU fingerprints)?***

3.4 Preventing violation of confidentiality and privacy (information disclosure)

The threat definition, from [9], of *information disclosure* is:

- **attackers get access to data in transit or in a data store**

The information disclosure threat violates the confidentiality and privacy properties.

Data protection laws exist to strike a balance between the rights of individuals to privacy and the ability of organisations to use data for the purposes of their business.

Before collecting personal data, a service provider or content provider should obtain the prior and unambiguous consent of the data subject, or inform the data subject of the collection of personal data and the indicated purposes of use according to domestic regulations.

There are needs for the data subject to have its vehicle and movement data protected for privacy reasons, and in the case of companies, for competition and/or security reasons.

There are countermeasures areas for information disclosure threats: 1/ If the content of the communication must be protected, then traditional cryptography may be sufficient. 2/ If the system controls all access to the data, a permission system can be used 3/ Encryption of stored data, prevents physical theft and access bypassing the system.

RT-ID-10 ***How to protect sensitive information from unauthorized access and shall not be revealed to any party not authorized?***

RT-ID-11 ***How to avoid that sensitive vehicle information do not leak from vehicle to nomadic devices?***

RT-ID-12 ***How to prevent new owners of vehicles from viewing personal data belonging to previous owners of the vehicles?***

3.5 Preventing absorbing of resources (denial of service)

Within the context of standardization there are a number of objectives for data security that are intended to ensure availability.

The threat definition, from [9], of *denial of service* is:

- **attackers interrupt a system's legitimate operation**

The denial of service threat violates the availability property.

Denial-of-service attacks work by exhausting some resource. Traditionally, those resources are CPU, memory (both RAM and hard drive space can be exhausted), and bandwidth. Denial-of-service can also exhaust human availability.

Denial-of-service attacks can be split into those that work while the attacker is attacking (say, filling up bandwidth) and those that persist. Persistent attack can remain in effect until a reboot, or even past a reboot.

There are basically two main countermeasure areas for denial of service threats: 1/ Use resource and bandwidth throttling techniques. 2/ Validate and filter input.

The below DoS needs primarily applies to information / data / services that are inside the vehicles, but can also be applicable to data that is stored / transferred to other devices, such as workshops, the corporate network or even in the cloud.

RT-DoS-10 ***How to avoid normal system functions to be affected by malicious activity?***

4 WP1 workshop around needs for automotive security and privacy

The industrial needs for automotive security and privacy, as described in this deliverable, represent security aspects that HoliSec processes and technologies should comply with, or at least relate to, while HoliSec proposes new technologies and ways of working.

More detailed requirements for the other WPs in the HoliSec project (WP2, WP3, WP4, WP5) and its results must therefore comply with these overall industrial needs.

To initiate the collection of industrial needs, the WP1 had an internal Workshop that took place on Tuesday October 4 2016 at AB Volvo Premises, Conf. Room M1:103, Lundbystrand, Gothenburg.

The WP1 Workshop identified and discussed different industrial security topics that will be the basis for the further work in the HoliSec Project related to:

- Autonomous Driving (Documented in Chapter 4.1),
- Connectivity – Telematics & Service platform (Documented in Chapter 4.2),
- Anonymous and Privacy Mechanisms (Documented in Chapter 4.3),
- Wired Diagnostics / Remote Diagnostics (Documented in Chapter 4.4),
- Big Data / Data Management (Documented in Chapter 4.5), and
- Security Engineering Process - right solution with the right cost (Documented in chapter 4.6),

4.1 Needs related to Autonomous Driving (AD)

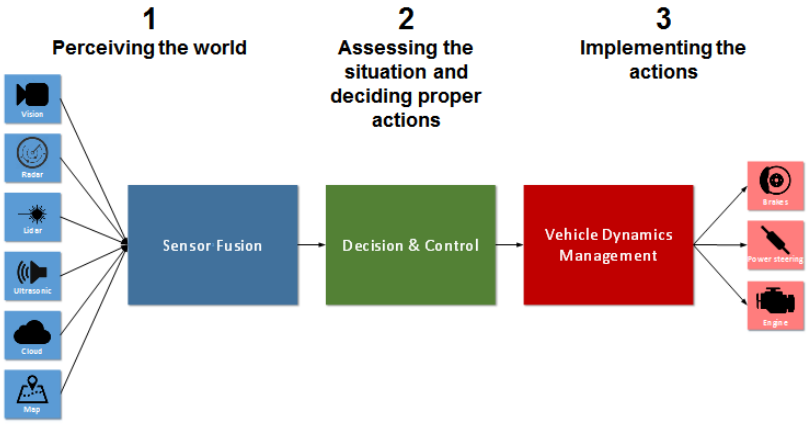
Highly automated vehicles (HAVs) utilizing technologies that corrects for human mistakes, or that takes over the full driving responsibility, have the potential to dramatically decrease the number of crashes tied to human choices and behavior.

The security challenge

- Keeping the vehicles connected is a safety risk with worldwide exposure
- Driver out of the loop with very limited controllability of the vehicle
- Traditional functional safety accidents occurs randomly in time and hit limited number of vehicles at same occasion
- A security breach may cause simultaneous accidents to ALL connected vehicles

4.1.1 WP1 workshop discussions

There was a discussion about trust in different components. The picture below describes logical architecture for AD:



Sensors as well as off board support with high resolution maps are needed.

Strategy is to not drive further if you are not confident on sensor input

- Machine learning and artificial intelligence are used
- Limit scope of the safety function
    - No dependence on information from other vehicles

There are needs for over the air (OTA) updates in order to account for new vulnerabilities.

RT-AD-10 ***What needs to be done to make sure the driving logic cannot be compromised?***

Data could be tainted with faulty data or exploits.

RT-AD-11 ***How do we keep concepts up to date when threats evolve?***

Is assuming that central repository of map data could be compromised a reasonable threat?

Probe sourced (Crowd sourcing) of map data could open up for privacy concerns for uploading of data as well as tainting of data.

RT-AD-12 ***What are foreseeable threats and what should be done to control the risks?***

Artificial intelligence (AI) systems

- Never ending learning system
- How should one control the learning process?
    - Safety net that is static
    - Dynamics that can learn

Mode confusion is not covered in ISO 26262 [4]

Scalable attacks are worse for an autonomous platform.

Intellectual property rights stolen

- Not so much of an operational attack for vehicles in the field
- Not constrained to the feature, but could be spy in organization
- Ransom
- Does this really increase with AD?

Liability if something happens?

- Logs etc. with non-repudiation etc.
    - Scenario replay for forensic investigation
    - Security of log data is important

What is the log data required to do forensic investigation?

- Not Everything
- In case of accident
    - Same as a police investigation
- Information to make the system better
    - Furher details are needed..
- When relying on outside information what log level
    - Volvo still need to be responsible for the amount of trust

4.1.2 UC-AD-01 Vehicle on demand

ID	*UC-AD-01*
Title	Vehicle on demand
Version	1.0
Status	Released
Motivation	AD in all scenarios with occupants, with cargo, but also completely without any payload. Vehicle available at any requested location. Passengers use the travel time completely independently for other activities than performing the driving task. The compartment is designed completely independently from any restrictions of a driver workplace, whatsoever. Cargo can be transported continuously for 24 hours a day, as long as not restricted by the energy supply for driving.
Threats	Examples of possible security attacks Items outside vehicle ▪ Access vehicle software to control motion ▪ Access cloud services and supply false map data ▪ Create false data sources (probe vehicles) to cloud services ▪ Change infrastructure e.g. paint new lane markings ▪ Spoof sensor data Items in vehicle ▪ Embed malicious code during development ▪ Install malware at service

4.2 Needs related to Connectivity – Telematics and Service platform

Cooperative intelligent transport systems (C-ITS) and specifically vehicle to x communication (V2X) are challenging areas, e.g. it is important to know if the one sending the info is authorized, thus authentication is a critical requirement.

These systems are a mix of commercial and private solutions communicating over both private and public networks enabling both telematics and infotainment services. The nodes in the network must verify that the received message is sent from an already known trusted source. Nodes must make sure that any intentional or unintentional modification to messages is detected. Many new services will be introduced in the coming years. Some services target drivers and passengers such as navigation and driver assistance systems, and other focus on the vehicle itself such as remote diagnostic and remote software updates. Most vehicle manufacturers have plans to offer a fairly large number of services and now face the challenge to implement new functionality without sacrificing traffic safety. If an attack is not prevented i.e. if data security problems have been detected, then the vehicle must operate in a degraded and fail-safe mode. Overall, technical limitations need to be considered, e.g. not having too long encryption keys since it will drive hardware costs and can reduce performance.

Data security operations require storing of credentials that are used for different cryptographic purposes such as encryption, signature verification etc. The storage must be tamper proof against unauthorized access such as memory read outs and side channel attacks. Hardware Security Modules (HSM) sometimes has cryptographic processors/accelerators, which are designed to do certain types of data security operations more efficient than the application processor. The data security overhead of the messages for a Public Key Infrastructure (PKI) solution tends to be large. For scenarios where the messages are set to transmit with high frequency, packet loss rate will increase, and might endanger safety critical systems.

