
# IoT4CPS – Trustworthy IoT for CPS

FFG - ICT of the Future
Project No. 863129

## Deliverable D6.1a
## Architecture for safe and secure automated driving platform demonstrator

The IoT4CPS Consortium:
AIT – Austrian Institute of Technology GmbH
AVL – AVL List GmbH
DUK – Donau-Universität Krems
IFAT – Infineon Technologies Austria AG
JKU – JK Universität Linz / Institute for Pervasive Computing
JR – Joanneum Research Forschungsgesellschaft mbH
NOKIA – Nokia Solutions and Networks Österreich GmbH
NXP – NXP Semiconductors Austria GmbH
SBA – SBA Research GmbH
SRFG – Salzburg Research Forschungsgesellschaft
SCCH – Software Competence Center Hagenberg GmbH
SAGÖ – Siemens AG Österreich
TTTech – TTTech Computertechnik AG
IAIK – TU Graz / Institute for Applied Information Processing and Communications
ITI – TU Graz / Institute for Technical Informatics
TUW – TU Wien / Institute of Computer Engineering
XNET – X-Net Services GmbH



*For more information on this document or the IoT4CPS project, please contact:*
Mario Drobics, AIT Austrian Institute of Technology, mario.drobics@ait.ac.at


Document Control

**Title:** Architecture for safe and secure automated driving platform demonstrator
**Type:** public
**Editor(s):** Edin Arnautovic
**E-mail:** edin.arnautovic@tttech.com
**Author(s):** Denise Ratasich, Edin Arnautovic, Omar Veledar, Leo Botler, Stefan Jaksic
**Doc ID:** D6.1

Amendment History

| Version | Date | Author | Description/Comments |
|---|---|---|---|
| V0.1 | 28.01.2019 | Edin Arnautovic, Denise Ratasich | Initial version prepared |
| V0.2 | 01.02.2019 | Denise Ratasich | Update TUW Contribution |
| V0.3 | 08.02.2019 | Edin Arnautovic | Update TTTech Constribution |
| V0.4 | 15.02.2019 | Christos Thomos | Architectural aspects of Secure and reliable V2X communication |
| V0.5 | 18.02.2019 | Edin Arnautovic | Editing and integrating contributions |
| V0.6 | 20.02.2019 | Omar Veledar | Vehicle Demonstrator description |
| V0.9 | 20.02.2019 | Edin Arnautovic | For review |
| V0.95 | 22.02.2019 | Leo Botler, Stefan Jaksic | Review recommendations |
| V1.0 | 26.02.2019 | Edin Arnautovic | Final |
| V1.1 | 27.1.2020 | Edin Arnautovic | Public version |

The IoT4CPS project is partially funded by the "ICT of the Future" Program of the FFG and the BMVIT.

# Contents

Contents ..... 4
Abbreviations ..... 5
1 Introduction ..... 6
2 Architecture of the Secure and Safe Platform for Automated Driving Use Case demonstration .. 6
2.1 Automated Driving Platform ..... 6
2.2 Self-Healing by Structural Adaptation ..... 7
2.3 Overview of the Platform Demonstrator ..... 8
3 Architectural aspects of Secure and reliable V2X communication ..... 12
4 Cognitive open vehicle platform ..... 16
4.1 Vehicle connectivity ..... 18
4.2 Automotive on-board diagnostic ..... 18
4.3 Vehicle demonstrator architecture ..... 19
5 Conclusion ..... 19

## Abbreviations

| | |
|---|---|
| ADAS | Advanced Driver Assistance Systems |
| AEB | Automated Emergency Braking |
| API | Application Programming Interface |
| ASIL | Automotive Safety Integrity Level |
| CPS | Cyber-Physical Systems |
| ECU | Electronic Control Unit |
| ECU | Electronic Control Unit |
| ROS | Robot Operating System |
| SHSA | Self-Healing by Structural Adaptation |

# 1 Introduction

This deliverable gives an overview of the demonstration of the platform for automated driving. As cyber-physical systems (CPS) exploit a range of capabilities e.g. computation and communication from the cyber sphere and physical objects and processes from the touchable world around us, digitalisation drive is pushing vehicles more than ever towards dependency on computational algorithms, high-performance computing and connectivity. Simultaneously, the inevitable shift towards autonomous driving, but also automation in general, does not couple well with safety-critical tasks. This is due to an underlying conflict between the vast number of possible situations that could be encountered by an autonomous vehicle and the safety domain's perspective, which demands extensive testing of every possibility. The accuracy and apt responsiveness of decisions are adamant factors for safety related functions. Those factors are the result of an ability to sense and analyse situations, as well as capacity to synthesise adequate responses within minimum acceptable time frame. Smart sensor systems and analysis of collected data is far from trivial. As there is a gradual transfer of decision making from humans towards machines in potentially hazardous circumstances that are common place in everyday traffic conditions, it is of utmost importance to ensure correct operation of all underlying technologies. That includes operation throughout the vehicle and in the outside environment, if the vehicle is connected. The full chain must be verified to be safe, secure, efficient and reliable.

# 2 Architecture of the Secure and Safe Platform for Automated Driving Use Case demonstration

Modern cars contain many different functions in all domains, such as Advanced Driver Assistance Systems (ADAS), infotainment systems, chassis and powertrains. Historically, for every new function, yet another electronic control unit (ECU) was developed. This solution does not scale well (due to high costs, flexibility, wiring complexity and weight, etc.). Thus, a consolidation and centralization of automotive functions is needed, and the implementation of new customer functions will be controlled by software only, effectively reducing the number of ECUs and opening possibilities for new features.

A central high-performance safe and secure central domain ECU will be a basis for future architectures for automated driving. For example, in the ADAS domain, the requirement for central sensor fusion quite naturally leads to a central fusion and application controller architecture. Generic platforms are needed to foster function SW reuse across different car models, which in turn is a mean to cope with the immense costs and efforts for function validation especially in the ADAS domain.**Automated Driving Platform**

Hardware, software and networking requirements on automated driving systems are influenced by safety, performance, integration efforts, modularity, scalability and other aspects. From the functional perspective, automated driving systems consist of multiple functions that require different properties from the underlying hardware and software platform. Figure 1 shows a general Automated Driving System Architecture. Sensor data (from radars, cameras, LIDARs, etc.) is integrated using sensor fusion to create a model of the environment (static and dynamic). This model is used to compute the driving strategy and control algorithms to control steering, breaking and the powertrain. Additional ADAS functions such as Automated Emergency Braking (AEB), lane assistance and surround view will be also deployed on such platform.

