Towards the evaluation of the Arrowhead SoA in ITS

Luís Ribeiro, Tiago Costa, Ricardo Severino and Luis Lino Ferreira
INESC-TEC/ISEP, Instituto Politécnico do Porto
Porto, Portugal
{lfsro, 1201329, rar, llf}@isep.ipp.pt

Abstract— **The evolution of autonomous driving is reshaping the automotive landscape into a highly cooperative and interconnected system, where vehicles and infrastructure exchange data to improve safety, efficiency, and responsiveness. In this context, service-based architectures are becoming essential to support the modular, scalable, and dynamic nature of automotive applications such as Cooperative Perception, demanding robust mechanisms for real-time communication, service discovery, interoperability, and secure data handling. This work aims at investigating the suitability of the Arrowhead Framework—a service-oriented architecture initially designed for industrial automation—as a middleware to enable and manage services in the context of cooperative autonomous driving. By integrating Arrowhead into a multi-dimensional co-simulation framework, encompassing the simulation of realistic vehicle models, control and communications, we evaluate its effectiveness in supporting service orchestration, system integration, and interoperability in different scenarios. In parallel, we aim to demonstrate how co-simulation environments can facilitate the rapid prototyping and deployment of distributed autonomous driving services.**

Keywords-ITS; service-oriented architecture; co-simulation;

I. Introduction

The rapid advancement of autonomous driving technologies is steering the automotive industry toward a cooperative and intelligent ecosystem. Vehicles are no longer isolated systems but integral nodes in a broader and more complex Intelligent Traffic System (ITS), in which they collaborate with the roadside infrastructure, traffic control systems, and even pedestrians. One of the most promising paradigms in this domain is Cooperative Perception, which relies on the fusion of sensor data shared among various entities to enhance the awareness and safety of automated driving systems. To realize this level of cooperation, future automotive applications must be highly modular, distributed, and interoperable—qualities well-aligned with service architectures. Services allow developers to break down complex functionalities into independently deployable units, enabling scalability, fault isolation, and easier updates. Particularly in ITS, services architectures hold the potential to enable a multitude of services while reusing and repurposing the roadside infrastructure and sensors. Roadside deployed sensors such as cameras or radars can support cooperative highway merging applications, while enabling traffic emergencies detection, and helping authorities enforce traffic rules, by monitoring vehicles' speed. However, deploying services in automotive contexts introduces several challenges, including low-latency requirements, dynamic service discovery, orchestration, data standardization, and secure communications.

In response to these challenges, our work explores the use of the Arrowhead Framework, an open-source service-oriented architecture originally developed for industrial automation, as a service orchestration platform tailored for automotive use cases. We present an integration of Arrowhead into a Co- Simulation Framework, designed to simulate cooperative driving scenarios in its multi-dimensional perspective, encompassing realistic vehicle dynamics, control and communications, to validate the service interactions in a controlled virtual environment. This setup enables the development, testing, and evaluation of automotive-grade services without the high cost and complexity of real-world testing.

The contributions of this work are threefold:

- Assessment of Arrowhead's applicability to cooperative autonomous driving, focusing on its support for multiple automotive service registration, discovery, and secure communication.
- Demonstration of service-based integration for Cooperative Driving applications, leveraging the modularity and extensibility of the Arrowhead architecture.
- Development of a unified co-simulation framework, which allows seamless integration of services across Autonomous Driving and Smart City scenarios, facilitating experimentation and iterative development.

II. Related Work

A. Service-Based Vehicular Networks

Service-based architectures for Vehicular Networks promise to deliver high flexibility, modularity and scalability for designing and implementing ITS applications. This architecture allows the breakdown of specific services into smaller functionalities, which can be further developed and improved on their own, allowing for more efficient, reliable, and fault tolerant applications. Moreover, it facilitates the integration of security, enabling finer-grained access control and monitoring, and reducing the impact of attacks on specific services. Nevertheless, it is fundamental to address the specific cyber-physical requirements such as bounded latency, prevalent in many ITS applications, particularly those involving Advanced Driving Assistance Systems (ADAS) or cooperative driving.