There is a high demand for applications and services around the connected vehicle, something similar to an “AppStore” for the vehicle where the owner, driver, and passengers can chose to install both free apps and subscribe to different services. However, there are obvious needs for integrity when allowing apps and third party applications in the vehicle, which also means data security with respect to remote access (wireless). A severe risk is if messages from nomadic devices lead to safety reaction in a vehicle. It is important to note that the OEM might not be the only one creating apps related to vehicle services. Partners and third parties create apps as well.

Hardware Security Modules are sometimes necessary, sometimes to help for maintaining secret keys, provide key generation, signature generation/verification etc.

4.2.1 WP 1 workshop discussions

RT-TS-10 ***How to handle trust between entities?***

Which technologies can be used? Asymmetric/symmetric keys? Nomadic devices, Google/Apple accounts? How to manage trust over time (how to deal with trust in an evolving environment)? How to verify trust in offline/semi-online environments?

Discussion notes:

- Impact is unique for the automotive industry - safety impact!
- Similarities to other industrial domains, e.g. supervisory control and data acquisition (SCADA) - but they can choose offline/segmented environment, vehicles are in an unknown environment.
- Automotive must support commercial services and entertainment/infotainment. Should they be mixed at all?
- It's driven by customers; "fun" features sell vehicles. There needs be a middle ground?
- How do we separate functions? Can we use the same network or do we need parallel?

RT-TS-11 ***How should the OEMs define the service side assuming that the apps are not fully secured?***

The vehicle network spans the whole vehicle and consists of networks of different bus-system technologies. New communication media in vehicles, like Ethernet is well recognized in the “hacker” community. Wireless sensor networks (WSNs), for example communication with tire pressure sensors, introduce a new range of potential data security issues. Connection points (wireless, ftp, usb, etc.) are important to analyze, to control and secure the complete chain. Communication with the vehicle-outside world can be done with many different technologies.

RT-TS-12 ***How to maintain security over product life cycle?***

How to deal with security patches?

- For how long can/must we support vehicle security patches?

How to keep up with security evolution (crypto agility etc.)?

- How can we design agile security in software?

How to deal with hardware constraints? How to design hardware that facilitates agile security?

Discussion notes:

- Can we "end-of-life" a vehicle like software suppliers do?
- Can it be equivalent to brakes; customer needs to replace telematics systems on regular intervals?
- Is this a technical problem or a sales/policy problem?
- Should ECUs have a "kill switch" which disconnects a vehicle once "end-of-life" occurs?
- Are there features/dependencies that prevent us from disconnecting a vehicle?
- Can subscription based service offerings pay for security upgrades?

RT-TS-13 ***Secure architecture for connectivity platform?***

How to design a secure automotive connectivity platform?

- Layered security?
- Separation of privileges?
- Secure network topologies?

Connection points (wireless, ftp, usb, etc.) are important to analyze, to control and secure the complete chain.

RT-TS-14 ***How to secure the complete chain, the remote services over the air (OTA) must be protected to work in public network and not only private.***

Main takeaway from discussion is: How can we add security to a system that was not designed with security in mind (CAN, etc.)?

4.2.2 UC-TS-01 (Nomadic device) Install application

ID	*UC-TS-01*
Title	(Nomadic Device) Install applications
Version	1.0
Status	Released
Related Use Cases	EVITA UC-01 to UC 13, Specification and evaluation of e-security relevant use cases, Version 1.2 [7]
Motivation	Installing and running applications in the car from an external device. These can be used through modules by the driver or occupants.
Threats	The security and integrity of the installed application has to be ensured. Besides user authentication for the installation process there must be systems of protection like certificates. Furthermore it should be assured that the nomadic device can only be used by authorized persons, e.g. by using identification systems to start the software system for application installation.

4.3 Needs related to Anonymization and Privacy Mechanisms

Some data is open and public but some data is private and need to be protected. For instance, there are needs for the data subject to have the data concerning their personal vehicle (and its movement) to be protected for privacy reasons. In case of company vehicles (or fleets), some information might even be regarded as intellectual property and, hence, to be protected against the competitors (although this is not strictly related to privacy).

Privacy and security, however, cannot be treated independently.

For instance, the information disclosure (security) threat violates both confidentiality and privacy properties. Information disclosure can happen with information at rest (in storage) or in motion (over a network). The information disclosed can range, e.g., from the content of a communication to the existence of an entity with which someone is communicating.

Privacy properties, of course, go beyond data confidentiality. As such, data protection laws exist to strike a balance between the rights of individuals to privacy and the ability of organisations to use data for the purposes of their business.

Before collecting personal data, a service provider or content provider should obtain the explicit and informed consent of the data subject, or inform the data subject of the collection of personal data and the indicated purposes of use according to domestic regulations.

Furthermore, future regulations (like the General Data Protection Regulation) will demand that Privacy by Design approaches are adopted, which requires that data protection and other privacy-oriented concerns are addressed into the development, and not as an afterthought.

A car has a lot of sensors and can collect lots of data, which could be personal data. How can we develop new features while preserving privacy? Further, regulations will impose additional constraints and obligations, like:

- Data minimization
- Privacy by design
- Privacy by default
- GPS-trajectories are sensitive information
- Vehicle data can be used to fingerprint drivers (by analyzing accelerator and brake signals from CAN bus)

To preserve privacy, several mechanisms can be used:

- Obfuscation: generalization, perturbation
- Encrypt
  - Suppression
  - Abstract: e.g., one can group or mix GPS trajectories

4.3.1 WP1 workshop discussions

Key questions that need to be addressed are:

*Technical solutions exist, but do they work?

*Is true anonymization possible?

*How to quantify privacy?

*Anonymization just becomes an issue as soon as you retrieve data and store it. Is this true? You can store any personal data on a phone, can you do the same with a car? Does the driver really own the hardware of the car? It might also be a problem if someone borrows your car.

RT-ID-17 ***Sensors in vehicles produce a lot of data that can be sensitive for drivers, passengers as well as 3rd parties outside the vehicle. How should we address these issues while still being able to develop secure, convenient and safe features for the connected vehicles?***

RT-ID-18 ***Privacy by design, both onboard and off-board is a best practice but what kind of mechanisms for anonymization and privacy are available***

What do we do with old data, such as data from diagnostic read outs (DROs)? Strategy for the future?

*GDPR also has obligations, such as the right to be forgotten and data breaches have to be communicated to affected users. Some are very technically challenging.

*Are we losing the battle with anonymization? Maybe transparency regarding how data is processed is a better way to go?

*Is it enough to remove the VIN? No, there are more identifiers. We can also get the GPS data. There's also the fingerprinting study mentioned earlier.

*The workshop said they store the error codes from DROs (plus some additional data about the service event). Is this privacy-preserving enough?

*Who owns the data? It depends on if it is personal data or not.

*Regarding DRO in the future, there is a proposal to process raw data on-board.

*Can we use differential privacy? Do we need to be able to identify the driver? Maybe it's enough to identify the number of people that does something?

We're usually not interested in the driver, we're interested in the vehicle. However, if we can identify the vehicle, this might also identify the driver.

*What if we need to remove one user from anonymized data? If it's truly anonymized the right to be forgotten isn't applicable. However, if we can't promise true anonymization we would need to be able to re-identify and remove individuals.

RT-ID-19 ***Removal or anonymization of diagnostic read out (DRO) data is required for stored data if consent cannot be obtained in arrears. Is removing or obfuscating VIN enough?***

RT-ID-20 ***Does diagnostic sets of live in-vehicle network traffic constitute sensitive***

4.3.2 UC-ID-01 Active brake (Privacy vs security)

ID	*UC-ID-01*
Title	Active brake
Version	1.0
Status	Released
Motivation	The vehicle receives a message that indicates that the vehicle is in immediate danger of collision with an object. The only way to avoid the collision is an instant brake manoeuvre.
Threats	– The privacy of the driver of the vehicle broadcasting the information has to be guaranteed. – At the same time, the information received from another car needs to be evaluated regarding integrity and trust (e.g. authenticity of data). This might require a trade-off between security (or safety) and privacy.

4.4 Needs related to wired diagnostics / remote diagnostics

The vehicle network is easily accessed through the standardized OBD-II port present in all vehicles. The OBD-II port can be used to check vehicle configurations, change settings, update software, and to read diagnostic error messages.

It is a legislative requirement in several markets that law enforcement shall be able to connect a universal tool to a vehicle and check emission levels, diagnostic trouble codes (DTC), on-board monitoring test results etc.

We distinguish between two different types of diagnosis: On-Board Diagnostics and Off-Board Diagnostics. The difference is that an Off-Board diagnostics is done with an off-vehicle system (e.g. diagnostics tool). It is important to mention that an Off-Board Diagnostics can be done by connecting the diagnostics tool to the On-Board Diagnostics system.

For better understanding, a few words about failure log entries. Each ECU has a diagnostics routine, which records failure events (e.g. sensor failure) in the failure log. Since the failure events can be sensitive for different ECUs, different failure records are made. A diagnostics tool will try to know where the real cause comes from, based on two points: the different entries made in different timeframes and the algorithms implemented.