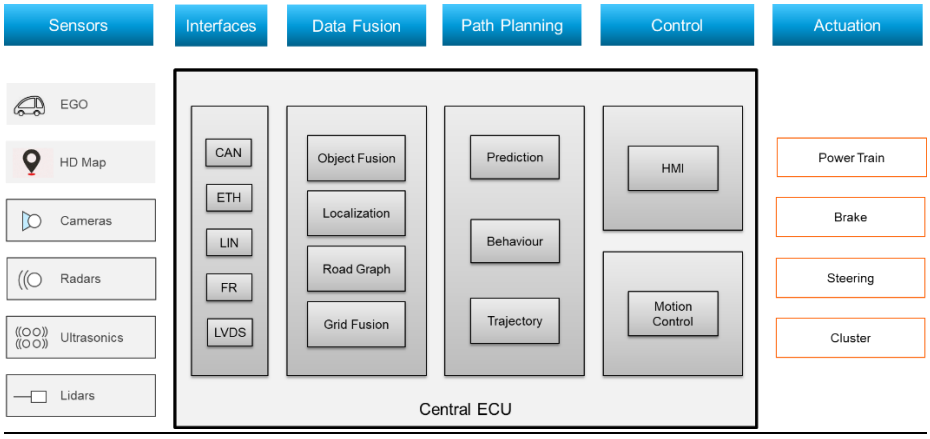


*Figure 1: General Automated Driving System Architecture*

This functional architecture has to be mapped to a concrete hardware architecture by the allocation of functions to hardware components. The architecture of the use case demonstrator consists of different computing modules: a safety microcontroller, two high-performance CPUs based on ARM architecture. These devices are connected by a Deterministic Ethernet (DE) switch. External interfaces such as CAN or Ethernet are also provided. Besides the normal best-effort Ethernet switching capability, the DE switch provides a time-triggered, deterministic mode of operation where all communication and switching takes place according to a predefined schedule in a completely predictable way. All processing elements are synchronized to the switch and schedule their internal tasks accordingly. It would be very difficult to develop software directly on the top of such heterogeneous hardware architecture,. The applications running on different CPUs (and operating systems) would have completely different environments and APIs. In addition, some basic mechanisms such as communication, safety features or data management would have to be treated differently. Applications would need to implement proprietary adaptation layers, and this would attach the application's code too strongly to the underlying hardware and software environment. Safety software platform on tops of this hardware architecture offers common execution environment, API, standardized abstraction from underlying hardware and operating systems, and location transparency. Location transparency means that software components can be developed independently regardless of the hardware constraints and can be also moved from one CPU to another depending on, e.g., safety requirements or CPU utilization. From the safety perspective, the platform offers safety execution of the software components as well as safe communication channel. Safety execution includes memory protection, timing supervision task supervision, etc.

## 2.2 Self-Healing by Structural Adaptation

Self-healing by structural adaptation (SHSA) replaces a failed component during runtime by exploiting redundancy [Hoeftberger2015], [RHISG2017].

In particular, SHSA can be used to replace failed observation data. It monitors and substitutes CPS variables (cf. signals) we refer to as *information atoms* (itoms) [Kopetz2014] in messages communicated between application components (e.g., sensors and controllers) based on a knowledge base modelling the relations between these itoms (Figure 2).

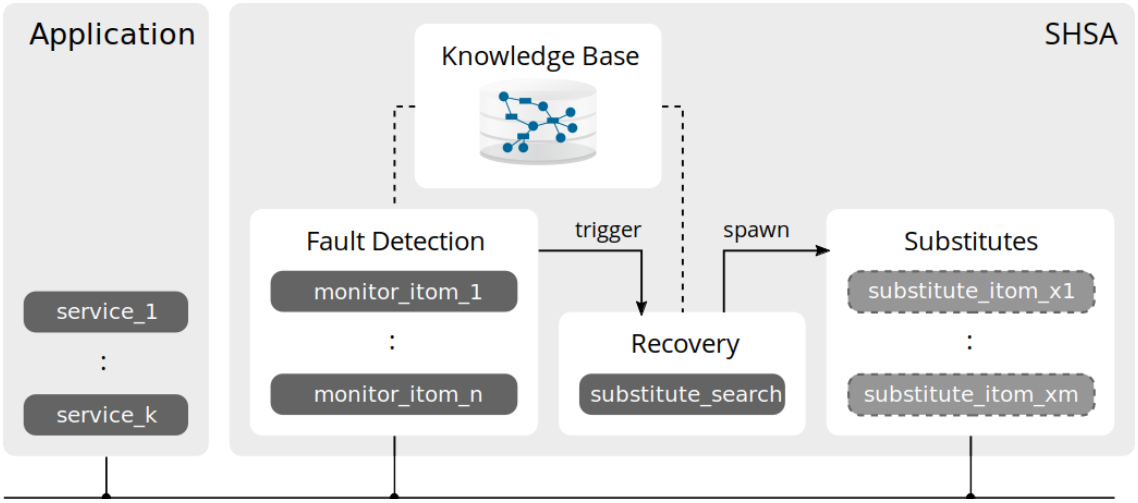


*Figure 2: Building blocks of SHSA.*

The monitors subscribe to related itoms and compares its values over time. Once the monitor detects a failed itom, it triggers the recovery process to search for a substitution in the knowledge base. The recovery spawns a substitute component publishing the previously failed itom.

In this work package we apply SHSA on an automotive use case. First results are:

- Demonstration of substitution in a first prototype – a mobile robot [RPSG2018].
- Case study of SHSA in a vehicular network [RKGGSB2019].
- Architectural requirements for SHSA and considerations w.r.t. security,

And the next steps are to:

1. equip the mobile robot [RPSG2018] with the safety platform from TTTech, and to
2. demonstrate fault detection exploiting implicit redundancy.

SHSA is developed in WP3, see D3.1 for an introduction and overview. Case studies, description of the demonstrator and experiments can be found in [RHISG2017],[RKGGSB2019].

## 2.3 Overview of the Platform Demonstrator

Use case demonstrator will show Self-Healing by Structural Adaptation on the safety-related automotive computing platform. SHSA has been initially showcased on a mobile robot avoiding collisions [RPSG2018] which is used as a starting point for the demonstrator. The mobile robot's application has been distributed onto two boards featuring ARM cores running Linux and the Robot Operating System (ROS). Though SHSA targets resilience its underlying platform (processor, OS, middleware) has limited capabilities regarding dependability or security. However, many CPS applications demand a resilient platform.

Our goals are therefore:

- Showcase a resilient platform.
- Port a ROS prototype or parts of it onto a resilient platform.

