SDV Alliance Integration Blueprint

This whitepaper outlines an integration blueprint for the use of standards and technologies from the SDV Alliance partners: COVESA, Eclipse SDV, AUTOSAR, and SOAFEE. The blueprint presented here is orchestrated by digital.auto, utilizing technologies and products from Bosch, ETAS, and AWS.

Bosch: Nhan Luong Nguyen, Achim Henkel, Dirk Slama

AWS: Stefano Marzani, Boris Novak

Content

1. Collaboration Overview

The integration blueprint outlined in this whitepaper combines the standards and technologies from the SDV Alliance partners: COVESA, Eclipse SDV, AUTOSAR, and SOAFEE. The blueprint is orchestrated by digital.auto, utilizing technologies and products from Bosch, ETAS, and AWS.

Find out more:
dreamkit.digital.auto

SDV Alliance

The SDV Alliance is combining the best of automotive standards and open source contributions to deliver on the promise on the Software Defined Vehicle (SDV).

COVESA – the Connected Vehicle Systems Alliance – is an open and member-driven global technology alliance focused on connected vehicles and their related technologies, emphasizing collaboration and open standard approaches. Its mission is to enable its community to unlock the full potential of connected vehicles.

Eclipse Software Defined Vehicle (SDV) is a Working Group within the Eclipse Foundation that facilitates open-source development of automotive software. The aim is to provide a forum for individuals and organizations to build and promote open source solutions for worldwide automotive industry markets. Using a “code first” approach, SDV-related projects focus on building the industry’s first open source software stacks and associated tooling for the core functionality of a new class of automobiles.

AUTOSAR (AUTomotive Open System ARchitecture) is a global partnership of leading companies in the automotive and software industry to develop and establish the standardized software framework and open E/E system architecture for intelligent mobility.

SOAFEE (Scalable Open Architecture For Embedded Edge) is a Special Interest Group (SIG) aimed at bringing the cloud-native development paradigm and its ubiquitous ecosystem to the highly diverse, heterogeneous compute platforms that will power the next generation of automotive and safety-critical systems.

Integration Blueprint Partners

The blueprint is orchestrated by digital.auto, an open community for car manufacturers, suppliers, start-ups, and end-users, working together to create the next generation of SDV experiences.

The blueprint industry partners are Bosch, ETAS, and AWS. Robert Bosch GmbH, commonly known as Bosch, is a German multinational engineering and technology company headquartered in Gerlingen, Germany. The ETAS Group is a German company that designs tools for the development of embedded systems for the automotive industry and other sectors of the embedded industry. ETAS is a 100-percent subsidiary of Robert Bosch GmbH. Amazon Web Services (AWS) is the world’s most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers —including the fastest-growing startups, largest enterprises, and leading government agencies—trust AWS to power their infrastructure, become more agile, and lower costs.

2. Introduction

The SDV Alliance integration blueprint outlined in this paper addresses four major problems currently faced by the automotive industry and the software-defined vehicle (SDV) in particular.

First, the move to SDVs requires an approach towards a standardized Hardware Abstraction Layer (HAL). This HAL has to support the interfacing between the signal-oriented embedded world and the service-oriented, SDV software world (Signal-to-Service). COVESA has standardized the Vehicle Signal Specification (VSS), which enables a Signal-to-Service layer to support the required semantic hardware abstraction.

Second, SDV requires easy access to the deeply embedded systems that are used to orchestrate the multitude of sensors and actuators found on modern vehicles. AUTOSAR is a leading standard for the integration of deeply embedded systems and application software with high requirements for predictability, safety, security, and responsiveness. By combining AUTOSAR with the COVESA-enabled Signal-to-Service layer, new and powerful, SDV-based applications can be created.

Third, modern automotive software factories have to support virtualized development. This means that complex target hardware environments are replicated in the cloud for virtual testing, validation, and release. Only once the integrated vehicle software has a high degree of maturity, it's going to be deployed from the cloud to the target hardware, making developer collaboration much easier and saving money that would have previously been allocated for costly real hardware, Hardware-in-the-Loop rigs, and test vehicles.

Fourth, the automotive industry must be enabled to reduce the high costs for the development of proprietary, non-differentiating functions and enabling software frameworks. Open source – and especially the work of the Eclipse SDV working group – is providing OEMs with a means to share costs for a common software framework.

3. Integration Blueprint Architecture

The integration blueprint described in this paper is based on an automotive-grade set of hardware and software standards, as outlined in Figure 1.