Vehicles have onboard diagnostics interface for accessing its electronics and software. This is primarily used for fault tracing and maintenance. Different tools are used to connect to this interface both OEM owned and third party tools. Many of these tools are available on the market for anyone to use. There are legal requirements for access mechanism, for example, in the form of communication protocols publically available for interaction between tool and vehicle. Such an access to the vehicle is often wired but wireless options are available. The ISO 14229 [8] specifies simple security mechanisms for unauthorized access e.g. in the form of challenge-response protocols. These measures and not sufficient to meet the security needs for future.

A more exhaustive way to systematically identify security needs and requirements in this area is to apply a TARA methodology similar to one proposed in HEAVENS project and which is being updated as part of this project. This will lead to creating more detailed security requirements.

However, the security problems related to diagnostics and software download in existing vehicles are well described in many previous studies. And based on that some of the most prominent security problems are described here. The list is not complete but can be seen as the areas where immediate attention is needed.

4.4.1 WP1 workshop discussions (Wired Diagnostics)

RT-DS-10 ***How do we authenticate diagnostic tool to the vehicle and vice-versa?***

Considering mechanism based on pre-stored keys or certificates, there are challenges around

- Precaution for lost keys
- Lifetime and procedures for updating keys/certificates
- Procedures for revoking
- Choice of authentication protocol keep in mind resource constraints of ECU
- Solution that works offline for extended periods

RT-DS-11 ***How do we secure the integrity of the communication between diagnostics tool and vehicle?***

Once the vehicle and tool have authenticated to each other, how do we ensure that the communication is carried out further in secure way for example using encryption. How do we secure DoIP communication between Tester and Gateway/ECU?

RT-DS-12 ***How to handle the access management?***

Access managements is done to make sure that persons or computers only have as much access as they really need i.e. how to ensure that the access to diagnostic services and data is enforced. For example, security principle of “least privilege”, which means that each user must only be able to access the information or resources necessary to do their job.

RT-DS-13 ***Which security mechanism should be implemented at what layer of the OSI model?***

- What are the security requirements for diagnostics or DoIP protocol stack?
- Should security mechanism be implemented as far down the network stack or mostly at the application layer?

For example, possible security mechanisms at different layers of the network stack can be ISO 15764, SSL/TLS, IpSec, VLAN, Cryptographic Link Layer, etc

RT-DS-14 ***What are the architectural impacts to security on diagnostics?***

- Should there be only one point of information flow into a vehicle so firewalls and intrusion detection systems can be implemented in one place? Should there be only one DoIP Edge node?
- How many DoIP nodes should be there in a vehicle and which of them should be exposed (addressable directly) outside a vehicle?

Which security mechanisms should be implemented in boot loader or application software of an edge node?

4.4.2 WP1 workshop discussions (Remote diagnostics)

Secure remote diagnostics enables a diagnostic tool to start a DoIP session with a vehicle over a mobile network connection. The vehicle can be diagnosed without the need for a visit to the workshop. When connected over the network the car appears to the workshop tool as if it was connected locally to the workshop network. Remote diagnostics must be initiated by both the user in the vehicle and from the technician in the workshop.
A cloud service provides a discovery mechanism to match vehicle and the diagnostic tool. A central network server enables a network connection for the DoIP session.

Drivers give consent via terms and conditions for each session. Storage of event log is required. The discovery mechanism is secured in the same way as all connected services using the cloud. The network is used for communication security of the DoIP session. The user is involved in initiating the session and a one -time session code makes sure the right diagnostic tool station connects to the right vehicle.

Only cars are allowed to initiate a remote diagnostic session (Authentication Authorization)
- How it works: The car acts as client and connects to the remote diagnostic broker to request network connection information.
- Why it works: The car does not act as server to allow incoming requests from external clients.
- What it prevents: That someone initiates a remote diagnostic session with the car without the control of the car user.

Only authorized cars are allowed to establish a remote diagnostic connection (Authentication Authorization)
- Security mechanism: Network with certificate authentication
- How it works: The network server authorizes the car by validating the car's client certificate.
- Why it works: The network server will not allow a communication link to be established unless the car certificate is authentic.
- What it prevents: That someone establishes a network tunnel to Volvos remote diagnostic broker

Only authorized diagnostic clients are allowed to establish a remote diagnostic connection (Authentication Authorization)
- Security mechanism: Network with certificate authentication
- How it works: The network server authorizes the diagnostic client by validating the client's certificate.
- Why it works: The network server will not allow a communication link to be established unless the client certificate is authentic.
- What it prevents: That someone establishes a network tunnel to Volvos remote diagnostic broker

The remote diagnostic broker is an authentic Volvo service (Authentication Authorization)
- Security mechanism: Network with certificate authentication
- How it works: The car and the diagnostic client validate the server certificate from the network server and only connect to it if it is authentic.

Communication between a diagnostic client and the car cannot be monitored (Confidentiality) or altered (Integrity) when using Remote Diagnostics
- Security mechanism: Encryption
- How it works: Application data is encrypted and signed when sent over the network
- Why it works: Cryptography suite is so strong that traffic cannot be decrypted (by ordinary means) nor altered without that being detected.

What it prevents: That someone listens to or alters traffic which is transported over the network.

RT-DS-15 ***How to distribute policy enforcement (allowed commands) as close to the target ECU (asset) as possible?***

There are several initiatives to force OEMs to share access to diagnostic services and information to independent workshops and authorities. The uncertainty is high when systems out of OEM control need to deal with credentials. Security mechanisms and key distribution need to consider these kind of scenarios.

RT-DS-16 ***How to authenticate and authorize access to diagnostic services in to 3:d parties that is resilient to misuse?***

4.5 Needs related to big data / data management.

Understanding data properties such as type, structure, legal ownership, criticality is important to give it a value and hence the need of protection from a security perspective. The properties may also be important factors to consider during risk assessment activities and having a good model of the data would increase the accuracy, consistency and efficiency of risk assessment activities. For example, time could be spent on analyzing protection of high value data instead of low value data.

Need: To describe data objects in order to make them understandable, this may be done by classification/categorization.

4.5.1 WP1 workshop discussions (big data/ machine learning)

Scientists often refer to the four dimensions of what makes up big data, the four V's, volume, veracity, variety and velocity, referring to the scale, the uncertainty, the different forms and the analysis of streaming data respectively. Together the four V's makes up what makes Big data different from just data.

Example applications for use of Big data in the automotive domain are predictive maintenance, product improvement, business improvement, autonomous driving, and intrusion detection system.

A challenge with big data, apart from the sheer volume that needs to be processed, is where and how to process it, in order to make sense of it and to make it useful. Mathematical statistics and machine learning and typical techniques used to make sense of data and create and improve prediction models. Processing may be done at different places, in the sensor itself, onboard in the vehicle or off-board at back office.

Collecting, processing and transferring vast amounts of data about vehicles inevitable raises privacy concerns and may big data applications require mapping of data to individual vehicles, i.e. to a VIN, Vehicle Identification Number. The obvious question is then, is it enough to anonymize the VIN number in order to make the data private, or is more required to achieve appropriate levels of privacy? Another example of sensitive data is GPS position.

Applications drive to use more and more sensor data as well as high resolution in time, data that may be combined to intelligence that is sensitive to the user.

Ownership

There are several potential owners of data in a vehicle, for instance the owner of the vehicle itself and the OEM, so how it the owner of a certain piece of data determined? Data that is considered to be personal data is regulated by regulations such as EU General Data Protection Regulation, GDPR, and does not belong to the OEM. Use of such data needs to be covered in Terms of use and contracts between OEM and the vehicle owner. This raises some concerns:

- Terms of use need to be communicated to the user in a comprehensible way
- User consent need to be given in advanced and cannot be generic

- ERC (Emission Related Components) require that data is extracted regardless of user consent due to regulatory requirements.
- What to do with the intelligence and who to share it with. Like predictions of part brake down after warranty period.
- What to do when authorities want information on data such as speed limit violation (speed, position)?
- Anonymization causes issues related to quality problem and recall since there is a need to know about individual components.

Questions to the presenter at the workshop

As a data analyst you use raw data sets, do you work with only the distribution?

- No , but it could be a possible way in some cases
- In many cases the raw data is needed to make the models

Big data often means ”Collect all the data” without knowing the use. If you do not collect all, what do you do when you discover that you needed another parameter? Are there any magical machine learning algorithms to find correlations that you cannot foresee?

- Cluster analysis and other techniques can be used in a explorative analysis
- You should ask the questions before you collect the data, because there are false correlations

How are Autonomous Drive systems taught to drive in different regions?

- There are base solutions and speculation to different regions

A discussion arose during the workshop regarding what data is stored by the workshop. An example was given of a workshop of a certain brand, where the workshop claimed they store the error codes from DROs, Diagnostic read-out, plus some additional data about the service event.

  – Is this privacy-preserving enough?

Solutions

One might broadcast a question to a group of vehicles and the individual vehicle will answer back.

  – Suggestion raised to design a distributed machine learning function instead of centralized where VIN is needed.

4.5.2 WP1 workshop discussions (data management - vehicle data classification)

RT-ID-21 ***What are the aspects to consider when determining the value of data?***

Label:

- Confidentiality (public, internal, confidential, ...)
- Integrity (safety, operational..)
- Privacy (sensitivity, ...)
- Regulations/Legislation...
- Availability...
- Life time / retention
- Quality of Data?

There is difficulties with different taxonomies during development. Need common language.