- Integrate SHSA with a resilient platform and middleware.

**Robot Operating System**

ROS [ROS][ROS_Wiki] is a robotic framework and mainly provides a service-oriented middleware to connect the processes – *ROS nodes* – of a distributed application. Nodes communicate via a message-based interface over TCP/IP sockets. In particular, ROS nodes publish and subscribe to *ROS topics* (cf. named channels). ROS can start new nodes and adapt the information flow of existing nodes during runtime and is therefore suitable for SHSA [RHISG2017][RPSG2018].

However, ROS doesn't provide security-related services, e.g., authentication or encryption. Once in the ROS network, any ROS command can be executed. The guidelines on ROS security [ROS_Security] propose to operate ROS only in a separate network (not connected to the Internet) or VPN, and provide services over the *rosbridge* (a websocket interface) if public access is necessary. Moreover, as ROS is running on top of Linux and using standard Ethernet, it is also not suitable for safety applications. Nevertheless, ROS remains the state-of-the-art *prototyping* platform for robotics and self-driving cars (see the programs of the ROS conferences [ROSCon]).

In late 2017 the first official release of ROS 2 became available which is intensively developed to overcome the shortages of ROS. The communication is based on the Data Distribution Service (DDS) [DDS][DDS_Standard] that is a light-weight and decentralized middleware providing real-time capabilities, quality-of-service (QoS) policies and security features. The resilient platform could provide a ROS 2 interface to benefit from the ROS community, e.g., by re-using drivers (for sensors and actuators), state-of-the-art (robotic) algorithms (e.g., localization, path planning) and tools (e.g., debugging, logging, visualization).

**Use Case**

The mobile robot in Figure 5 is able to move and avoid collisions autonomously.

The application is distributed into several nodes representing interfaces or drivers to the sensors and actuators, and controllers (Figure 6, cf. automated driving architecture in Figure 4).

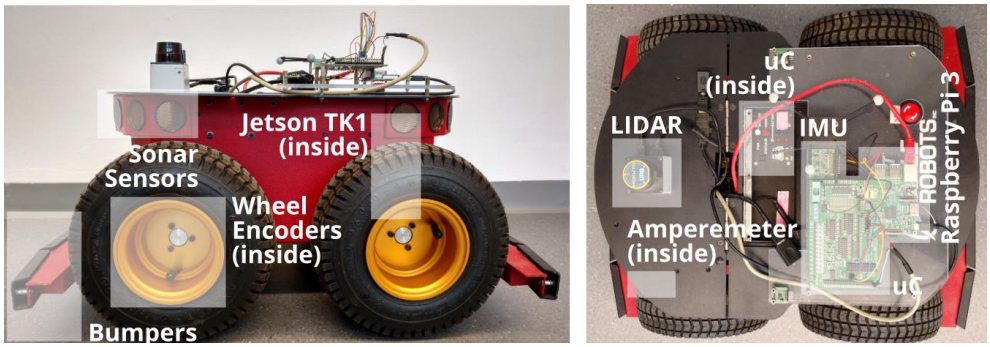


*Figure 3: Mobile robot equipped with its sensors and processing units [RPSG2018].*

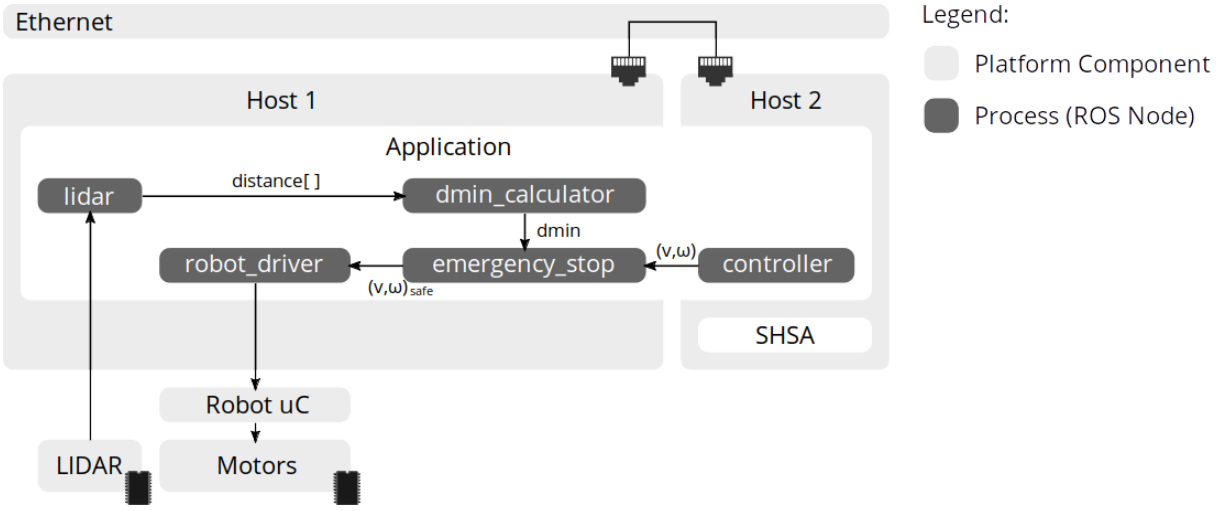


*Figure 4: Platform overview and nodes of the application.*

In particular, a controller sends the desired linear and angular velocity (v,ω) to the robot's microcontroller (Robot uC) controlling the wheel motors. The LIDAR (or laser scanner) on top of the rover provides distance measurements within 270°. When the minimum distance in front of the rover (dmin) – calculated by another ROS node dmin_calculator – falls below a threshold, the robot is stopped, that is the velocity commands from the controller are replaced by (0,0) (which is implemented by the node emergency_stop). Acceptance of the controller commands is resumed when dmin again exceeds the threshold.

Figure 6 shows how could we split the application to run on two hosts. Host 1 holds the critical nodes namely the ones to avoid a collision. Host 2 implements the controller, i.e., publishing velocity commands for robot movement..

**Platform Integration**

The rover is currently equipped with a Jetson TK1 (NVIDIA GPU board) and a Raspberry Pi 3 (RPi), both running Linux and ROS, and microcontrollers (uC) sampling sensors and controlling the wheel motors. The rover will be equipped with TTTech's safety platform extending and/or replacing the existing platforms.