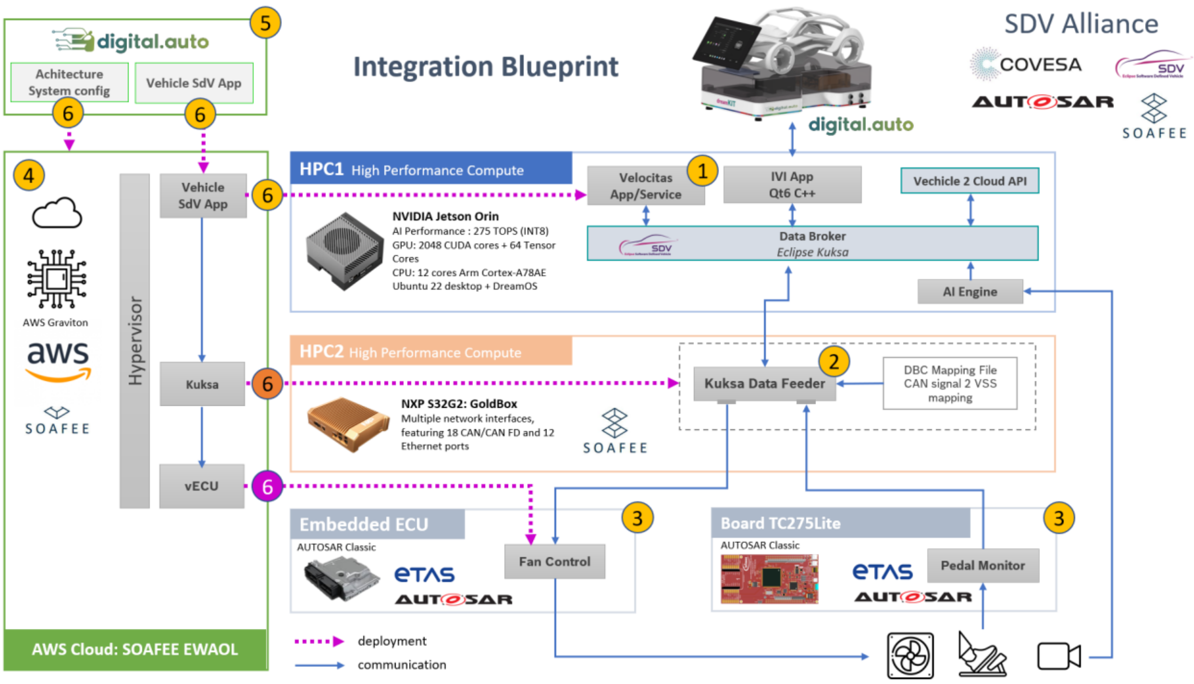

Figure 1: Integration Blueprint based on SDV Alliance standards and technologies.

A Vehicle Compute Unit (HPC1) is provided to support on-board applications running on the [Velocitas SDV runtime (1) from Eclipse](), as well as inference processing loads for on-board AI processing. In this instance of the blueprint, the VCU is an NVIDIA Jetson Orin AGX.

A Zone Computer (HPC2) for managing different ECUs in an example zone is included, based on an NXP S32G Goldbox. This will run, for example, the KUKSA message broker to integrate SDV APIs provided to application on the VCU with the actual functions on the ECUs (2).

Two ECUs are included in the system (3), to manage actuators like a fan for air conditioning, as well as a accelerator pedal. Both are integrated using AUTOSAR Classic.

AWS (4) is used to provide a virtual development environment in the cloud. At the moment, this is mainly used for the cloud-based development of SDV applications, running SOAFEE's EWAOL reference implementation as a native Amazon Machine Image. In the future, support for virtual ECUs will be added.

The digital.auto playground (5) is used for the rapid prototyping of SDV applications, as well as for the creation of the initial architecture systems configuration.

Automatic deployment in this blueprint is supported from the digital.auto playground to AWS, and from there to the target hardware (6).

4. Eclipse SDV Runtime

The blueprint described in this paper uses two key SDV-enabling technologies from the Eclipse SDV workgroup: Velocitas and KUKSA.

Eclipse Velocitas provides an end-to-end, scalable, modular, and open source development toolchain for creating containerized and non-containerized in-vehicle applications.

One of the main features of KUKSA is the ability to abstract vehicle data and interfaces to a common format based, for example, on the COVESA VSS. That way, all functions sitting on top of KUKSA can run on all enabled cars. KUKSA itself focuses on the translation of various vehicle signals into a common language based on VSS. This enables you to add your preferred onboard or offboard tech stack to new vehicle architectures more easily.

5. COVESA-Enabled Signal-to-Service Interface

The Vehicle Signal Specification (VSS) is an initiative by COVESA to define and normalize the semantics and a catalog for vehicle signals. It can be used as a standard in automotive applications to communicate information about the vehicle, which is semantically well-defined. It focuses on vehicle signals for sensors and actuators with the raw data communicated over vehicle buses and communication networks.