Value:

The value may be different depending of the context and also change over time.

Data in the connected vehicle system (i.e. on- and off-board) is an asset, i.e. it has a value.

The value depends on how it can be used, exploitability.

The value of data is used to determine risk when handling the data (which in turn is used to determine degree of protection required when data is handled)

To determine how valuable the data is for the Business or Data Subject, it needs to be categorized (preferably when Defined)

The value is related to the quality of the data. If we know if it is correct, how correct is it.

Need to know where data comes from, with different quality ”trustability”. The categories (labels) are the same over time, but the value changes

- Intended use (only for in-vehicle use etc.)?
- Raw data might not have much value, but the analysis models have. Refined data has often higher value than raw data.

Structure:

Not only are the labels needed, but also an information model. Potentially more than one model needs to be used on the same data at the same time. Could be different views.

Need to know how data fits together. As an example, engineers may know the few signals they work with quite well, but not in a broader context.

RT-ID-22 ***What is to be considered when rating each aspect?***

The step from individual data objects to combination of objects requires ground work. Sometimes the combination of things is private, but not the parts.

- E.g. one has a budget. If signal A and B are sensitive together, in your experiment you may use any one of the two, but not both

Integrity: Example, in order to handle a risk, one must use several signals with certain quality value. If signals disappear, maybe the remaining signals may not be used. Depends on the functional layer (e.g. Advance Driver Assist Systems (ADAS) sensor data).

It is important to find the right level of the information model. Too detailed and it becomes too much work, however, it may indicate areas where to dig deeper. It should be in parity with the work tasks that use the data, e.g. risk analysis.

A model can be used for more things than security, e.g. communication with suppliers.

Confidentiality Integrity Availability (CIA) rating is useful to work with at several levels, engineering as well as business (to discuss risks)

Failure Mode Effect Analysis (FMEA). It is being discussed how several FMEA fit together (interrelationships/connections). Maybe this is something to consider from a security perspective (e.g. Divide a problem into smaller and then connect them together again).

Is there tool support for automation of classification? Combination of manual and automatic?

What part of an organisation should own the content of the information model/data?

- Vehicle Engineering, IT? Enterprise Architecture?

Reaching the data classification goal

It may be difficult to reach 100%, but it might be possible to identify some middle steps and reach for them as a start

4.6 Needs related to Security Engineering Process (right solution with the right cost)

It is recognized that without an understanding of the system, the systematic analysis of the threats and countermeasures cannot be made. Appropriate handling of data security issues in the automotive industry can potentially lead to increased competitive advantage, revenue, and uptime.

A threat model describes data security aspects with respect to a particular system under investigation by associating a set a potential vulnerabilities, threats and attacks with a potential set of assets. Assets and threats are closely correlated as a threat cannot exist without a target asset.

The process of correlating threats to an asset involves creating adversary hypotheses. The output of threat identification process is a threat profile for a system, describing all the potential attacks, each of which needs to be mitigated or accepted. When improving data security, it is important to think of the development environment, to be able to develop the data security for the vehicle system in an effective way.

RT-ES-01 ***What does a risk driven security process look like?***

The data security environment describes the data security aspects of the environment in which the asset is intended to be used.

It shall include:

- Security assumptions
  - The intended use of the implementation
  - The physical, user and connection aspects of the environment in which an implementation will operate.
- Assets
  - The assets with which the assets under analysis will interact with
  - The nature of the asset's interaction with other assets.
- Threats and threat agents
  - All threat agents which specific protection is required within either the implementation of standard or its expected environment
  - The threat agents that will be used to enact the identified threats
- Organizational security policies
  - Any security policies or rules with which an implementation of a standard must comply.

There are no strict rules on how to determine what is in the system and what is in its environment but, as a guide, in communication systems it is likely that the boundary will pass through interfaces.

A ToE for data security is defined such that all potential attack interfaces are exposed; i.e. each interface has one end-point inside the ToE and one end-point outside of the ToE boundaries.

RT-ES-02: ***The ToE description for data security shall include a high level description of the main assets of the ToE and its environment.***

Limiting the attack surface makes the problem more traceable. A ToE can only be attacked through its exposed interfaces and the presence of a threat agent is necessary to launch an attack. This means that the assets (most often the functional entities) associated with the exposed interfaces are potential threat agents, and that the ToE environment should include all exposed interfaces and all assets associated with the end-points of these interfaces that are outside the ToE.

RT-ES-03 ***What is a good balance between centralizing and distributing security and privacy efforts?***

Security and privacy is a cross functional and cross domain effort where knowledge from several domains is needed. At a minimum knowledge from domains like:

- information and computer security,
- embedded development
- legal

To find people that master all domains is rare so some form of collaboration is needed. There is need to finding a good balance between a centralized approach where security experts do all security and privacy work or distributing the responsibility to all involved. There are several facts that need to be considered, finding security experts is hard, it may take longer time to understand the system than to make the embedded engineers understand the security and privacy concerns.

What is a minimum set of security to teach an embedded developer to produce reproducible results?

What is a minimum set of embedded & system knowledge to teach a security expert to produce reproducible results?

4.6.1 Overall System Design

Scenarios in this category are concerned with an overall system (architecture) in terms of its composition, necessary behaviours and non-functional qualities, as well as indispensable implementation concerns without taking details of technologies into consideration.

RT-ES-10 ***Security aspects to be considered in the overall product design***

SAE J3061[3] is published, but leaves many degrees of freedom to implementation so claiming that a best practice has been applied is hard at the moment. ISO 26262[4] is an international standard for automotive functional safety. Similar framework for standardization needs to be pushed for within the data security area. Joint work between SAE and ISO has already started.

RT-ES-11 ***How do we decide on what is an appropriate security solution/level that meets design and project requirements?***

It is of outermost importance to align hazard analysis with data security threat analysis

RT-ES-12 ***The safety hazard analysis shall be aligned with data security threat analysis.***

Threat analysis in general describes security aspects with respect to a particular system under investigation or TOE by associating a set of potential vulnerabilities, threats and attacks with a potential set of assets. Assets and threats are closely correlated as a threat cannot exist without a target asset. When including data security threats as potential sources of hazards, the risk assessment needs to be reevaluated and the assessment value might increase in integrity level. The work consists of identification of threats and their analysis. The result of the activity is a list of threats and their components. Threat analysis and vulnerability analysis can only be performed to the level of detail up to which the system under development has already been established (needs to be verified and refined during technical solution phase).

4.6.2 UC-ES-01 OEM and HW/SW supplier

ID	*UC-ES-01*
Title	OEM_and_HW/SW_supplier
Version	1.0
Status	Released
Motivation	Composing new sub-systems from simpler sub-systems and describing their composed behavior, which is a major architectural task. - **Issues of concern from suppliers' perspective:** economies of scale, i.e. minimize number of product variants to reduce costs - **Issues of concern from OEM perspective:** to have supplier implement innovative functionality exclusively for this OEM (product differentiation against competition) - **Issues of concern for both stakeholders**: security, reluctant to disclose intellectual property. clarify interface and freeze design of functionality
Stakeholders	- HW-supplier; SW-supplier; - manufacturer (OEM)
Process context	Deployment of functionality (SW) to different HW components.
Product context	Requirements definition and joined architectural design of sub-system to be delivered
Organization context	Not performed within one single organizational unit but between two development partners (OEM and supplier on a contractual basis) and refined on sub-systems parts.

This scenario represents a major architectural task that aims at composing new sub-systems from simpler / basic sub-systems and to describe their composed behaviour, with a particular focus on the (sub-system) interfaces to be shared between different vendors.

RT-ES-13 ***Security aspects to be considered between OEM and HW/SW supplier***

Security aspects must be considered both from OEM and HW/SW supplier. This example is a part of early architecture design that provides requirements definition and joined architectural design of sub-systems when the deployment of functionality (SW) to different HW components are considered. Both the OEM and various suppliers perform this scenario. This scenario reflects current practices and indicates some future needs. One time-proven method for "interface matching" is, to use a common standardized bus (e.g. CAN) and both partners agree on the protocol, how to use it.

Basic activities and process

The activities and process of this scenario can take different forms, depending on the interfaces to be shared. The process is characterized by communications between the partners. The process is ended when the partners have reached agreements at least on an interface, electrical interfaces, procedures for fault reporting and diagnosis and non-functional properties (like security).

Information refinement/Transformation flow

This scenario produces a precise interface description (i.e., outputs) with inputs from both OEM and suppliers. That is, while the OEM provides the specification of functionality, the suppliers provide specifications of available basic sub-systems. In this information flow, one key issue is *variants*

management.

- Innovative functionality usually requires the current system description to be extended and sometimes even modified.
- Extension and modification usually force the supplier to adapt his subsystem to every OEM's needs differently, introducing a lot of (mostly software) variants.

Technical and managerial techniques

To ensure the success of performing this scenario, a variety of techniques listed below are considered necessary.