Figure 7 shows the envisioned network and computing architecture of the proposed demonstrator. It should contain TTTech's Safety Platform and one computing board with Linux and ROS (during the initial developments, the system can also contain two such boards as mentioned above). TTTech's safety platform contains three CPUs connected over a Deterministic Ethernet switch (SW). Safety CPU (Safety Host – SH) is capable of execution safety related functions up to ASIL-D standardized automotive safety level and it runs a specialized real-time safety operating system (Safety OS). Two performance CPUs (Performance Host – PH1, PH2) are multicore CPUs based on ARM and offer processing power necessary for ADAS and automated driving applications. PH1 and PH2 run Linux.

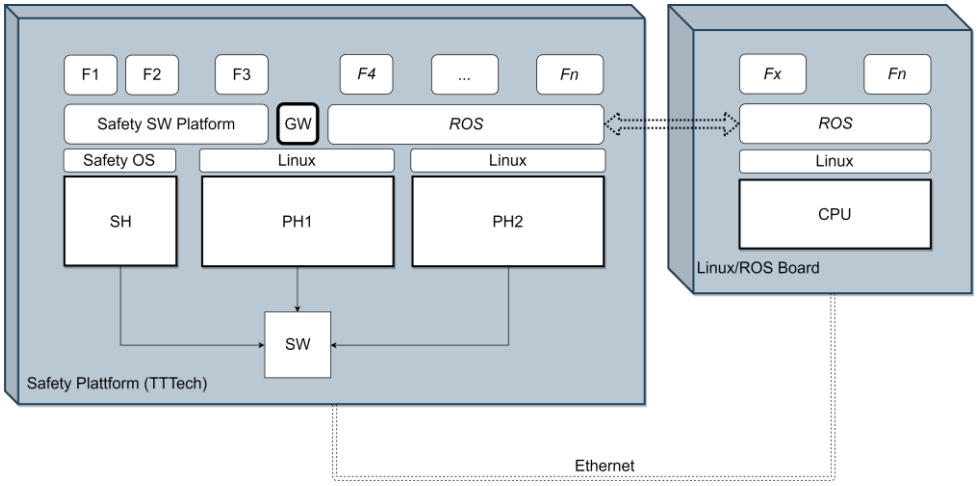


*Figure 5: Safety Platform – SHSA integration*

Table 1 presents the possibilities to deploy the application to the safety-related platform.

| # | lidar | dmin_calculator | emergency_stop | robot_driver | controller | shsa |
|---|---|---|---|---|---|---|
| 1 | Jetson | PH2 | PH2 | Jetson | RPi | RPi |
| 2 | PH2 | PH2 | PH2 | PH2 | RPi | RPi |
| 3 | PH1 | PH1 | PH1 | PH1 | PH2 | PH2 |
| 4 | SH | SH | SH | SH | PH2 | PH2 |

*Table 1: Possibilities to map the nodes to the platforms: Safety Host (SH, microcontroller, bare or possibly running a real-time OS), Performance Host 1 (PH1, real-time OS), Performance Host 2 (PH2, Linux+ROS) on the Safety Platform, existing Jetson and Raspberry Pi (RPi, Linux+ROS).*

PH2 shall run Linux and ROS to easily migrate the existing application. The node emergency_stop is a simple ROS node subscribing and publishing to topics (it does not interface to HW like the lidar node) which makes this node a good first candidate to migrate to a performance host of Safety Platform (Table 1). Similar is true for dmin_calculator.

Depending on the available interfaces of Safety Platform, the sensor drivers can be moved next. The controller and SHSA may be kept on the Raspberry Pi (Table <mapping> #2). In contrast to the hosts on the Safety Platform, the RPi provides a WiFi interface which may come in handy to visualize data and/or tele-operate the rover for demonstrating purposes. However, controller and SHSA could run on the Safety Platform PH2 too.

Finally, the critical part can be deployed to a host with real-time capabilities, PH1 (Table 1 #3) or the SH of the safety platform (Table 1 #4). However, this would require a reimplementing of several nodes (including the sensor drivers), integration into the Autosar stack in case of SH, and a ROS gateway to

benefit from ROS capabilities (e.g., logging, visualization, existing robotic algorithms). The Safety Host and Performance Host are connected via Ethernet to PH2. Because all critical nodes already run on the SH or PH1, the best-effort Ethernet switching capability could be used to receive controller inputs.

To summarise, the proposed integration provides following resilience features on different CPS layers (physical/hardware, communication and information):

- Safety-critical application components run on a resilient platform enabling deterministic execution and providing fault-tolerance on the physical layer by redundant cores (space) or lockstep execution (time).
- Safe communication for monitoring, logging and visualization by deterministic Ethernet isolating critical (time-triggered) and non-critical (best-effort) messages.
- Additional fault-tolerance on the information layer is provided through SHSA.

# 3 Architectural aspects of Secure and reliable V2X communication

V2X communications can be defined as the bidirectional transmission of accurate, reliable and timely information from/to a vehicle to/by any other entity that is possible to have an impact to it. A primary goal is vehicle's safety and prevention of accidents by extending the sensing capabilities of other on-board sensors, as well as vehicle's computing power using specialized ad-hoc and cloud network services. V2X connectivity includes four main sub-categories, namely vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I – road infrastructure), vehicle-to-network (V2N – backend/internet), and vehicle-to-pedestrian (V2P) communications. These can be extended by including a variety of other use-cases like vehicle-to-home (V2H), vehicle-to-grid (V2G), and vehicle-to-device (V2D) connectivity (e.g. 3GPP TS 22.185, 3GPP TR 22.885, TR 22.886 and TS 22.185). An architecture like this is illustrated in Figure 6. The information acquired from the vehicle's on-board sensors has to be transmitted to its surrounding, i.e. nearby vehicles, the road infrastructure or pedestrians, and to the network. At the same time, the vehicle's sensors information could be fused and enhanced by the dynamic information that is received from the same surrounding entities. Alongside, cellular base stations are gathering all behavioural data of the vehicles and their environment, in order to provide specialized cloud services for critical software and firmware over-the-air upgrades, lifecycle management, predictive maintenance processes, remote monitoring, etc.

These interactions make V2X connectivity a key enabler and a vital component for autonomous and automated driving. When fully enabled, this technology will offer a wide range of benefits, able to ensure vehicle's, passengers' and pedestrians' active safety by extending driver's viewing and visibility range, increasing situational awareness beyond line-of-sight, adding cooperative driving features and automated vehicle takeover. At the same time, V2X connectivity will enhance driving comfort by providing novel infotainment capabilities (e.g. augmented navigation, 4k 360 VR surround view, real-time multimedia streaming) and innovative cloud services such as those mentioned previously, covering a plethora of use-cases which exploit information for road conditions (V2N, V2I), local traffic congestion data, jam avoidance and alerts (V2N, V2V), nearby accidents (V2V, V2N), blind intersections (V2V, V2I), local weather and environmental conditions (V2N, V2I), urgent road hazard warnings (V2I), traffic control signals and signs (V2I), pedestrian crossings (V2I, V2P), forward collision, emergency brake light, sudden lane change and other nearby vehicle warnings (V2V), parking information (V2N,

V2I), theft monitoring (V2N), emergency-call (V2N), eco-driving and optimization of road network (V2N, V2I), instant FOTA/SOTA updates (V2N), data uploading for deep learning algorithms and big data analytics (V2N), and many other types of alerts, warnings and notifications.

