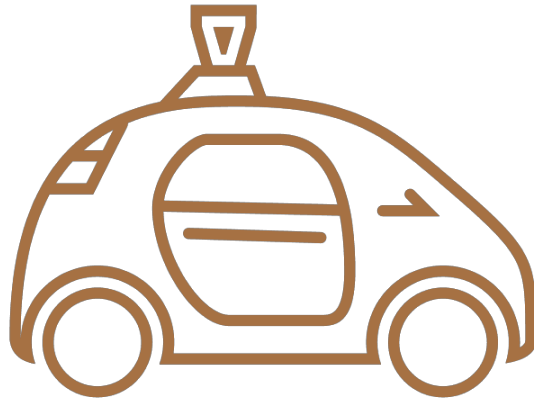


R4.1 Existing Open-Source V&V Frameworks

SafeAV · Deliverable R4.1 · CC BY-NC



SafeAV

SafeAV Project — WP4 / Task 4.1

Deliverable Type: Report, public

Deliverable No.: R4.1

Related KPI: 4.1

Coordinator: ITT Group, **Contributors:** TalTech, RTU, SUT, CTU, PRO

Authors: Mohsen Malayjerdi (TalTech), Raivo Sell (ITT), Tomasz Siwy (PRO), Piotr Czekalski (SUT), Roman Czyba (SUT), Lukáš Bertl (CTU), David Zahrádka (CTU), Agris Nikitenko (RTU), Karlis Berkolds (RTU)

External evaluators: Ehsan Malayjerdi, Volvo Autonomous Solutions (Sweden, Europe); Mustafa İlhan Akbaş, Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University (FL, USA)

Date: February 2025

Licence: CC BY-NC

Project title: SafeAV - Harmonizations of Autonomous Vehicle Safety Validation and Verification for Higher Education



Co-funded by the European Union

Table of Contents

Contents

1. [Executive Summary](#)
 2. [Purpose and Context](#)
 3. [Methodology](#)
 4. [Background and relevant sources](#)
 5. [Framework Landscape](#)
 6. [Educational Applicability](#)
 7. [Use-case requirements](#)
 8. [Conclusions and Decisions](#)
 9. [Key Findings and Recommendations](#)
-

Executive Summary

This report presents an overview and comparative analysis of open-source verification and validation (V&V) frameworks for autonomous vehicles (AVs), including full-scale cars, mobile robots, 1/10 model cars, and unmanned aerial vehicles (UAVs, i.e., drones), to identify a suitable platform for practical, higher-education use.

The study was coordinated by ITT Group and supported by Prodron, with contributions from all higher-education partners (TalTech, RTU, SUT, CTU).

Key results

- Comprehensive survey of open-source AV and UAV V&V tools.
- Framework selection and evaluation for each SafeAV use case.
- Foundation for use case developments and Task 4.2 - Adaptation of framework and educational integration.

Main recommendation: Autoware.Universe, with AWSIM, CARLA, SUMO, and complementary ROS-based tools, provides the best foundation for SafeAV's educational V&V environment.

Purpose and Context

The purpose of Task 4.1 is to:

1. Identify and analyse existing open-source V&V frameworks for AVs.
2. Evaluate their suitability for higher-education implementation.
3. Select a framework to serve as the base for SafeAV's learning and validation ecosystem.

The activity aligns with Erasmus+ KA220-HED objectives: supporting cooperation between HEIs and SMEs, promoting open educational resources, and strengthening competencies in AI-driven autonomous system safety.

Methodology

- **Desk research:** review of current scientific and industrial V&V frameworks, including EU projects, scientific publications, and software solutions.
 - **Partner contributions:** each HE partner analysed frameworks relevant to its domain (AV, UAV, mobile robot, F1/10 race car).
 - **Evaluation matrix:** all frameworks were assessed by openness, documentation, integration potential, scenario realism, educational usability, and hardware requirements.
-

Background and relevant sources

This chapter provides an overview of the key projects, research initiatives, and reference materials that form the foundation for the SafeAV framework study. It highlights both internal consortium activities and external European initiatives that contribute to the development of open, research-oriented approaches to verification and validation in autonomous systems (AS).

Projects inside the consortium indirectly related to SafeAV

- Horizon Europe, TalTech: XTRUST / Security / Safeguarding 6G-connected autonomous vehicles.
- Horizon Europe, TalTech: PLIADES / DataSpace, and collecting the operational AV data for different purposes, including V&V.
- Project-Based Learning - projects related to UAV carried out with students from SUT:
 - [Unmanned Aerial Platform Dedicated to Diagnostic Tasks](#)
 - [Hardware-software Module for Vaccine Distribution Using Unmanned Multi-rotor Platform](#)
 - [Unmanned Aerial System for Monitoring State Borders Using Thermal Imaging and Artificial Intelligence Algorithms](#)

EU initiatives

- [Strategic Dialogue on the Future of the European Automotive Industry](#), [Commission boosts European automotive industry's global competitiveness](#)

- The [AI Act](#) (Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence)
- European Union Aviation Safety Agency, Drones regulations: [Drones & Air Mobility | EASA](#)