Techniques for information communication :	*(needed) tool for transformation of the interface description information into the different sub-system development tools in use at both partners;*
Techniques for quality assessment:	*(needed) conformance checker.*
Techniques for creating technical solutions:	*- Modeling of systems structure especially the interface between the supplier- and the OEM-side of the system* *- modeling of systems behaviors (both in normal mode and various degraded modes);*

4.6.3 UC-ES-02 Creating requirement specification document

ID	*UC-ES-02*
Title	Creating requirement specification document
Version	1.0
Status	Released
Motivation	Misleading / insufficient requirement specifications and high number of iterations for settling on requirements cause higher costs and time to market.
Stakeholders	- OEM; - Supplier
Process context	Threat Analysis and Risk Assessment Creation of requirement specification document
Product context	Requirement
Organization context	Not performed within one single organizational unit but between two development partners (OEM and supplier on a contractual basis) and refined on subsystems parts.

This scenario aims at improving the quality of requirement specifications by detecting and removing potential problems – such as mistakes and insufficient specifications – and hence forms a basis for efficient product development.

RT-ES-14 ***Security aspects to be considered in when creating requirement specification document***

It is performed at the requirement specification phase of system development, involving both OEM and suppliers.

Engineering process needs to be risk driven in order to prioritize and selects security solutions that meets the requirements. Risk management must be an integral part of the engineering / development process!

- Risk management - gives the assets at risk
- Risk assessment - gives the level of risk for the assets

RT-ES-15 ***Who has the authority to take security risk decision and how does this affect the engineering process?***

Risk decisions are often taken at the wrong level in the organization.

It is a misguided perception that information security mainly is a technical matter

Threat analysis – Description of the functional use cases is the input to the threat analysis process. Threat analysis produces two outputs: (a) a mapping between threats and assets, for each asset in the context of the use case, and (b) a mapping between threats and security attributes to establish which security attributes are affected due to a particular threat in the context of an asset.

Risk assessment – Once the threats for the relevant assets are identified, the next step is to rank the threats. This is what is done during risk assessment.

Security requirements – Based on input from the risk assessment, formulation of security requirements.

RT-ES-16 ***How shall we manage prioritization of security requirements together with other requirements?***

Requirements from many different stakeholders such as legal, quality assurance, business case, safety etc.

need to be satisfied.

- We need to find and manage all stakeholder requirements
- We need a complete view and traceability

This scenario reflects partly the real world today (which is based on Excel, Word documents). The vision is having more formal exchange mechanisms between OEM and suppliers. It would then cover most aspects of requirements development and specification between OEM and supplier except the validation, verification, and prototyping.

Basic activities and process

This scenario consists of four activities as listed below.

Activity_1: *Analyzing Requirements.*

Activity_2: *Discussing existing solutions and reusing them.*

Activity_3: *Calculating cost and planning.*

Activity_4: *Documenting.*

The process starts with the OEM and suppliers capturing and assessing some original requirement statements (i.e., activity_1). Thereafter, the requirement statements are analysed and traded off by taking existing solutions into consideration (i.e., activity_2), such as regarding to the feasibility, testability, security, and reusability, and then evaluated with respect to the cost (i.e., activity_3). In parallel with these two activities, the requirements specification document is produced by the documenting activity (i.e., activity_4). These three activities iterates until a satisfactory solution concept is agreed between these two stakeholders.

Information refinement/Transformation flow

With the given customer requirements, this scenario produces the requirements specification document by consolidating a variety of information listed below.

Inputs:	*- Customer requirements.* *- Formal and non-formal system specification – model, system descriptions.*
Internal_Info1:	*From OEM* *- Revised customer requirements.*
Internal_Info2:	*From OEM* *- Specifications of high level functionalities (features) of product and their deployment on ECUs.* *- (Optional) Partial specification of implementation constraints (e.g. a particular Micro-Controller or technology to be used).*
Internal_Info3:	*From Supplier* *- specifications of the implementation solutions for the functionalities and the needed technological resources (e.g. choice of memory, Micro-Controller etc.).*
Outputs:	*Requirements specification document*

To generate the requirements specification document, this scenario first receives the statements of customer needs and the specifications of overall systems (i.e., Inputs). The original requirement statements are then transformed into revised versions (i.e., Internal_Info1) on the basis of the overall system specifications. With the system specifications and the revised customer requirements including test criteria, specifications of high level functionalities (features) of product and their deployment on ECUs (i.e., Internal_Info2) are provided by the OEM, possibly together with a partial specification of implementation constraints and preferences (e.g. a particular Micro-Controller or technology to be used). Based on these OEM decisions, specifications of the implementation solutions as well as necessary technological resources (i.e., Internal_Info3. such as choice of memory, sensors and actuators, Micro-Controller etc.) are then

provided by the suppliers. The requirement specifications in their final form are finally released as the requirements specification document (i.e., Outputs).

Technical and managerial techniques

One central issue in this requirement specification scenario is concerned with the choice of appropriate hardware including sensors, actuators, and architectural design of software. Often existing concepts and technologies are reused for producing a new system. However, sometimes it is necessary to prove new concepts concerning costs, development time, etc. via prototyping. To ensure the success, a variety of techniques listed below are considered necessary.

Techniques for information communication :	*requirements analysis*
Techniques for quality assessment:	*- requirements analysis and break down* *- Prototyping* *- Simulation*
Techniques for creating technical solutions:	*- Architectural design of software.* *- Choice of appropriate hardware.*
Techniques for information management:	*Documentation*

4.6.4 UC-ES-03 Extending product specifications

ID	*UC-ES-03*
Title	Extending product specifications
Version	1.0
Status	Released
Motivation	Only few systems are developed from scratch; often there is an existing solution, which is extended for the next generation or modified to fit other vehicle types.
Stakeholders	- (software) developer from supplier; - integration tester, system architect/ specifies from OEM
Process context	Development phase
Product context	- Completed (and implemented) product specification (to be extended); - list of (additional) features to be realized by the product
Organization context	(advanced) development department of OEM

This scenario aims at extending or modifying existing products for new products or product variants in which some additional needs will be satisfied. It provides an extended system specification by extending the specifications of existing products with a list of additional features. This scenario is performed at product development phase within the organization of OEM while involving (software) developers of suppliers.

RT-ES-17 ***Security aspects to be considered when extending product specification***

Current practices are based on a *copy-modify* approach characterized by the following two steps: 1-copying the requirements description of the existing system, and then 2-modifying its requirements to fit the extended needs.

Basic activities and process

This scenario has a process composed of six major activities as listed below.

Activity_1: *modeling/describing existing system*

Activity_2: *modeling/describing/developing additional functionality/services*

Activity_3: *Integrating additional functionality*

Activity_4: *Performing test / regression test of the extended system*

Activity_5: *Ensuring non-functional properties (weight, EMI, safety, security …)*

Activity_6: *Extending TARA, FMEA, production, user and service documentation database*

The process starts with modelling the existing system (i.e., activity_1). Then, the system specification is extended (i.e., activity_2), such as by adding additional functionality on the same execution environment, by using the same functionality on an extended Operational / Technical / Hardware Architecture. These new features are then integrated (i.e., activity_3), while taking the non-functional properties into consideration (i.e., activity_4). Thereafter, the new system is tested with respect to the consequences of changes (i.e., activity_5). During the process, the existing documents in database are also updated (i.e., activity_6).

Product information refinement/Transformation flow

This scenario generates an extended/modified product specification (i.e., Outputs) with inputs in terms of existing product specification and a list of additional requirements.

Technical and managerial techniques

To perform this product modification/extension scenario, techniques listed below are considered necessary.

Techniques for creating technical solutions:	*Modeling techniques that support* *- extending an existing model (discerning between previous versions and current version)* *- handling of model variants (discerning between alternative execution environments)*

4.6.5 UC-ES-04 System quote

ID	*UC-ES-04*
Title	System quote
Version	1.0
Status	Released
Motivation	Define (sub-)system product architecture at quotation process
Stakeholders	System lead with project team (SW , HW , System)
Process context	Quotation process
Product context	(Sub-) System product definition with selection of sensor and actuator component, hardware features selection, system function selection
Organization context	(Sub-) System supplier in system division

This scenario is performed by a project team of suppliers at the quotation process, aiming at defining (sub) system product architecture in terms of basic system functions and implementation choices. The quotation scenario plays a key role in product development. It is highly expected that an information exchange between OEM and suppliers would be supported. Moreover, the product definition, at least with at least black-box descriptions of new functions, is also expected. This will form a basis for Start-of-Development as well as for recording hypothesis recorded during quotation.

RT-ES-18 ***Security aspects to be considered when producing system quote***

Basic activities and process

This scenario consists of three major activities as listed below.

Activity_1: *Analysing and managing original requirements (set of requirement reviewed)*

Activity_2: *Building technical solutions, including*
- Selecting sensors/actuators.
- Selecting/configuring function to reuse
- Identifying function to be developed (function landscape and external data flow definition)
- Selecting HW platform configuration (Reuse/adaptation of HW platform + block diagram selection)

Activity_3: *Verifying and validating technical solutions, including*
- Controlling consistency of data flow (check on high level of abstraction : function)
- Checking conformity matrix with requirements
- Validating safety and security requirements (basics concepts)
- Estimating performance (RAM, ROM, CPU)

The process starts with the project team reviewing the given requirements (i.e., *activity_1*). After completing the requirement analysis, the project team produces the architecture definition (i.e., activity_2) by taking the existing solutions into consideration, and then evaluates the solutions (i.e., activity_3) iteratively until all the requirements are satisfied by the solutions.