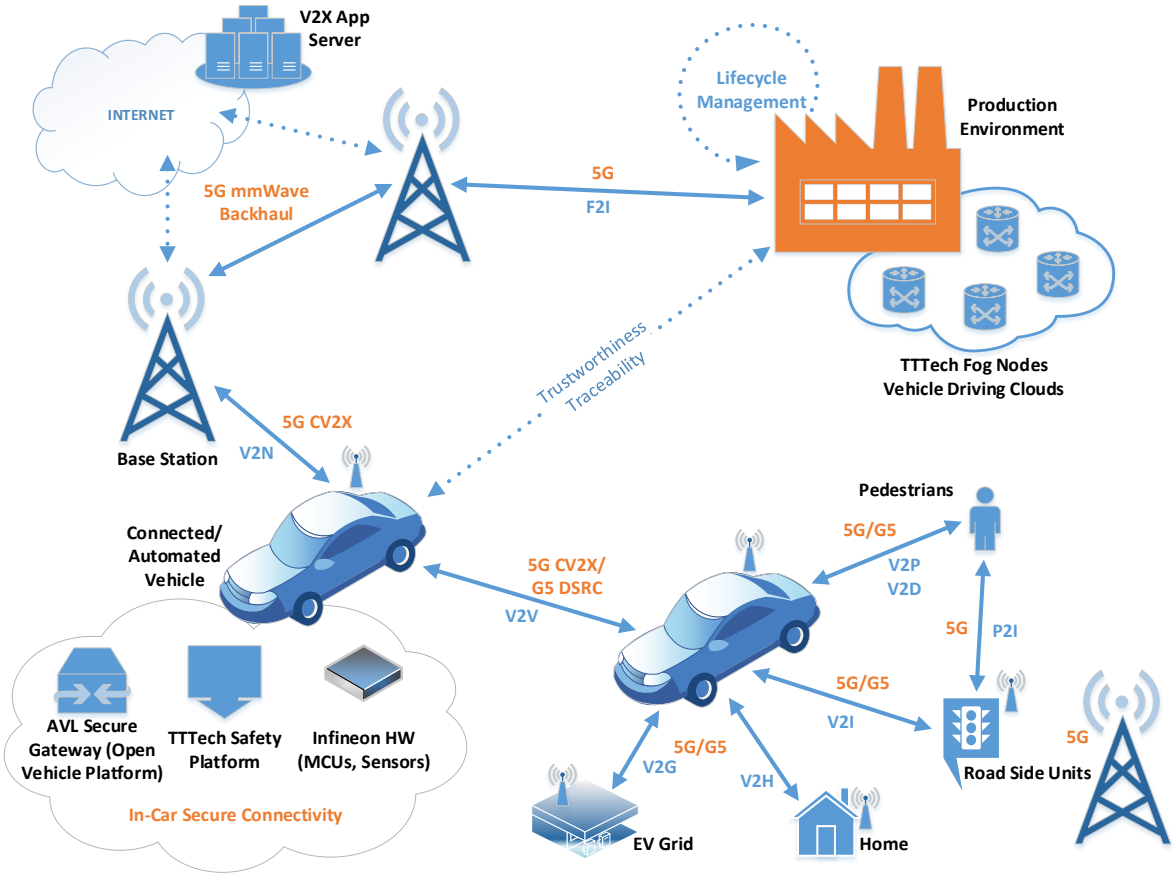


*Figure 6 V2X architectural aspects for secure and reliable communications*

For this purpose, transceiver modules dedicated for V2V, V2I, V2N, V2P and in-vehicle communications are installed peripheral on the vehicle, inter-connected to the on-board advanced central safety platform and secure gateway (networking vehicle sensors, actuators, storage and computing systems, with Electronic Control Units – ECUs and Telematics Control Units – TCUs) and supported by cellular base stations that serve as the infrastructure for reliable and trustworthy connectivity.

Vehicle connectivity currently exists in SAE level 0 and 1 vehicle use-cases but in a basic form, primarily for non-safety critical applications (e.g. navigation, internet access, voice/video communications, emergency call, infotainment, etc.) with just a growing demand on higher bandwidth. External and in-vehicle connectivity use-cases are mostly supported by integrating personal smartphones into the vehicle platforms (e.g. Apple CarPlay, Android Auto, MirrorLink, etc.), by creating vehicle Wi-Fi hotspots and by exploiting other types of vehicle wireless access technologies such as NFC and Bluetooth for acquiring data from vehicle sensors or for trivial control and for audio/video applications. Since the reliability of present wireless communication technologies could not guarantee robust links that are sufficient enough to enable safe and secure V2X applications, only advanced on-board state-of-the-art sensors (e.g. HD cameras, radars, LIDAR, GNSS, ultrasound, etc.) and other platforms are used for safety and security mechanisms.

However, for SAE Level 2-5 safety-critical applications that support use-cases like the ones mentioned above, a definitely different level of dependability, latency and capacity of wireless data communications is necessary in order to empower V2X connectivity as a sensor extension that is able to provide redundancy when on-board sensors fail or underperform. A connected autonomous vehicle will strongly depend upon the integration of profoundly heterogeneous connectivity technologies (both for in-vehicle and external radio access), with immense interoperability and cross-interference issues and a great difference in technology lifecycles between the vehicle and the communication modems. On top of that, wireless links should be able to operate successfully in an extremely complex vehicular environment in terms of mobility and dynamics, with highly diverse and contradictory demands with respect to the communications characteristics (e.g. network density, range, operational efficiency, throughput, latency and response schemes in cases of misbehaviour and loss of connectivity), as well as to other characteristics such as positioning accuracy, security, privacy and functional safety. Furthermore, since wireless connectivity introduces multiple security vulnerabilities either into the vehicle's control, infotainment, telematics and navigation systems or the supporting cloud infrastructure, security-by-design will be imperative for sandboxing communication system drivers, embedding security along the entire V2X development phase and ensuring end-to-end encrypted communications and data privacy safeguarding autonomous vehicles on a system level.