Relevant resources

- Pioneering AI Safety in Automated Mobility [Home](#)
- Autonomy and Verification Network, [Home](#)
- Verifiability Node, [Home](#)
- Awesome Formal Verification, [GitHub - ElNiak/awesome-formal-verification: Welcome to the ultimate list of resources for formal verification techniques and tools. This repository aims to provide an organized collection of high-quality resources to help professionals, researchers, and enthusiasts stay updated and advance their knowledge in the field.](#)
- Using ROS and Gazebo to Safely Validate and Verify Autonomous Systems, Joshua P. Hecker, [roscon.ros.org](#)
- PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration [github.com](#)
- [Autoware.Universe Official Documentation](#)
- [AWSIM \(Autoware Simulator\) by Tier IV Official Documentation](#)
- [CARLA Simulator Official Documentation](#)
- [SUMO Official Documentation](#)
- [CommonRoad Official Documentation](#)
- ArduPilot [Welcome to the ArduPilot Development Site — Dev documentation](#)
- QGroundControl [QGroundControl Guide \(Daily Builds\) | QGC Guide](#)
- AirSim official documentation [Welcome to AirSim — airsim documentation](#)

Scientific Papers

- Rajabli, N., Flammini, F., Nardone, R., & Vittorini, V. (2021). **Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review.** *IEEE Access*, 9, 4797–4819. <https://doi.org/10.1109/ACCESS.2020.3048047>
- Alirezaei, M., Forrai, A., Singh, T., et al. (2024). **Tool Chain for the Virtual Validation of Autonomous Vehicles.** *ATZ Worldw*, 126, 16–21. <https://doi.org/10.1007/s38311-024-1948-x>
- Araujo, H., Mousavi, M. R., & Varshosaz, M. (2023). **Testing, Validation, and Verification of Robotic and Autonomous Systems: A Systematic Review.** *ACM Transactions on Software Engineering and Methodology*, 32(2), 1–61. <https://doi.org/10.1145/3542945>
- Razdan, R., Akbaş, M. İ., Sell, R., Bellone, M., Menase, M., & Malayjerdi, M. (2023). **PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration.** *IEEE Access*, 11, 28343–28354. <https://doi.org/10.1109/ACCESS.2023.3258681>
- Akbas, M. I. (2021). **Testing and Validation Framework for Autonomous Aerial Vehicles.** *Journal of Aviation/Aerospace Education & Research*, 30(1). <https://doi.org/10.15394/jaaer.2021.1849>
- Giovagnola, J., Moro Megías, J. B., Fernández, M. M., Cuéllar, M. P., & Morales Santos, D. P. (2023). **AirLoop: A Simulation Framework for Testing of UAV Services.** *IEEE Access*, 11, 23309–23325. <https://doi.org/10.1109/ACCESS.2023.3253788>
- Xie, D., Hu, R., Wang, C., Zhu, C., Xu, H., & Li, Q. (2023). **A Simulation Framework of Unmanned Aerial Vehicles Route Planning Design and Validation for Landslide Monitoring.** *Remote Sensing*, 15(24), 5758. <https://doi.org/10.3390/rs15245758>
- Vierhauser, M., Meixner, K., & Biffl, S. (2024). **Scenario-Based Field Testing of Drone Missions.** *arXiv preprint arXiv:2407.08359*. <https://arxiv.org/abs/2407.08359>. [arxiv.org+1](https://arxiv.org/abs/2407.08359)
- Rezaei Khavas, Z., Meriaux, E., Majdi, A., & Robinette, P. (2023). **Validating Drone Trust Testing in Navigation Deviation Cases in Simulation.** *arXiv preprint arXiv:2310.04794*. <https://arxiv.org/abs/2310.04794>. [arxiv.org+1](https://arxiv.org/abs/2310.04794)

Framework Landscape

V&V frameworks for AVs include various tools and simulators that help test AV software in virtual environments. The Framework Landscape section presents the main open-source ecosystems used for autonomous vehicle verification and validation, with a focus on Autoware.Universe and the Robot Operating System (ROS/ROS2). While Autoware provides a complete, high-level stack for autonomous driving functions, ROS offers a modular and flexible foundation for developing, testing, and integrating individual robotic components. Together, these frameworks form the technological backbone of SafeAV's educational and research environments, enabling scalable experimentation from simulation-based validation to real-world deployment.

Robot Operating System-based tools

The ROS is a flexible and modular framework for developing robotic software. Unlike Autoware.Universe—which provides a complete autonomous driving stack—ROS itself acts as a middleware and development environment that enables communication between distributed components of robotic systems. It provides standardized message passing, node orchestration, hardware abstraction, and package management, making it an ideal foundation for research and prototyping. ROS is widely used in academia and industry because of its open-source nature, active community, and vast ecosystem of reusable packages. It supports multiple languages (primarily C++ and Python) and offers extensive integration with simulation environments such as Gazebo, Webots, and CARLA. While ROS is a lower-level framework compared to Autoware, it offers greater versatility. It does not impose a specific architecture or workflow, allowing developers to assemble custom robotic systems—ranging from mobile robots and drones to full-scale autonomous vehicles. This flexibility makes ROS indispensable for educational use, where experimentation and modular learning are key.

ROS2 — evolution and key improvements

ROS2 represents the next generation of the ROS ecosystem, designed to address limitations of ROS1 and meet the growing needs of industrial and safety-critical robotics.

Key updates and improvements include:

- **DDS-based communication (Data Distribution Service):** replaces ROS1’s custom message transport, enabling real-time, decentralized, and reliable communication.
- **Multi-platform and real-time support:** ROS2 runs on Linux, Windows, and macOS, and supports real-time execution — a major step toward safety validation and embedded robotics.
- **Enhanced security and scalability:** introduces native security layers (SROS2) with authentication, encryption, and access control.
- **Improved package management and lifecycle control:** allows finer control over node startup/shutdown, lifecycle states, and deterministic behavior.
- **Better modularity and multi-robot support:** provides built-in tools for distributed systems, simulation synchronization, and collaborative robotics.