Product information refinement/Transformation flow

With the given requirements and constraints, this scenario produces the product definitions by consolidating

a variety of information as listed below.

Inputs:	*- System requirement specification from OEM* *- Supplier generic specification of SW*
Internal_Info1:	*Break down of requirements in regards to generic product architecture*
Internal_Info2:	*Definition of system functions, including* *- reused system function in regards to requirement* *- new system functions*
Internal_Info3:	*Definition of hardware platform, including* *- reused hardware platform* *- hardware platform adaptation area*
Outputs:	*System product definition and architecture definition, including* *- System component selection actuators / sensors;* *- HW product definition (block diagram view) and platform selection, - system function selection,* *- software specification reference list.*

To generate the system product definition and architecture definition, this scenario first receives the system requirement specification from OEM and a generic platform specification of the supplier itself (i.e., Inputs). The external requirements are then transformed into revised versions (i.e., Internal_Info1) in regards to the generic platform specification. Thereafter, the necessary system functions (i.e., Internal_Info2) and platform solutions (i.e., Internal_Info3) are defined on the basis of existing solutions. The design then provides a setup of "rough" software specification reference list (i.e., Internal_Info4).

Technical and managerial techniques

To perform this (sub-) system architecture definition scenario, the techniques listed below are considered necessary.

Techniques for information communication :	*- Workshop;* *- Brainstorming;* *- Analysis report*
Techniques for quality assessment:	*- Reviewing.* *- Analysis.*
Techniques for creating technical solutions:	*Reuse of existing solutions;* *- Estimation for new solutions;.*
Techniques for information management:	*- Requirement and solution management* *- Documentation* *-configuration management support*

4.6.6 UC-ES-05 System design

ID	*UC-ES-05*
Title	System design
Version	1.0
Status	Released
Motivation	Initial composition of a (sub-) system
Stakeholders	System integration responsible
Process context	Start of development of a project (basis set up) or major change in architecture Refined Threat Analysis and Risk Assessment
Product context	- (sub-) system product definition with refinement of selection of sensor and actuator component, hardware features selection, system function selection done during quote phase or at major of system architecture change; - Architecture setup with detailed analysis, detailing of function portioning and components;
Organization context	(Sub-)System supplier in system division

This scenario is carried out by the system integration responsible of (sub) system suppliers. It aims at defining the initial composition of the (sub-) system for description and verification and validation purposes.

RT-ES-19 ***Security aspects to be considered during system design***

Performed at the start of a development project or after major change, it refines the architecture setup produced in the quote phase or after major changes by providing detailed analysis and refining the function portioning and components definitions.

RT-ES-20 ***How can we create an escalation path for risk decisions that is easy and quick to use?***

The work to reduce data security risks involves defining and evaluation of countermeasures.

RT-ES-21 ***Data security countermeasures shall be added, and properly evaluated to the system to reduce the weighted risk to the system.***

The goal of data security design is to ensure a low likelihood of an unwanted incident arising.

In the security engineering process, the next step to be followed by threat analysis and risk assessment is to define countermeasures to mitigate risks. Risk analysis provides the rationale for identifying requirements for specific countermeasures Data security risk countermeasures are assets that are added to the system, and properly evaluated to reduce the weighted risk to the system.

Identification of data security goals and objectives

RT-ES-22 ***The data security goals and objectives shall be identified.***

The work consists of identifying data security goals and objectives i.e. top-level automotive security requirements for the asset that identify the broad aims of the data security protection to be given.

The data security goal and objectives are expressed in broad terms rather than in detail and are segregated into two distinct groups:

- Data security goals and objectives for the asset
  - It should be clear which aspects of the identified threats and policies are addressed by each objective
  - If the base data security standard specifies a protocol, it is likely that the asset data security objectives will be specified in the standard documentation.
- Data security goals and objectives for the environment
  - It should be clear which aspects of the identified threats and policies are expected to be addressed by the environment because they are not covered (either wholly or in part) by asset data security objectives
  - Communication standards rarely specify requirements for the environment so it is unlikely that the data security objectives for the environment will be stated explicitly in the base data security standard or any associated document.

Deriving functional and technical data security requirements

RT-ES-23 ***Functional and technical data security requirements shall be derived from data security goal and objectives.***

The work consists of deriving functional and technical security requirements from the security goals and objectives. Input to this activity is reports or knowledge of previous misuses, reported incidents on similar products in the past from the field or on competitor products.

A feature is not complete if (end-to-end) security and privacy requirements are not fulfilled and cannot be delivered without risk acceptance. We need to understand that features might introduce risk

- by the feature itself
- to other features in the system
- to the underlying system holding the feature
- to other systems interacting with the feature

To consider the feature complete, the security requirements from the feature risks must not be managed / prioritized separately from the feature requirements.

A refined TARA is recommended.

Refined threat analysis – Review of the functional use cases is the input to the refined threat analysis process

Risk assessment – Once the threats have been re-evaluated, the relevant assets are also re-evaluated, the ranks of the threats are reviewed and possible re-considered. This is what is done during risk assessment.

Security requirements – Based on input from the refined risk assessment, possible re-formulation of security requirements or even identification of new security requirements.

This scenario is performed by a project team of suppliers at the quotation process, aiming at defining (sub) system product architecture in terms of basic system functions and implementation choices.

This scenario addresses the support for handling initial requirement and part of platform, function, hardware reused by the product, record system configuration quoted.

Basic activities and process

This scenario consists of three major activities as listed below.

Activity_1: *Analysing and managing original requirements (set of requirement reviewed)*

Activity_2: *Building technical solutions, including*
- Selecting sensors/actuators.

- Selecting/configuring function to reuse
- Identifying function to be developed (function landscape and external data flow definition)
- Selecting HW platform configuration (Reuse/adaptation of HW platform + block diagram selection)

Activity_3: *Verifying and validating technical solutions, including*
- Controlling consistency of data flow (check on high level of abstraction : function)
- Checking conformity matrix with requirements
- Validating safety and security requirements (basics concepts)
- Estimating performance (RAM, ROM, CPU)

The process starts with the project team reviewing the given requirements (i.e., *activity_1*). After completing the requirement analysis, the project team produces the architecture definition (i.e., activity_2) by taking the existing solutions into consideration, and then evaluates the solutions (i.e., activity_3) iteratively until all the requirements are satisfied by the solutions.

Product information refinement/Transformation flow

With the given requirements and constraints, this scenario produces the product definitions by consolidating a variety of information as listed below.

Inputs: *- System requirement specification from OEM*
- Supplier generic specification of SW

Internal_Info1: *Break down of requirements in regards to generic product architecture*

Internal_Info2: *Definition of system functions, including*
- reused system function in regards to requirement
- new system functions

Internal_Info3: *Definition of hardware platform, including*
- reused hardware platform
- hardware platform adaptation area

Outputs: *System product definition and architecture definition, including*
- System component selection actuators / sensors;
- HW product definition (block diagram view) and platform selection, - system function selection,
- software specification reference list.

To generate the system product definition and architecture definition, this scenario first receives the system requirement specification from OEM and a generic platform specification of the supplier itself (i.e., Inputs). The external requirements are then transformed into revised versions (i.e., Internal_Info1) in regards to the generic platform specification. Thereafter, the necessary system functions (i.e., Internal_Info2) and platform solutions (i.e., Internal_Info3) are defined on the basis of existing solutions. The design then provides a setup of "rough" software specification reference list (i.e., Internal_Info4).

Technical and managerial techniques

To perform this (sub-) system architecture definition scenario, the techniques listed below are considered necessary.

Techniques for information communication : *- Workshop;*
- Brainstorming;
- Analysis report

Techniques for quality assessment:	*- Reviewing.* *- Analysis.*
Techniques for creating technical solutions:	*Reuse of existing solutions;* *- Estimation for new solutions;.*
Techniques for information management:	*- Requirement and solution management* *- Documentation* *-configuration management support*

4.6.7 Functional design

Scenarios in this category focus on the design of system functionality and qualities at logical levels that are independent of technology details (e.g., design of software programs).

4.6.8 UC-ES-06 Function design

ID	*UC-ES-06*
Title	Function design
Version	1.0
Status	Released
Motivation	Design of a function for a (sub-)system
Stakeholders	- OEM; - System function designer;
Process context	Prototyping loop phase (including generic development)
Product context	Development of function for OEM project considering generic purpose of reuse
Organization context	(Sub-) system supplier in system division

This scenario is carried out by (sub) system suppliers as a part of the product prototyping-loop phase on the request of OEMs. It aims at designing the functions of (sub-) systems for description and verification and validation purposes before the implementation.

RT-ES-30 ***Security aspects to be considered during function design***

This is a development scenario in which the management of variability is explicitly considered.

Basic activities and process

This scenario consists of three major activities as listed below.