Currently, requirements, specifications and evaluation metrics for V2X wireless link quality are being derived from typical connectivity use cases that assume classical mobility and traffic patterns of generic mobile devices and rather static infrastructure environments for both safety and non-safety applications. However, these may not take into account the variety of V2X use cases, limitations of current communication technologies (the huge amount of vehicle generated data and low latency support may challenge even the next generation cellular networks protocols in terms of capacity, efficiency and cost), relationships between conflicting connectivity requirements, and exceptional challenges in combining connectivity, security and safety needs. At the moment, an initial evaluation of the quality of a V2X communications technology is performed using end-to-end latency, communication range and coverage, position accuracy, data rate and reliability. The assessment of fundamental parameters can determine the main V2X technology options, system architectures, constraints and HW specifications that fulfil the promises of each use-case connectivity scenario. Preferably, the capabilities of connectivity technology candidates and limitations of today's vehicle embedded systems have to be examined and resolved very early in the entire V2X system design through simulations and large sets of real-world field trials for as many V2X use-cases as possible. Modelling, development, integration and testing of components and systems on this kind of environment is an exceptional challenge. As more connected and autonomous vehicles roll up on the roads and become part of this complex ecosystem, these requirements will become more and more strict and demanding for unfolding the full potential of autonomous and cooperative driving.

Autonomous driving ecosystem must natively provide solutions that deal with systematic and random faults due to V2X connectivity technology limitations and vehicle dynamics alongside with the increasing complexity of autonomous driving systems for an early and inherent system integration of such features. It is a common belief that a commitment in power and resources from all V2X ecosystem stakeholders is necessary in order to establish a complete V2X connectivity framework that fulfils all of the ambition towards the evolution from SAE Level 3 to 5. An extension to the ISO 26262 functional safety standard (ASIL B to D) and IEC 61508 standard for general industrial E/E/PE systems functional safety is also essential in this regard. This evolution is often firmly associated to the development of

5G-and-beyond communications technology which is expected to support all the requirements of autonomous vehicle use cases as explained below (3GPP TR 22.885/TR 22.8863/TR 22.185/TR 22.891, 5G-PPP, 5GAA WG1, NGMN Alliance, EATA – European Automotive and Telecom Alliance, ISO Standards for Intelligent Transport Systems).

As far as external V2X connectivity is concerned, currently there are two major options which are DSRC (Dedicated Short Range Communications based on IEEE 802.11p WLAN standard) and 3GPP C-V2X based on cellular 4G and the upcoming 5G technologies. Both of these technology options focus on the definition of lower layers in order to offer long range, non-line-of sight capabilities as an extension of other vehicle sensors to add a redundancy factor to them towards enhanced vehicle safety and cooperative driving features. The upper layers are based on standardization activities such as IEEE (e.g. 1609), SAE (Society of Automotive Engineers), IETF (Internet Engineering Task Force), ETSI (European Telecommunications Standards Institute), ISO and CEN (European Committee for Standardization).

DSRC technology (also known as ITS-G5 in Europe and WAVE – Wireless Access in Vehicular Environments in US) was developed for safety-critical V2V and V2I use cases, by specifying ad-hoc networks (no communication infrastructure is necessary) in dedicated license spectrum bands (5.9 GHz ITS band), aiming at the exchange of basic safety messages with very low latency (<50 ms) for a radius of up to 2 km, using public key infrastructure and data security based on IEEE 1609.2. The physical layer and lower MAC is based on the IEEE 802.11p WLAN standard, the work of which has initiated in 2004 and finished in 2010. The IEEE 1609 Family of Standards addresses issues of interface homogeneity between different car manufactures and provides a sufficient foundation regarding the organization of management functions and modes of operation of system devices for ubiquitous high-speed communications between vehicles and service providers. For example, network and transport layer services for devices and systems are provided in IEEE 1609.3 standard (WAVE short message protocol – WSMP), and Upper MAC layer is an extension of the IEEE 1609.4 standard. Data link layer is defined by the IEEE 802.2 standard and upper layers are implemented using the IPv6/UDP/TCP elements (IETF RFC 2460/768/793). Additionally, in Europe ETSI EN-302-636 series of standards define GeoNetworking (Network layer protocol) which is a geographically aware routing technique establishing an ad-hoc network that is efficiently arranged according to the physical locations of nodes, offering connectivity over multiple wireless hops in order to extend the communications range. Society of automotive engineers (SAE) specifies also standards for V2V communication applications based on IEEE802.11p technology such as the SAE J2735 that defines a message set dictionary, its data frames and data elements and SAE J2945 family of standards that sets a system environment for the information exchange between a host vehicle and other DSRC enabled devices in order to address safety, mobility and environmental system needs. Other relevant standardization efforts for Cooperative Intelligent Transportation Systems (C-ITS) in Europe are based on ETSI TS 102/103.x families.

IEEE802.11p technology, in the form of WAVE and ITS-G5, has been extensively tested and validated for a broad range of use cases by many vehicle ecosystem stakeholders over the last few years and is currently in a very stable form. Its availability is also very mature and many products are ready for deployment even within this year (e.g. C-ITS by CAR 2 CAR Communications Consortium (C2C-CC) in Europe).

However, further advancements of this technology over the next years are hardly to be realized, which prevents fulfilling the needs of future V2X use cases. In the long run this gives an obvious advantage to 3GPP Cellular V2X (C-V2X) technology which will leverage on the gradual evolution from 4G LTE to

the 5G NR air interface including many new features, better QoS support, solutions for functional safety requirements, better coexistence with other technologies, cost efficiency and backwards compatibility. Currently, the autonomous driving requirements are challenging for both 4G LTE C-V2X and DSRC technologies, but 5G NR progressive advancement promises much higher data rates, ultra-low end-to-end latency, higher capacity and reliability, greater coverage and number of device connections, enhanced security, location precision, energy efficiency, consistent performance and full application support that can address ultra-reliable and low-latency connectivity for mission-critical services, device-to-device (V2V, V2I) communications, enhanced mobile broadband (V2P, V2N) and massive machine type communication use cases which is exactly what is needed for a fully autonomous vehicle ecosystem. 3GPP has started in Rel. 14 the development of V2X connectivity features (Phase 1, Basic Safety based on Proximity Services (ProSe) and PC5 interface, 2017), certain key features have gradually matured and added in Rel. 15 (Phase 2, Enhanced V2X, June 2018) but there are still many open issues that need to be treated and solutions to be formalized on the upcoming Rel. 16 (Phase 3, strengthening of safety applications towards autonomous vehicles, 2020) and Rel. 17 which has also been planned for the following years. The upper layers of LTE C-V2X are making use of the same standards as the IEEE 802.11p based technologies, since these are specified by the automotive industry (e.g. SAE J2735, IEEE P1609.x, ETSI ITS, ISO/IEC 8824, etc.). However, for the reason that these two technologies have been implemented independently, they are having different assumptions and principles, and so far, there exist no interoperability mechanisms rendering them incompatible.