This means that COVESA VSS effectively provides the foundation for the creation of a hardware abstract layer (HAL), as the foundation for interoperability.

This is important for two reasons:

1. Easy re-use of VSS signals in different applications, e.g., both an infotainment application and a comfort application can use VSS signals from the vehicle body.
2. Smooth flow of code through the value stream, e.g., by running VSS clients against simulated functions first, and only later against signals from real vehicle hardware.

6. ECUs with AUTOSAR Classic and ADAPTIVE

In the blueprint described in this paper, the integration between the SDV world and the embedded world is done on the hardware as well as on the software level. On the hardware level, a dedicated zone computer is introduced as a physical link between the central computer and the ECUs in the given zone. On the software level, VSS signals are passed from the SDV application on the central computer to the zone computer via the KUKSA data broker. On the zone computer, the VSS signals are mapped to AUTOSAR signals. The middleware on the zone computer is also responsible for identifying the matching ECU for a given signal.

7. Virtualized Development with SOAFEE

In large, distributed teams working across geographical and organizational boundaries, it can be challenging to give developers easy access to the required test systems. Building up a full hardware stack to thoroughly test software-defined vehicle applications can be a challenge, especially if it requires a full test vehicle. Hardware-in-the-Loop (HiL) can often not be easily accessed remotely. HiL is usually also not ideal for remote collaboration and is also very difficult to clone for additional test beds.

The EWAOL reference implementation developed by SOAFEE aim to make the creation of virtual development environments straightforward by focusing on establishing standards that ought to help ensure environment parity between the cloud development environment and the actual target hardware (e.g., in the vehicle). This way, the native-cloud environment, and the vehicle computer can use the same binary app – meaning there will be no need to recompile the application for the vehicle computer.

This makes it much easier to set up and evaluate different kinds of test architectures. In addition, cloning and modifying existing test setups also becomes much easier. This makes the management of complex test scenarios and collaboration between core developers, integration architects, and test engineers much easier.

8. Prototyping with digital.auto

The digital.auto playground used in this blueprint makes it easy to rapidly build and test SDV applications against standardized vehicle APIs, (e.g., COVESA VSS). The integrated UX widgets make it very easy to create prototypes with rich and powerful visualizations. This is important to get early feedback from internal stakeholders or even external customers.

Native support for the automatic creation of Eclipse Velocitas applications makes it easy to migrate prototypes to the real development environment (e.g., starting with virtualized development in the cloud, and then deploying to the real test hardware).

Various integrations with market-leading vehicle simulation systems such as from Dassault Systèmes and ANSYS provide an easy on-ramp to SDV development and UX evaluation.

9. Deployments

Effective management of deployments in such complex distributed environments is essential. Support for partial as well as full deployments is important. Deployment targets are containerized, making sure that every application service has its own, fully isolated runtime environment with its own configuration and no dependencies on libraries, etc. Another critical factor is the order of deployment (e.g., which part should deploy and finish first).

In this version of the blueprint, the focus of deployment is on the containerized SDV applications. In future releases, fully virtualized ECUs (vECU) with automatic deployment from vECU to ECU are planned to be added.

10. Conclusions & Benefits

The blueprint outlined in this paper provides several benefits:

- This paper presents a joint industry effort enabled by the SDV Alliance, paving the way for improved interoperability and software development efficiency in the automotive industry.
- COVESA VSS provides the foundation for the creation of a hardware abstract layer (HAL), as the foundation for interoperability.
- This new level of standardization enables plug-and-play integration, e.g., between AUTOSAR and SDV applications.
- In addition, virtualization enabled by SOAFEE helps development teams to save time because they can quickly build up test scenarios for new applications in a virtualized yet realistic environment.
- Collaboration in distributed teams is becoming much more efficient because multiple vehicle applications can be developed and tested together in a virtualized environment.
- Migration from the virtual test environment to the target environment is frictionless, due to binary parity, enabling seamless deployment from Software- and Hardware-in-the-Loop to the target vehicle without the need to recompile source code, and no need for cross-compile toolchains.
- Adding open source to automotive software development via Eclipse SDV will enable OEMs to benefit from an open community and reduce their costs for the development and maintenance of non-differentiating features and software frameworks.

The industry collaboration by the members of the SDV Alliance has the potential to dramatically shorten development cycles of automotive software, improve cross-domain collaboration, and enable a whole slew of new vehicle experiences – thus providing huge benefits for both OEMs and their suppliers, as well as end-customers.