These advancements make ROS2 the preferred choice for modern autonomous systems — including those developed in the SafeAV framework. It bridges the gap between research-grade robotics and industry-ready deployment, aligning with emerging standards like ISO 26262 and the EU AI Act.

Simulators

ROS simulators provide virtual testing environments where robotic systems can be developed, validated, and visualized without physical hardware. They replicate real-world dynamics, sensors, and interactions, enabling safe and repeatable experiments for algorithm development and verification. These tools are fully compatible with the ROS2 ecosystem, supporting topics, services, and actions natively. In the SafeAV context, ROS-based simulators serve as the foundation for hands-on education and validation, allowing students and researchers to test control, perception, and planning algorithms in scalable and realistic scenarios — from small-scale mobile robots to full autonomous vehicles.

Tool / Framework	Type / Purpose	Key Features	Educational Use & Advantages	Typical Integration / Use Case	Source code	Resource description/links
Gazebo / Ignition	Physics-based simulator	Realistic dynamics, sensor simulation, 3D environments	Widely used in academia; suitable for teaching motion, sensing, and control	Robot navigation and autonomous driving co-simulation	Gazebo	gazebosim.org
Webots	Educational simulator	User-friendly UI, integrated ROS2 bridge	Excellent for introductory robotics courses and multi-robot experiments	Entry-level simulation for student projects and labs	GitHub - cyberbotics/webots_ros2: Webots ROS 2 packages	Cyberbotics: Robotics simulation with Webots
BeamNG.tech / BeamNG.research	Vehicle dynamics simulator	High-fidelity physics, HITL support, Unreal integration	Demonstrates realistic vehicle responses and safety scenarios	Testing motion planning, crashes, or physical stress validation	GitHub - BeamNG/beamng-ros2-integration	BeamNG.tech

Verification and Validation Tools

Verification and Validation (V&V) tools in the ROS ecosystem provide essential capabilities for ensuring the correctness, reliability, and safety of robotic and autonomous systems. These tools complement simulators by focusing not on the environment itself, but on the behavior and performance of the software components operating within it. V&V in ROS-based systems typically includes unit and integration testing, runtime monitoring, formal specification checking, and automated fault detection.

In the context of the SafeAV project, these tools are instrumental for teaching structured testing practices and safety assurance methodologies. They allow students and researchers to evaluate whether a system behaves according to design specifications, complies with safety requirements, and remains robust under unexpected conditions. Combined with simulation environments, ROS V&V tools provide a complete experimental workflow — from design and testing to continuous integration and validation.

Tool / Framework	Type / Purpose	Key Features	Educational Use & Advantages	Typical Integration / Use Case	Source code	Resource Description / Links
launch_testing	Testing tool for ROS2	Integration and unit test framework	Introduces software testing and CI/CD concepts	Used to verify ROS nodes and communication pipelines	launch/launch_testing at rolling - ros2/launch	Writing Basic Integration Tests with launch_testing — ROS 2 Documentation: Jazzy documentation
ROSMonitoring	Runtime verification	Monitors and verifies message exchanges	Teaches runtime verification and safety monitoring	Applied in real-time data validation for AVs and robots	GitHub - autonomy-and-verification-uol/ROSMonitoring at ros2	ROSMonitoring 2.0: Extending ROS Runtime Verification to Services...
TeSSLa	Formal verification	Temporal stream logic for cyber-physical systems	Demonstrates formal specification and timing validation	Used for research-oriented courses on safety and CPS analysis	git.tessla.io	https://www.tessla.io/ https://doi.org/10.1007/978-3-031-47963-2_23
RoboFuzz	Fuzz testing	Automated generation of random test inputs	Helps understand robustness testing and failure detection	Testing node stability and message integrity under noise	github.com	RoboFuzz: fuzzing robotic systems over robot operating system (ROS) for finding correctness bugs Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering
AsFault	Scenario/test generator	Automatic test case generation for self-driving cars	Connects algorithmic scenario creation with simulation practice	BeamNG-based testing and regression analysis	github.com	doi.org
MOBATSim	MATLAB/Simulink-based simulator	Model-based autonomous traffic simulation	Combines control theory and simulation; easy for engineering students	Suitable for system-level control validation labs	GitHub - MOBATSIm/MOBATSIm: MOBATSIm (Model-based Autonomous Traffic Simulation Framework)	doi.org
Robot Framework	Automation and test framework	Keyword-driven, supports ROS2 extensions	Simplifies test automation; cross-domain tool for robotics and RPA	Used for test scripting in ROS2 systems or integration tests	GitHub - robotframework/robotframework: Generic automation framework for acceptance testing and RPA	Robot Framework

Evaluation Summary

Tool / Framework	Open-source Licence	Autware / ROS2 Integration	Scenario Support	Educational Readiness	Hardware Demand	Community Activity
ROS2 Core	□□	□□	□	□□	□	□□
Gazebo / Ignition	□□	□□	□	□□	□	□□
Webots	□□	□	□	□□	□	□
BeamNG.tech / research	□	□	□□	□	□	□
launch_testing	□□	□□	□	□	□	□
ROSMonitoring	□□	□□	□	□	□	□
TeSSLa	□□	□	□	□	□	□
RoboFuzz	□□	□	□	□	□	□
AsFault	□□	□	□□	□	□	□
MOBATSim	□	□	□	□	□	□
Robot Framework	□□	□	□	□□	□	□

□□ = Excellent □ = Good □ = Moderate □ = High barrier

Autware autonomous driving stack connected tools

Autware.Universe, as one of the main core autonomous software stacks, categorizes these frameworks into: **Ad Hoc**

Simulation - Informal, quick feedback or subsystem testing, and **Scenario-Based Simulation** - Systematic testing with defined scenarios. This chapter summarizes open-source V&V frameworks aligned with Autoware.Universe's simulation structure, highlighting key features and recommended usage.