C-V2X technology is much more modern than DSRC for direct communications (almost two decades separate them), therefore the level of maturity for C-V2X technology is still very low and it will take quite some time until wide-scale thorough field testing has been successfully completed and products are available for deployment. This situation leads to a dilemma within automotive manufacturers for a successful selection among these two V2X technologies. Since a vehicle lifespan is about ten years, products with DSRC technology might occupy the upcoming generation of vehicles in the 5.9 GHz ITS band. In the meantime, C-V2X technology will enter the market (expected at early 2020s) as a complementary technology for V2V and V2I connectivity (safety critical ADAS and autonomous and driving functions) and will enhance V2X connectivity with its unique V2N and V2P capabilities.

For faster deployment of C-V2X technology, wireless infrastructure technology advancements and achievements by the relevant stakeholders will play the most crucial role, since this is the important differentiating factor from a useful but outdated DSRC technology. In this context, IoT4CPS will examine 5G PHY layer technologies for the cellular access points, HW architectures and behavioural models, transceiver modules capabilities and limitations, specifications for key building blocks that can compellingly address the requirements of secure C-V2X connectivity taking also into account the progress for the vehicle communication modules, sensors, actuators and computing hardware, as well as the solutions and tools developed for secure and safe autonomous driving platforms.

# 4 Cognitive open vehicle platform

As the need for vehicle E/E architectures with centralised high-performing ECUs is increasing, especially due to the trend of autonomous driving, the intent is increasing towards migration of control strategies to more sophisticated computing platforms to ensure optimal operation. These also must include appropriate test environments to eliminate safety risks and offer possibilities for implementation of mitigation strategies.

The current offerings of heterogeneous solutions and lack of standardisation, are creating a push towards usage of technology agnostic platforms. Provision of unique solutions that are only applicable to a small range of vehicles, limits software portability across different vehicles. Integrating secure solutions with enabled privacy preserving options deepens trust and offers a possibility to conquer resistance to new technology that could potentially be deemed as unsecure and unsafe by the society.

The safe and secure platform presents a trusted computing base that is potentially secure from other software and hardware influence. It offers several possibilities to help resolve current challenges presented by the autonomous driving. Its computational power enables monitoring, control and collection of vehicle data close to the physical world. The analytics performed near the source, at the edge of IoT network, help ensure component operational integrity and supports data security.

The key factors of interest in the use case are security, safety and reliability of the intelligence at the edge of IoT network and connectivity solution. The use case does not consider different sensor types and their fusion but is interested in hardware and software platforms that enables data collection from those sensors, as well as processing and communication of the same to the outside world. The encompassed IoT connectivity solution, which is prone to cyber-attacks and could be easily compromised, is considered so that it is possible to integrate security algorithms. As the architecture includes centralized high-performance in-car computing, it unavoidably contributes towards distribution of intelligence which in turn endorses reliable and robust operation. The proposed architecture components are presented in Figure 9. The architecture components are the ones marked as '*within scope'*. The components marked as '*testability*' may be included for demonstration purposes, but are outside of the scope of task. The connectivity security is also prepared, but not implemented in this task. The eventual potential applications that are not within the scope of the task, but could be used for demonstration are: optimisation of traffic flow, parking, energy consumption and emissions predictive energy management, exhaust after-treatment control strategy, traffic light assistance etc.
The proposed concept is to demonstrate safe and secure high-performance platform for autonomous driving. It utilizes multi-CPU control unit with real-time, safety-related software platform. Focus is shifted away from algorithms and autonomous driving function development towards hardware and software platform.

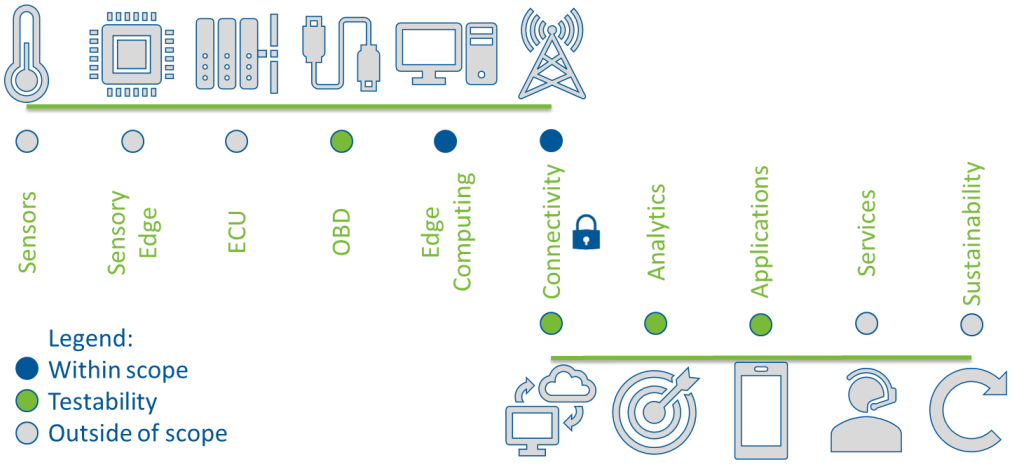


*Figure 7: Proposed architecture components*

## 4.1 Vehicle connectivity

The present vehicle connectivity options deliver limited data collection and analysis functionality and do not guarantee support for safety critical applications. Simultaneously, the growing number of services, be it operational, safety relevant or entertainment, are increasing demands for data throughput. The extent of IoT involvement in the current connectivity solutions is limited. There is still expectation for development of a beneficial wireless technologies that could provide safe and secure communication adequate for autonomous driving applications. Hence, the proposed solution is directed towards resolution of some of those issues. It will integrate existing ideas and will be supported by existing tools. The solution must relate to automotive safety standard ISO26262, however is also to contribute towards increased influence of E/E architecture on environmental standard e.g. energy consumption or pollution. The IoT connectivity is to enable combining of the newly gathered vehicle data with the existing knowledge and hence, realise true value and generate insights that are currently not available. The aim of such digital feedback loops is to develop new control strategies, as well as support maintenance services through realisation of the digital transformation.