Activity_1: *Requirement analysis and management (set of requirement reviewed)*

Activity_2: *Building of technical solution, including the following sub activities*
- Architecture definition with components definitions (functional oriented) via package definition
- Modeling of components interfaces
- Diversity and configuration definition
- Definition of system integration constraints
- Definition/Selection of Hardware abstraction layer services
- Selection of HW platform configuration (Sensor, Actuator, Driver)
- Development of algorithms for control law (MatLab/Simulink)
- Definition of detailed test case scenario (for all configuration)
- Definition of safety and security scenarios
- Generation of software specification (or Model for code generation)

Activity_3: *Verification of technical solution, including the following sub activities*
- Data Flow Interface check
- Consistency of Data and control Flow (historical data) via scheduling policy
- Integration verification (including for diversity of function)

- Functional requirement validation (including for diversity of function)
- Safety and Security Requirement validation (critical reaction test scenario, Fault Tree Analysis scenario, Attack Tree Analysis scenario ...)
- Timing performance valuation (Dead Line respect)
- Resource Estimation (RAM, ROM, CPU)
- Timing performance evaluation (Dead Line respect)
- Early validation via rapid-prototype verification

The process starts with a reviewing of the given requirements (i.e., activity_1). After completing the requirement analysis, the functional solutions are defined (i.e., activity_2) by taking the overall architecture definition, implementation and platform constraints, as well as variability, testing, security, and safety into consideration. The generated solutions are then evaluated (i.e., activity_3) with respect to conformances, functionality, timing, and resource utilization. The last two activities are performed iteratively until all the requirements are satisfied.

Product information refinement/Transformation flow

This scenario is performed with information listed below.

Inputs:	*- System Requirement Specification (OEMs and all stakeholders)* *- Generic Platform Specification*
Internal_Info1:	*Breakdown of Requirements in regards to standard architecture*
Internal_Info2:	*Reuse of system function in regards to requirement*
Internal_Info3:	*Reuse of hardware platform.*
Internal_Info4:	*Design of new system functions and software specification*
Outputs:	*Function system definition and architecture definition (System component selection actuators / sensors; HW product definition (block diagram view) and platform selection, system function selection).*

Technical and managerial techniques

This function design scenario requires a variety of techniques as listed below.

Techniques for information communication:	*- Workshop;* *- Software specification or model* *- Analysis report*
Techniques for quality assessment:	*- Reviewing.* *- Simulation and rapid prototyping solution* *- testing*
Techniques for creating technical solutions:	*Modeling technique (including architecture features : Middleware and OS features);*
Techniques for information management:	*- Configuration management support* *- Documentation* *- Specifications and models*

4.6.9 UC-ES-07 Behaviour definition

ID	*UC-ES-07*
Title	Behavior definition
Version	1.0
Status	Released
Motivation	Behavioral definitions are important to capture the dynamics of the systems.
Stakeholders	Software engineer, control engineer,
Process context	System design
Product context	Product and system/sub system
Organization context	Engineering department at OEM/Supplier

This scenario aims at defining and modelling the behaviours of a (sub) system on various levels of detail. The software and control engineers perform it at the product design phase.

RT-ES-31 ***Security aspects to be considered during behavior definition***

Basic activities and process

This scenario is performed incrementally with increasing levels of detail ranging from use cases to behaviour specification of the application software.

Product information refinement/Transformation flow

This scenario provides definitions of the behaviours of functions (i.e., Outputs) that compose the EE system. The required inputs include specifications of requirements, feature descriptions, interface definitions, and structural definitions.

Technical and managerial techniques

The functions are defined using the language's support for structure, behaviour, requirements, etc. in which the techniques listed below are considered important.

Techniques for creating technical solutions:	*- Simulation* *- graphical editing* *- code generation (e.g., from vehicle level to analysis level, from analysis to design, from design to implementation)* *- traces*

4.6.10 UC-ES-08 Definition of functions

ID	*UC-ES-08*
Title	Definition of functions
Version	1.0
Status	Released
Motivation	Functions have to be defined as they are the basis of the features
Stakeholders	Control engineer, Domain Expert
Process context	Design phase
Product context	Definition of functional content of the product, both systems and sub systems and the entire vehicle
Organization context	Engineering department at OEMs and suppliers

This scenario aims at defining functions that form the basis for features. The control engineers and domain experts at product design phase perform it.

RT-ES-32 ***Security aspects to be considered during definition of functions***

It is not enough to isolate security functions into their own components. Each security-critical component must, to the greatest extent possible, be designed or refactored to remove any functionality that is not part of its security-enforcing function.

Dividing a system into components requires that they have well-defined interfaces.

Basic activities and process

This scenario provides the definitions of functions based on the features. It is performed iteratively with increasing levels of detail with the following five basic activities.

Activity_1: *Identifying functional content of the feature.*

Activity_2: *Decomposing functions into sub functions.*

Activity_3: *Identifying internal and external interfaces.*

Activity_4: *Defining behaviour of functions.*

Activity_5: *Refining and defining requirements on function.*

Product information refinement/Transformation flow

This scenario provides definitions of the functions (i.e., Outputs) that compose the vehicle's EE. The inputs are specifications of feature content and overall requirements on vehicle, as well as other constraints and legacy components.

4.6.11 Implementation

Scenarios in this category are concerned with the implementation of system functions on a platform, where the given functions are refined by taking the implementation means (e.g., software programs) and details of platform specific issues into consideration.

4.6.12 UC-ES-09 Function implementation

ID	*UC-ES-09*
Title	Function implementation
Version	1.0
Status	Released
Motivation	Implementation of a function for a (sub-) system (possible code generation and verification and validation objective; considering generic purpose of reuse)
Stakeholders	- OEM; - Software developers, System Engineers, HW Engineers of suppliers.
Process context	Prototyping loop phase (including generic development)
Product context	Development of functions (for OEM project)
Organization context	(Sub-) System supplier

This scenario aims at defining implementation solutions for the predefined global and/or local functions with respect to an execution platform. Being a part of the prototyping-loop in the development of product functions, this scenario is performed by software developers of suppliers on the request of OEMs.

RT-ES-40 ***Security aspects to be considered during function implementation***

This scenario is important since platform related issues are considered central.

Numerous static code analysers, and some compilers, can automate large portions of a typical secure coding standard.

The CERT C Coding standard, Second Edition [5], and MISRA C 2012 Amendment 1 for security [6] are primarily intended for developers of C language programs but may also be used by software acquirers to define the requirements for bespoke software.

Security coding standards are of particular interest to developers who are interested in building high-quality systems that are reliable, robust, and resist to attack.

If a code-generating tool is to be used, it is necessary to select an appropriate tool and undertake validation.

Basic activities and process

This scenario consists of five major activities as listed below.

Activity_1: *Analysing and managing original requirements (set of requirement reviewed);*

Activity_2: *Building technical solutions, including*
- determining code configuration for diversity purpose (version configuration and data set configuration)
- determining code generation and configuration for platform environment

- defining allocation of logical cluster/tasks for integration
- performing configuration management
- linking and Locating
- writing test case for SW parts

Activity_3: *Verifying and validating technical solutions, including*
- code reviewing
- verifying coding rules
- verifying platform integration
- verifying integration (link and locate) including diversity of component configuration
- measuring resource utilization (RAM, ROM, CPU timing)
- performing unitary test (reuse of pattern from modelling) or proof via formal verification (prospective)
- fuzzy testing of components
- performing scheduling and timing requirement validation (and WCET calculation: prospective)
- Performing early validation / integration on PC virtual platform (prospective)

Activity_4: *Releasing and delivering technical solutions*

Activity_5: *Configuration and change management*

The process starts with the software developers reviewing the given requirements (i.e., activity_1) together with systems and hardware engineers, such as with respect to feasibility, appropriateness, and relevance. After completing the requirement analysis, the software developers carry out the design in terms of code configuration, allocation, and scheduling (i.e., activity_2) and then evaluate the solutions with respect to coding quality, platform integration, and resource utilization (i.e., activity_3). The two activities are performed iteratively until all the requirements are satisfied by the solutions. In parallel with the technical activities, a management activity (i.e., activity_5) is performed to provide necessary configuration and change management support. The final solutions are made available and transferred to the OEM (i.e., activity_4).

Product information refinement/Transformation flow

This scenario produces the implementation solutions with information listed below.

Inputs:	*- Specifications of software including the non-functional requirements;* *- Generic platform specification;*
Internal_Info1:	*- Revised specifications of software including the non-functional* requirements*;* *- Revised generic platform specification;*
Internal_Info2:	*- Instantiation of software components for the given functions;* *- Decomposed/refined non-functional software requirements;* *- Choice/implementation of platform;*
Internal_Info3:	*- Implementation of data/code versus platform*
Internal_Info4:	*- Creation/validation of OS Logical Cluster* *- Creation of implementation details related to Software component (memory allocation ...)*
Internal_Info5:	*- Validation results on target of SW component*
Outputs:	*Software component, software implementation details, link with physical electronic product* *- Software component definition;* *- Software integration definition*

To determine the implementation solutions, this scenario first receives a set of specifications defining the desired software functionality and non-functional properties, together with a specification of the preferred target platform (i.e., Inputs). These given requirements and constraints are then transformed into revised versions for the upcoming development work (i.e., Internal_Info1). With the specifications, the software components for the given functions are then instantiated with logical variables and features replaced by their implementation specific correspondences (i.e., Internal_Info2). Thereafter, the design is refined in terms of code level solutions (i.e., Internal_Info3) as well as solutions for the platform integration (i.e., Internal_Info4). The implementation solutions is then evaluated and documented (i.e., Internal_Info5).