Ad Hoc Simulation Frameworks

Tool / Framework	Type / Purpose	Key Features	Educational Use & Advantages	Typical Integration / Use Case	Source code	Resource Description / Links
Autoware Planning Simulation	Planning and control testing (low fidelity)	Predefined HD maps, vehicle poses, deterministic runs; no sensor simulation	Ideal for teaching core control and planning logic without heavy computation	Unit testing of planning algorithms; microscopic and continuous traffic simulation	github.com	Planning simulation - Autoware Documentation
Rosbag Replay Simulation	Replay of recorded real-world sensor data	High-fidelity playback of LiDAR, radar, and camera data; limited scenario flexibility	Enables comparison between real-world and simulated outputs	Regression testing of perception and planning modules using real logs	github.com	Rosbag replay simulation - Autoware Documentation
CARLA Simulator	Full-stack, high-fidelity simulator	Realistic 3D environments, configurable sensors, ROS bridge, active community	Industry-standard AV simulator for education and research	End-to-end AV testing, ML training, scenario-based evaluation	github.com	CARLA
AWSIM (Autoware Simulator)	Digital twin simulation	Unity-based environments, full integration with Autoware.Universe, realistic sensor simulation	Teaching full-stack validation and synthetic data generation	Digital twin modeling of AV environments; safety case demonstration	github.com	AWSIM simulator - Autoware Documentation
SUMO	Urban traffic and mobility simulation	Large-scale traffic networks, efficient co-simulation, lightweight	Helps explain traffic flow modeling and multi-agent interaction	Background traffic modeling; coupled with CARLA or AWSIM for hybrid tests	github.com	SUMO Documentation
Scenic	Data generation for autonomous intelligent cyber-physical systems	Scenario modeling language, domain-specific syntax, probabilistic programming model, fully integration with CARLA	Generating Data for Training and Testing, Debugging and Verification, Interfacing with Simulators.	Scenario generation core for CARLA simulator.	Scenic Git	Scenic Website and Doc

Framework	Fidelity	Scenario Flexibility	Autoware Integration	Primary Use-case
Autoware Planning Sim	Low	Low	Built-in	Algorithm testing
Rosbag Replay	High	Low (recorded)	Built-in	Real-world regression
CARLA	High	High	ROS bridge	Full-stack simulation, ML training
AWSIM	High	High	Full	Digital twin simulation
SUMO	Medium	High	Co-simulation	Traffic scenario simulation
Scenic	Low	High	Bridge	Scenario modeling language

Scenario-Based Simulation Frameworks

Tool / Framework	Type / Purpose	Key Features	Educational Use & Advantages	Typical Integration / Use Case	Source code	Resource Description / Links
Scenario Simulator v2 (SS2)	Scenario-based simulator for Autoware	OpenSCENARIO 1.x support, flexible actor control, weather and traffic modeling, ROS2 bridge	Enables students to create repeatable and parameterized AV scenarios	Validation of perception and decision-making using standard scenario formats	github.com	Scenario testing framework for Autoware
CommonRoad	Motion planning and verification framework	OpenSCENARIO interface, benchmark datasets, vehicle models, trajectory validation tools	Ideal for algorithmic teaching in motion planning and trajectory optimization	Benchmarking of motion planners under diverse traffic scenarios	CommonRoad	CommonRoad
PolyVerif	Formal verification tool for scenario safety	Hybrid simulation + formal verification, automated reachability analysis	Teaches verification concepts for safety-critical AV functions	Analyzing correctness of planning algorithms and verifying system limits	github.com	Autonomous Vehicle Verification Consortium
ScenarioXP (X-in-the-loop Platform)	Extended simulation platform for scenario generation and HIL integration	Scenario randomization, co-simulation APIs, sensor and actuator feedback loop integration	Bridges simulation and physical testing; ideal for student labs	Used for hardware-in-loop experiments and reproducible scenario testing	github.com	scenarioxp
esmini (Essential SCENARIO Player)	Lightweight OpenSCENARIO runtime	Open-source C++ library, playback of OpenDRIVE and OpenSCENARIO files, integration with external simulators	Good for teaching simulation standards without high resource demands	Scenario playback, interface testing, and cross-platform integration	github.com	esmini user guide

Tool / Framework	Type / Purpose	Key Features	Educational Use & Advantages	Typical Integration / Use Case	Source code	Resource Description / Links
Scenic	Scenario generator	Compose complex scenarios from simpler ones, and, with the supporting VeriAI toolkit, seamlessly specify and verify AI-CPS. built-in support for a variety of simulators.	good for generating probability distributions and understanding the edge cases by students	data generation(Scenarios) for autonomous intelligent cyber-physical systems	Scenic Git	Scenic