The connectivity solution should integrate existing IoT platform standards for interoperability between smart components. Where beneficial, these standards could be extended if they do not cater for the needs of the demonstrator. Semantic interoperability is a crucial asset to achieve the desired intelligent behaviour in the system, but also to help gather and analyse onboard vehicle data and integrate with other data sources.

The potentially sensitive data demands secure communication. Hence, the offered solution must have the ability to enable privacy-preserving authentication, data confidentiality mechanisms and definition of communication protocols and high-level APIs. The connectivity option also presents an opportunity for remote updates of relevant software and firmware. Rather than freely rolling out such updates, they must be synchronised with safety standards. The secure communication is one of the prerequisites for such a service.

## 4.2 Automotive on-board diagnostic

Automotive on-board diagnostic (OBD) systems provide access to vehicle's self-diagnostic and reporting capability. They aim to provide data about critical components during vehicle operation. The available data, which is sourced from the vehicle's control units, informs about the state of a running vehicle and its various sub-systems. Some data is exploited to alert drivers in event of severe vehicle malfunctioning. Certain OBD functionality is also legally binding on new vehicles.

The indirect power of OBD systems stems from the available access to a range of vehicle data. Various tools can access OBD functions. The complexity of the tools varies from simplified consumer level products to sophisticated telematic analysers. Majority of the tools focus on detection of faults. Some of those are aimed data logging for post-processing, but a trend towards real-time display of data is also generating pace. Some of the current applications include fleet monitoring and monitoring of energy efficiency.

## 4.3 Vehicle demonstrator architecture

The proposed hardware configuration safe and secure automated driving platform demonstrator is shown in Figure 10. Data is collected via OBD from the vehicle's ECUs. OBD device is connected to the centralized safety-related platform. The secure platform is to enable gathering of vehicle data at the edge of the network. As the computing power is shifted towards the edge, the proximity of the computing tasks to the vehicle itself is to remove the time latency issue for the safety-critical application, reduce the need for huge amounts of data traffic and hence, advances data analysis efficiency and eases IoT security requirements due to minimised data set to be transported. The resulting data accuracy is to enable further enhancement of traffic safety. The hardware platform will integrate safety software platform. It may host some automotive applications for demonstration purposes. The additional functionality is added by an AVL device that would provide interoperable IoT connectivity to cloud and data brokerage.

Planed activities are to involve some simulations, hardware integration and verification, construction of prototype hardware gateway and preparation for integration of security features. Retrospective addition of security features at the end of the design processes is not seen as a feasible option. Hence, the security aspect is to be considered from the design stages.

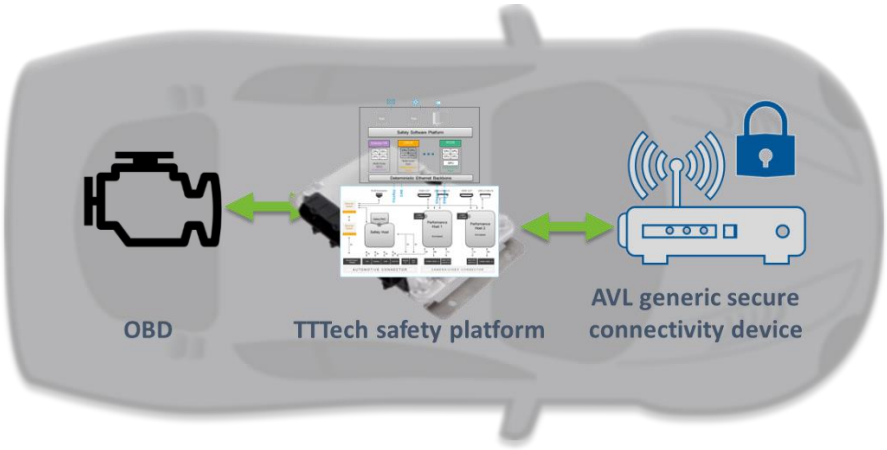


*Figure 8: Vehicle Demonstrator Architecture*

# 5 Conclusion

Different platform aspects for automated driving presented above serve as the architectural basis for the IoT4CPS demonstrations in WP6. These demonstrations will include the research results from the technical work packages and will integrate several approaches into technical demonstrators. For example, the safety-related automotive platform will serve as the underlying computational infrastructure for the approach for *Self-Healing by Structural Adaptation* implemented initially only on a prototypical development boards and software. From the connectivity side, this deliverable presented different architectural aspects of V2X communication. Autonomous driving ecosystem must natively provide solutions that deal with systematic and random faults due to V2X connectivity technology limitations and vehicle dynamics alongside with the increasing complexity of autonomous driving systems for an early and inherent system integration of such features. V2X is also a prerequisite for the smart instrumentation and open vehicle platform.

References

[DDS] G. Pardo-Castellote, “OMG Data-Distribution Service: architectural overview,” in 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings., 2003, pp. 200–206.

[DDS_Standard] “DDS Portal – Data Distribution Services.” [Online]. Available: https://www.omgwiki.org/dds/. [Accessed: 29-Jan-2019].

[Hoeftberger2015] Oliver Höftberger. Knowledge-Based Dynamic Reconfiguration for Embedded Real-Time Systems. PhD thesis, TU Wien, Institute of Computer Engineering, Wien, 2015.

[Kopetz2014] H. Kopetz. A Conceptual Model for the Information Transfer in Systems-of-Systems. In 2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), pages 17-24, June 2014.

[RHISG2017] D. Ratasich, O. Höftberger, H. Isakovic, M. Shafique, and R. Grosu. A Self-Healing Framework for Building Resilient Cyber-Physical Systems. In 2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC), pages 133-140, May 2017.

[ROS] M. Quigley et al., “ROS: an open-source Robot Operating System,” in ICRA workshop on open source software, 2009, vol. 3, p. 5.

[ROSCon] “ROSCon 2018 Program” [Online] Available: https://roscon.ros.org/2018/#program. [Accessed: 01-Feb-2019].

[ROS_Security] “Security - ROS Wiki.” [Online]. Available: http://wiki.ros.org/Security. [Accessed: 29-Jan-2019].

[ROS_Wiki] Robot Operating System Wiki, “ROS Documentation - Introduction,” May-2014. [Online]. Available: http://wiki.ros.org/ROS/Introduction.

[RKGGSB2019] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci. A Roadmap Towards Resilient Internet of Things for Cyber-Physical Systems. IEEE Access, pages 1-24, Jan 2019.

[RPSG2018] D. Ratasich, T. Preindl, K. Selyunin, and R. Grosu. Self-healing by property-guided structural adaptation. In 2018 IEEE Industrial Cyber-Physical Systems (ICPS), pages 199-205, May 2018.