Issues that are of concern during the evaluation include for example the integration, functionality, platform compliance etc. The solutions in its release form are given as a set of software component definitions and integration definitions (i.e., Outputs).

Technical and managerial techniques

To perform this function implementation scenario, the techniques listed below are considered necessary.

Techniques for information communication:	*Workshop;* *Coding module;* *Document report*
Techniques for quality assessment:	*Reviewing.* *test : Simulation and test bench ;* *Analysis techniques*
Techniques for creating technical solutions:	*Coding, Compiler, Linker,…;*
Techniques for information management:	*Configuration management support.*

4.6.13 UC-ES-10 Function to node allocation

ID	*UC-ES-10*
Title	Function to node allocation
Version	1.0
Status	Released
Motivation	The function allocation is central for design of a distributed system
Stakeholders	System engineer,
Process context	System analysis, design and implementation
Product context	Product and system/subsystem at the implementation or functional level
Organization context	Engineering department at OEM/Supplier

This scenario aims at defining the allocation of functions on a distributed platform at the system analysis, design, or implementation phase.

RT-ES-41 ***Security aspects to be considered during function to node allocation***

The product or subsystems under consideration is normally the implementation design, where the chosen allocation is implemented as tasks and frames on specific busses and ECUs. Optionally, the allocation can also be considered in terms of the functional design on analysis level.

Basic activities and process

The process starts with the designers identifying the function set, the node set, and the allocation constraints. Thereafter, the designers decide an optimal allocated solution that meets the constraints and qualities of concern. The evaluation is supported by analysis and testing activities and other qualities.

The systems engineer wants to know the value and requirements of data being used in order to take the decision to expose said data if e.g. it has to be transported over physical wire, or if node allocation can be done so that the information is ECU internal (e.g. consolidating ECUs).

Product information refinement/Transformation flow

Inputs to this scenario include the set of functions to be allocated, the nodes and networks of target platform, and the allocation constraints that represents the concerns of legacy, supplier choice, modularization, efficiency, etc. Outputs from this scenario define the locations of functions on the platform in terms of the corresponding tasks and frames on ECUs and specific busses.

4.6.14 UC-ES-11 Task scheduling

ID	*UC-ES-11*
Title	Task scheduling
Version	1.0
Status	Released
Motivation	Task scheduling is necessary to meet timing constraints of the system
Stakeholders	System engineer, platform SW engineer
Process context	Implementation
Product context	Product and system/subsystem
Organization context	Engineering department at OEM/Supplier

This scenario aims at defining the execution schedule of software tasks in order to meet timing constraints of functions. It is performed by the system engineer and platform SW engineer at the implementation phase of a product or its sub-systems.

This scenario reflects current practice in the sense that task scheduling information, including both requirements (deadlines, rates, etc.) and parameters (WCET, priorities, timing, etc.), has to be supported.

RT-ES-42 ***Security aspects to be considered during task scheduling***

The platform SW engineer wants to ensure temporal isolation, so that a task being attacked does not starve other tasks from CPU time, i.e. denial of service.

Basic activities and process

The process starts with the designers identifying the timing constraints, partitioning and allocation decisions of functions, as well as other constraints of the execution and communication platforms. Thereafter, the execution schema is synthesized according to task-models defined on the basis of a scheduling theory/algorithm. The process ends when satisfactory solutions have been identified after formal verification or testing.

Product information refinement/Transformation flow

Inputs to this scenario include the timing definition of hosted functions (deadlines and rates) and the (required) execution time of task. Essential information in the execution schema (i.e., Outputs) includes the definitions of periods or priorities for each task, according to the timing constraints of its software and data.

4.6.15 UC-ES-12 Communication Scheduling

ID	*UC-ES-12*
Title	Communication scheduling
Version	1.0
Status	Released
Motivation	Message scheduling is necessary to meet timing constraints of the system
Stakeholders	System engineer, platform SW engineer
Process context	Implementation
Product context	Product and system/subsystem
Organization context	Engineering department at OEM/Supplier

This scenario aims at scheduling communication messages in order to meet timing constraints during the execution. It is performed by the system engineer and platform SW engineer at the implementation phase of a product or its subsystems.

This scenario reflects a current practice in the sense that message scheduling information, including both requirements (deadlines, rates, etc.) and parameters (transmission time, message size/bit rate, priorities, timing, etc.), has to be supported.

RT-ES-43 ***Security aspects to be considered during communication scheduling***

Basic activities and process

This scenario is performed by first identifying the configuration constraints, messages of concern, and the desired timing of signals/data contained in the messages. Then, the communication schema is modelled and defined on the basis of a chosen communication infrastructure.

The System engineer needs to consider the resynchronization of communication in case a secure channel is lost, e.g. how frequently do you need to send data to ensure that data is received and understood by the receiver with a certain probability even if there is a need for resynchronisation of freshness parameters at a critical time.

Product information refinement/Transformation flow

To produce the communication schema, important inputs includes the signals/data set to be communicated and their sizes, rates and deadlines, the senders and receivers of data and the characteristics of the communication network such as available speed limitations.

4.6.16 UC-ES-13 Hardware design

ID	*UC-ES-13*
Title	Hardware design
Version	1.0
Status	Released
Motivation	To define the hardware with sufficient detail level to support SW development
Stakeholders	Hardware engineer and systems engineer of OEM
Process context	System design
Product context	Product and system/subsystem
Organization context	Engineering department of hardware supplier

This scenario aims at providing accurate definition about the HW platform with sufficient detail level to support SW development. It is not concerned with the logical design of hardware architecture as input for detailed ECU design, wire harness design, etc. However, the difference in terms of information content can be quite small. For example, the pins and wires connecting ECU and a sensor have to be modelled for correct driver configuration, interpretation of sensor reading, etc. At the same time, this is the basis for wire harness design, connector definition, etc. This scenario is performed by the hardware engineer based on systems engineer requirement selected during system design phase of a product or its subsystems.

This scenario reflects a current practice in the sense that hardware design with sufficient detail to support system modelling with allocation, driver SW configuration, bus scheduling etc. has to be supported.

RT-ES-44 **Security aspects to be considered during hardware design**

From security point of view you probably want to disable all kinds of debugging possibilities (physically remove JTAG on production units), consider putting data in internal flash instead of external, whether to use hardware security module or not, and also look at hardware countermeasures against side channel attacks, such as measuring EMF, heat and so on.

Basic activities and process

This scenario provides an accurate definition about the HW platform incrementally according to the inherent structure of the platform and the hierarchy and other characteristics of services that are of concern for the application SW development.

Product information refinement/Transformation flow

This scenario provides a definition of the hardware with sufficient detail level to support application SW development with the following inputs:

- Rough estimation of functional content, HW constraints and requirements.
- Hardware architecture with ECUs, busses, memories, pins, connections, sensors, actuators,

4.6.17 UC-ES-14 Sensor and actuator integration

ID	*UC-ES-14*
Title	Sensor and actuator integration
Version	1.0
Status	Released
Motivation	Sensors and actuators integration as parts of the embedded system
Stakeholders	Hardware engineer, Systems engineer, Application expert, Software engineer of OEM
Process context	This is done with different level of detail on analysis, design and implementation level.
Product context	Product and system/subsystem
Organization context	Engineering department at OEM.

This scenario aims at integrating the sensors and actuators of an embedded system. Performed on the analysis, design, and implementation level of a product or its sub-systems, this scenario involves the software, hardware, and systems engineers, as well as the application experts of OEM.

This scenario is relevant in the sense that sensor/actuators must be considered with sufficient detail and with appropriate structure to support integration on different levels.

RT-ES-45 ***Security aspects to be considered during sensor and actuator integration***

Basic activities and process

This scenario is done with different level of detail on analysis, design and implementation level. Basically, the sensor/actuator-related entities are defined with increasing level of details on lower abstraction levels.

Product information refinement/Transformation flow

This scenario has a refinement flow with information listed below.

Inputs:	*Information about functional input and output.*
Internal_info:	*Mappings of the functional needs on the analysis level onto definitions of sensors and actuators.*
Outputs:	*- Definition of actuator/sensor, its control/interpretation algorithms, its interface in SW (drivers) and HW (electronic components), its pinning and connections to ECU pins, etc.* *- The sensor/actuator HW, control SW, driver SW, interface electronics, pinning/connection definition.*

5 References

[1] Application within Automotive Security and Privacy, HoliSec: Holistic Approach to Improve Data Security, Edition: 2.0

[2] Deliverable D1.1: Needs and requirements; Version 2.0; Vinnova Project name: HEAVENS – HEAling Vulnerabilities to ENhance Software Security and Safety; Vinnova Dnr 2012-04625

[3] SAE J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, 2016

[4] ISO 26262, International standard - Road vehicles – Functional safety, 2011

[5] The CERT C Coding standard, Second Edition, 2014

[6] MISRA C, Amendment 1 for security, 2012

[7] EVITA Deliverable D2.1: Specification and evaluation of e-security relevant use cases

[8] ISO 14229 International standard - Road vehicles -- Unified diagnostic services (UDS), 2013

[9] Deliverable D2: Security models; Version 2.0; Vinnova Project name: HEAVENS – HEAling Vulnerabilities to ENhance Software Security and Safety; Vinnova Dnr 2012-04625