Framework	Scenario Generation	Autoware Integration	Primary Use-case
Scenario Simulator v2	Manual	Full	Scenario logic validation
CommonRoad	Extensive library	Partial	Planning benchmarking
PolyVerif	Automated (Scenic)	Autoware.Auto only	Randomized scenario testing
ScenarioXP	Automated	Indirect	Scenario parameter exploration
Scenic	Automated	Through a bridge	Scenario modeling language

Evaluation Summary

Criterion	AWSIM	CARLA	SUMO	Scenario Simulator v2	CommonRoad	ROS Tools	Scenic
Open-source licence	□	□	□	□	□	□	□
Autoware integration	□□	□	□	□	□	□	□
Scenario support	□	□□	□	□□	□□	□	□□
Educational readiness	□□	□	□	□	□	□	□
Hardware demand	□	□	□	□	□	□	□□
Community activity	□□	□□	□□	□	□	□	□

□□ = Excellent □ = Good □ = Moderate □ = High barrier

Recommendations for Autoware.Universe

Autoware.Universe users should prioritize the following open-source frameworks:

- **AWSIM** for high-fidelity digital twin simulations.
- **CARLA** for general full-stack AV testing and active community support.
- **Scenario Simulator v2** for efficient and rapid scenario-based decision-making tests.
- **Rosbag Replay** for validating scenarios using real-world sensor data.
- **SUMO** integrated with CARLA or AWSIM for a realistic traffic scenario context.
- **CommonRoad** for benchmarking and systematic motion planning evaluations.
- **Scenic** integrated with CARLA for modeling variations of dynamic and stochastic multi-agent interactions between Autoware and traffic participants such as pedestrians, cyclists, etc.

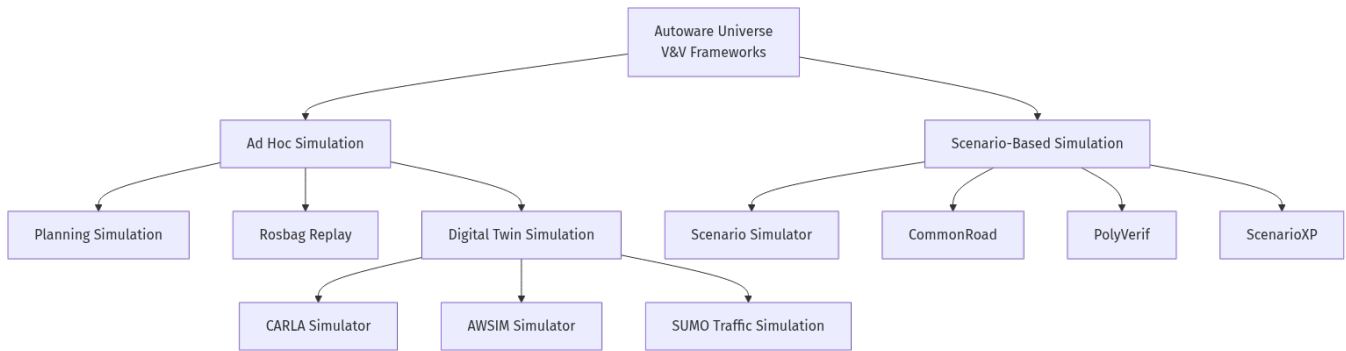
Combining these frameworks provides comprehensive V&V capabilities, enabling thorough AV software V&V aligned with industry best practices and open standards.

While many existing frameworks (e.g., CommonRoad, Scenario Simulator v2) primarily focus on testing planning and decision-making under structured scenarios, perception modules require higher-fidelity simulation. These include realistic sensor modeling, synthetic data generation, and benchmarking against ground-truth annotations.

- Autoware.Universe itself should be explicitly considered as the software under test, with the frameworks acting as test environments.
- For perception validation, CARLA and AWSIM provide realistic LiDAR, radar, and camera models, enabling end-to-end testing of detection and tracking.
- Tools such as Rosbag Replay allow regression testing with real-world sensor logs, which is crucial for evaluating perception algorithms in authentic conditions.

Together, this combination ensures that both planning and perception modules in Autoware.Universe are systematically validated across abstraction levels.

Visualization of Autoware V&V Framework Categories



Educational Applicability

Overall, ROS-based tools complement higher-level Autoware frameworks by providing fine-grained control, modularity, and educational accessibility. While Autoware offers an integrated stack for end-to-end autonomous driving validation, the ROS ecosystem enables customized experimentation and testing at the node, subsystem, and middleware levels.

This combination allows educators and researchers to:

- Teach both system-level integration and component-level verification;
- Demonstrate runtime monitoring, formal verification, and fuzz testing techniques;
- Build flexible hands-on exercises adaptable to varying computational capacities;
- Support continuous testing and CI/CD methodologies relevant to modern robotics engineering.

Use-case requirements

This chapter defines the minimum requirements for the use cases that will be developed, validated, and comparatively assessed within the SafeAV framework (AV shuttle, F1TENTH, mobile robot, UAV). The goal is to align learning outcomes with technical, safety, and regulatory constraints, and to ensure smooth integration with the selected toolchain (e.g., Autoware/ROS2, simulators, and V&V tools). Requirements cover system boundaries and assumptions, environment and scenario descriptions, data flows, performance and safety targets, acceptance criteria, and end-to-end traceability to course outcomes and WP-level KPIs. We explicitly emphasize compliance with relevant standards and regulations (e.g., UNECE; EASA, where applicable), educational reusability (SITL/HITL), and reproducibility: each use case must ship with standardized scenarios, test scripts, and evaluation report templates. Use cases are selected to cover a wide range of AV, both in the ground and air domains.

Use Case #1 AV Shuttle

The **TalTech iseAuto AV shuttle** is Estonia's first self-driving vehicle developed as an academic–industry collaboration led by Tallinn University of Technology. TalTech iseAuto operates as a fully electric vehicle with a top speed of approximately 25 km/h and a capacity of up to eight passengers. It can run for around eight hours on a single charge, making it well-suited for short urban routes and campus loops. The shuttle is equipped with a comprehensive perception system that includes three LiDAR sensors and five cameras, providing 360-degree environmental awareness. Navigation is based on pre-mapped routes, while a remote control room enables teleoperation and system monitoring when necessary. Within TalTech, iseAuto serves as a research and educational platform that bridges theoretical learning and real-world experimentation in autonomous driving. The shuttle integrates with the Autoware open-source software stack for perception, planning, and control, and it supports a digital twin simulation environment that allows testing of algorithms in virtual conditions before deploying them on the physical vehicle. This approach has made iseAuto an essential testbed for validating autonomous vehicle safety, sensor fusion, and human–machine interaction.



The focus on developing a simulation-based use case that supports education, prototyping, and pre-deployment testing. The requirements reflect both academic and technical needs, guiding the selection of suitable V&V tools and simulation environments.

- **Multi-level simulation:** The V&V setup must support both low-fidelity and high-fidelity 3D simulation tools, enabling validation across different abstraction levels.
- **Scenario testing:** The environment should support OpenSCENARIO-based scenario definitions to allow testing of complex multi-agent interactions, traffic behaviors, and reproducible validation sequences.
- **Autoware compatibility:** All V&V tools must integrate smoothly with Autoware.Universe, allowing validation of core modules such as localization, planning, and control within the same architecture.
- **Evaluation metrics:** Tools must enable automated analysis of safety metrics such as collisions, mission completion, traffic rule violations, and behavioral KPIs, suitable for both assessment and comparison.
- **Containerized deployment:** All software should support Docker-based environments for consistent, platform-independent deployment across student and research systems.
- **Educational accessibility:** Tools must be open-source, Linux-compatible, and well-documented, with a gentle learning curve suitable for university-level instruction in robotics and autonomy.
- **Iterative development:** The system should allow for quick modifications and testing of nodes or logic, supporting hands-on experimentation and frequent updates in a classroom setting.
- **Sustainability:** Tools and content must be maintainable post-project, avoiding dependencies on commercial services or non-portable infrastructure to ensure long-term use.

The AV Shuttle use case requires a flexible and scalable V&V setup that supports both low- and high-fidelity simulations, OpenSCENARIO-based scenario testing, and full integration with Autoware.Universe stack. The environment must enable automated safety and performance evaluation, containerized deployment, and open-source accessibility suitable for higher education. It emphasizes iterative, hands-on experimentation and long-term sustainability without reliance on commercial tools

Use Case #2 F1TENTH

The F1TENTH platform is an open-source, small-scale autonomous racing car designed for research and education in autonomous systems. Built on a 1/10-scale RC chassis, it integrates sensors such as LiDAR, camera, and IMU, all running on a ROS2-based control stack. Its modular architecture allows experiments in perception, planning, and control, while remaining low-cost and portable for classroom and laboratory use. The platform is supported by an active international community, offering simulation environments, datasets, and open course materials that make it ideal for hands-on learning and benchmarking in robotics and self-driving research. In academia, it serves as a standardized benchmark for teaching autonomous driving algorithms, allowing students to bridge theory and practice through competitions, lab assignments, and project-based learning. Universities use F1TENTH to demonstrate safety validation, sensor fusion, and real-time decision-making concepts within a manageable and reproducible framework, making it an ideal entry point for higher education in



The interface requirements for the F1TENTH use case define how the simulation and control systems communicate within the ROS2 environment (ROS2 Humble, with possible but nontrivial porting to Jazzy). The platform's sensor outputs include LiDAR scans, RGB or RGB-D camera feeds, and IMU data, which can be noisy and unreliable due to the small size of the vehicle. Additional outputs, such as pose and velocity estimates, may be derived using external localization methods. Control inputs consist of speed commands—either normalized or expressed in meters per second—and steering angle values, typically represented as PWM signals that can be translated into degrees. These interfaces ensure smooth integration between the physical and simulated components, supporting real-time testing, algorithm verification, and reproducibility in educational and research contexts.

The F1TENTH use case defines simulation requirements that ensure accessible and reproducible environments for validating autonomous driving algorithms within academic and research settings.

- The simulation environment must support lightweight, scalable, and educationally accessible setups suitable for university use.
- It should utilize kinematic single-track or Ackermann steering models to balance computational efficiency with sufficient accuracy for low-speed vehicles.
- The framework must enable reproducible validation of perception, planning, and control algorithms in both 2D and 3D environments.
- ROS2 compatibility is required, with optional integration to simplified Autoware vehicle models for advanced testing.
- Simulations should run smoothly on standard laptops without the need for high-end GPUs, ensuring broad accessibility

for students.

- The overall design must support iterative experimentation, open-source deployment, and hands-on learning in line with SafeAV's educational objectives.

The F1TENTH use case focuses on providing an open, reproducible, and educational platform for validating perception, planning, and control algorithms in small-scale autonomous vehicles. Its simulation and interface requirements emphasize lightweight, ROS2-compatible environments that run efficiently on standard hardware, enabling hands-on learning, iterative testing, and scalable experimentation in autonomous systems education.

Use Case #3 Mobile Robot

The Mobile Robot (RTU) use case focuses on cooperative indoor logistics systems designed to demonstrate autonomous navigation, coordination, and task management in controlled environments. The setup consists of two mobile robot platforms, a central server for planning and task distribution, and MQTT-based communication for asynchronous message exchange. Each robot operates under a ROS2-based control architecture integrating LiDAR, camera, and deep learning-based segmentation for enhanced mapping and path planning. Within the academic context, this use case provides a practical environment for teaching multi-robot coordination, communication reliability, and safety validation, bridging theoretical concepts in robotics and AI with real-world industrial applications.



The V&V requirements for the Mobile Robot (RTU) use case focus on ensuring safe, reliable, and verifiable operation of cooperative indoor logistics robots within both simulated and physical environments. The framework is designed to validate system behavior across all levels—communication, perception, navigation, and task management—while supporting educational objectives through accessible, open-source tools. These requirements align with SafeAV's broader goal of integrating real-world industrial practices into higher-education robotics training.

- The verification and validation framework must support systematic testing across design, simulation, and real-world operation, ensuring traceability from requirements to test results.
- It should enable component-level and interface testing of ROS2 nodes, MQTT communication, sensor data pipelines, and deep learning-based segmentation modules.
- The setup must allow hardware-in-the-loop (HIL) and simulation-based validation, confirming that control, navigation, and planning functions behave safely under realistic indoor conditions.
- Runtime monitoring and safety shields should be implemented to detect failures, prevent unsafe actions, and support continuous verification in CI/CD workflows.
- The framework must assess system performance under communication losses or operator intervention, ensuring resilience and fault tolerance in distributed multi-robot environments.
- All tools and procedures should remain open-source, modular, and reproducible, allowing students to perform iterative experiments, analyze safety properties, and understand industrial V&V practices within an educational context.

The defined V&V requirements establish a comprehensive validation chain that connects design, simulation, and real-world testing. They emphasize fault tolerance, runtime monitoring, and reproducibility using open-source ROS2 and MQTT-based architectures. This ensures that students can study and experiment with advanced verification techniques while developing safe and resilient autonomous robotic systems.

Use Case #4 Drone

The Drone (SUT, PRO) use case focuses on unmanned aerial vehicle (UAV) systems that bridge aviation safety principles with autonomous mobility education. Developed through Prodrone's extensive experience in UAV training and system design, this use case explores real-world challenges such as emergency response, navigation under uncertainty, sensor fusion, and communication reliability. The drones operate using open-source frameworks like ArduPilot and QGroundControl, supporting both software-in-the-loop (SIL) and hardware-in-the-loop (HIL) validation. In the academic context, this use case provides a versatile platform for teaching autonomous flight control, safety verification, and resilience testing, allowing students to apply V&V methodologies from aerial robotics to broader autonomous vehicle domains.



The UAV use-case requirements focus on defining the essential verification and validation conditions for safe and reliable autonomous flight operations. They emphasize emergency response handling, navigation under uncertainty, sensor integration, and communication robustness, including lessons from MAVLink telemetry that can directly apply to V2X communication challenges in automotive systems. These requirements ensure realistic, reproducible, and educationally relevant validation of UAV systems.

- The framework must enable **emergency response validation**, including fault injection, communication loss recovery, and system redundancy testing.
- It should support **navigation under uncertainty**, allowing for dynamic obstacle avoidance, no-fly zone compliance, and route optimization under changing conditions.
- The simulation environment must facilitate **sensor integration testing**, including GNSS, IMU, and vision-based systems, with modeling of environmental factors such as wind or weather effects.
- **Communication reliability and latency simulation** should be included to evaluate telemetry, interference resilience, and cybersecurity aspects of UAV operations.
- The setup must support **SIL and HIL configurations**, enabling both pure software and mixed real-hardware validation of control and perception modules.
- Tools and models should support, where reasonable, a **dual-tier approach**, combining **solutions** with **open-source frameworks** for advanced experimentation.

The UAV V&V requirements ensure comprehensive testing of autonomous flight systems under realistic and variable conditions, supporting both educational and research-oriented objectives. By integrating open-source simulation, communication reliability testing, and hardware-in-the-loop validation, they provide a robust foundation for safety assurance and hands-on learning. The SafeAV study recommends a dual-tier approach for UAV simulation and validation—combining commercial off-the-shelf (COTS) systems for rapid onboarding with open-source ecosystems for advanced, research-driven experimentation. This structure enables both immediate applicability in training contexts and long-term scalability for integration

into academic courses, laboratories, and testbeds.

Conclusions and Decisions

In conclusion, the consortium has evaluated the available verification and validation frameworks based on current research, technical feasibility, and educational applicability. The resulting decisions reflect a balanced consideration of open-source maturity, interoperability, and relevance to the specific SafeAV use cases. The selected frameworks demonstrate strong community support, active ongoing development, and proven suitability for academic integration. Their open-source nature ensures transparency, adaptability, and long-term sustainability, while their functionality aligns closely with the technical and pedagogical goals defined for each use case.

Use case No	Name	Description	Selected Frameworks and Software Set	Responsible partner
#1	AV shuttle	Simulation-based V&V use case for an autonomous shuttle used in education, prototyping, and pre-deployment testing. Requirements include multi-level simulation, scenario testing with OpenSCENARIO, Autoware.Universe integration, automated safety metrics, containerized deployment, and educational accessibility. Both planning and perception modules are targeted for validation.	Autoware.Universe (software under test), AWSIM (digital twin, sensor simulation, Autoware integration), CARLA (high-fidelity perception & planning validation, synthetic data), Rosbag Replay (real-world perception regression), Scenario Simulator v2 (scenario-based planning validation), CommonRoad (planning benchmarking), SUMO (traffic co-simulation) Scenic (Scenario generator for CARLA)	TalTech
#2	F1Tenth	The F1TENTH use case demonstrates the application of open-source V&V frameworks for validating autonomous driving functions on a 1/10-scale research platform. Verification focuses on trajectory tracking, ensuring that the vehicle follows safe and stable paths without collisions or oscillations when encountering static or dynamic obstacles. In addition, the setup supports perception-level validation, including object and traffic sign detection, persistence of observations, and accuracy of estimated object velocities. Sensor data are also subject to pre-processing validation, guaranteeing their reliability for mapping, localization, and higher-level decision-making. This use case highlights how compact, reproducible, and ROS2-based V&V environments can provide students and researchers with an accessible platform for hands-on testing of autonomous vehicle algorithms and safety assurance methods.	Rosbag Replay (real-world perception), F1TENTH Gym (simple 2D simulation) ROSMonitoring (verifying at runtime) RoboFuzz (testing with noisy inputs)	CTU
#3	Mobile robot	The V&V framework for the cooperative indoor logistics robot system ensure that V&V are systematically applied across design, simulation, and real-world operation. Verification focuses on requirement traceability, component and interface testing (ROS2 nodes, MQTT messaging, sensors, DL segmentation, and planners), and formal checks for safety and deadlock freedom. Validation uses simulation, hardware-in-the-loop, and on-site trials to confirm that the system performs safely and efficiently under realistic conditions, including communication losses and operator overrides. Runtime monitoring, safety shields, and continuous testing in the CI/CD pipeline maintain ongoing assurance, ensuring reliable autonomous operation and operator control in dynamic indoor logistics environments.	Nav2 and MoveIt 2 (navigation and motion planning) Gazebo/Ignition (simulation and testing) MQTT broker Eclipse Mosquitto for message exchange OpenCV and PCL vision and LIDAR processing ROS2 testing tools	RTU

Use case No	Name	Description	Selected Frameworks and Software Set	Responsible partner
#4	Drone	Simulation-based verification and validation (V&V) for unmanned aerial vehicles (UAVs) is essential before executing real-world missions to ensure software correctness, safety, and performance. The framework must simulate the drone's plant model, environmental perception, and control system with sufficient fidelity while maintaining real-time execution capability. Both software-in-the-loop (SITL) and hardware-in-the-loop (HIL) configurations are required to validate autonomous behaviors such as navigation in complex environments, object detection, and precise landing. Modern 3D simulation engines, such as Unreal Engine used in CARLA, now enable highly realistic testing conditions—some even experimentally extended to aerial vehicles—providing a robust basis for safe, repeatable, and educational UAV validation.	AirSim (real-world perception) Gazebo (open source simulator) ArduPilot , QGroundControl (primary framework) UAV Matlab/Simulink Toolbox (real-time simulation and deployment for UAV) CARLA (Unreal Engine based simulator)	SUT

Key Findings and Recommendations

ROS-based frameworks thus form a critical part of the **SafeAV educational toolchain**, ensuring scalability from lightweight student projects to advanced V&V experiments in research and industrial contexts.

The selected frameworks fulfill higher-education requirements:

- **Accessibility:** open-source and licence-free use.
- **Ease of setup:** Docker-based deployment for classroom or remote use.
- **Pedagogical link:** supports blended learning (MOOCs + hands-on labs).
- **Interdisciplinary use:** applicable in robotics, AI, safety engineering, and mechatronics courses.

These outcomes directly contribute to WP3 educational digital content and WP2 modular curriculum design.

1. **Autoware.Universe** should be the baseline platform for SafeAV framework adaptation.
2. **AWSIM/CARLA + SUMO/Scenic** provide complementary environments for digital twin, perception, and traffic-level validation.
3. **ROS-based verification tools** enable node-level and formal validation, aligning with ISO 26262 / SOTIF principles.
4. Partner use cases ensure coverage of ground, aerial, and hybrid autonomous systems for educational demonstration.

Next Steps → T4.2 Adaptation

- Containerize selected frameworks for student deployment.
- Develop a hands-on educational guide linking WP3 digital content to WP4 V&V examples.
- Integrate simulation exercises into the SafeAV MOOC platform.
- Define data interfaces (ROS bag, OpenSCENARIO) for cross-use of materials.
- Establish KPIs for student learning outcomes (practical validation success, reproducibility, safety comprehension).

Erasmus+ Visibility and Licensing

This content was implemented under the following project: Cooperation Partnerships in higher education (KA220-HED): SafeAV: Harmonizations of Autonomous Vehicle Safety Validation and Verification for Higher Education: **2024-1-EE01-KA220-HED-000245441**

Permanent URL of this document: <https://www.ittgroup.ee/index.php/active-projects/safeav>

Erasmus+ Disclaimer

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Copyright Notice

This content was created by the SafeAV Consortium 2024–2027.

The content is copyrighted and distributed under CC BY-NC [Creative Commons Licence](#) and is free for non-commercial use.

Consortium Partners

- ITT Group, Estonia (Coordinator)

- Silesian University of Technology, Poland
- Riga Technical University, Latvia
- Tallinn University of Technology, Estonia
- Czech Technical University in Prague, Czech Republic
- Prodron sp. z o.o., Poland