The authors in [1] proposed a service-based architecture that integrates modular V2X services such as Cooperative Awareness with autonomous vehicle systems. The framework developed relies on the Data Distribution Service (DDS) and

Manuscript received July 14, 2025; revised July 31, 2025; accepted July 25, 2025. Published September 2, 2025.
Issue category: Special Issue on DSD/SEAA 2025 on Works in Progress (WiP) Session, Salerno, Italy, Sept. 2025
Paper category: Short
DOI: doi.org/10.64552/wipiec.v11i1.91

introduces a bridge between the V2X services and the autonomous navigation stacks of the vehicles, the Vehicle Programming Interface (VPI). It's capable of dynamic handling of emergency notifications and traffic light states, and its effectiveness was validated by the use of a hybrid experimental setup that combines real world simulation and real-world V2X infrastructure using ITS-G5 communications. This being said, the validation relies on hybrid small-scale tests. Large-scale deployments or edge cases with dense traffic or adverse weather conditions are not addressed. Moreover, security aspects are not addressed.

Another service-based architecture was proposed in [2], promising to handle reliability and latency requirements of 5G and upcoming 6G networks. This solution integrates multiple layers of computing with both fog and mobile edge computing (MEC) and software-defined networking (SDN) that enables adaptive resource management and seamless communication in highly dynamic vehicular environments. Furthermore, the architecture incorporates efficient task offloading, service migration, and handover mechanisms that ensure continuity of service operation under high-speed vehicular mobility. Unlike in our work, the authors used ns-3 for network simulation, using C-V2X. Tests were done with Sumo and ns-3, using the C-V2X as the foundation technology for communications. While the work offers valuable contributes for scaled networks that have a need for the use of handover, task offloading and service migration mechanisms, it also fails to address a few security concerns, such as authentication and encryption and relies heavily on C-V2X/NR and SDN, therefore compatibility with legacy systems such as ITS-G5 is unclear.

Din et al [3] presented a multilayered framework combining services and Named Data Networking (NDN) for efficient in-network computation in autonomous vehicular systems. Their architecture is composed of physical edge servers, and cloud infrastructure to support distributed, computation-intensive tasks. In comparison with more monolithic architecture, through the use of hybrid simulation, the system was capable of enabling efficient offloading of tasks, reducing latency, and optimizing bandwidth usage.

All these works provide a meaningful foundation for the development and improvement of Service-based architectures. While we develop an architecture of our own, we intend to provide an expansible framework that offers the tools to develop these types of architectures.

B. *Arrowhead Framework*

The Arrowhead Framework is a Service-Oriented Architecture (SOA) designed to enable IoT interoperability in-between almost any IoT elements [4]. The framework provides a well-defined structure for managing loosely coupled, event-driven automation systems through core services such as Service Registry, Authorization, and Orchestration [5]. These services enable dynamic discovery, secure access control, and more efficient coordination of distributed systems, making them suitable for industrial automation, IoT, and smart infrastructures. The framework also supports local cloud deployments and inter-cloud collaboration for broader scenarios, which is important to ensure the low-latency and scalability requirements needed for autonomous driving applications. Arrowhead has been applied to a few related scenarios with success, however, none with the stringent requirements imposed by ADAS services. For instance, Joniken et al. [6] demonstrated the capabilities of the use of the Arrowhead Framework for smart city service integration, implementing and connecting two distinct urban infrastructure systems (a street lighting system and a car engine block heating system) into a unified, collaborative automation environment. In this work each system was wrapped with an Arrowhead-compliant interface, exposing RESTful services for monitoring and control, and integrated using Arrowhead's core systems. The control system, acting as a consumer, was capable of dynamically discovering and orchestrating services based on environmental sensor data, such as temperature and luminosity. Passerone et al. [7] presented a comprehensive design methodology for secure and safety-compliant communication in autonomous vehicle systems, specifically those who rely on V2V communication. They introduced a contract-based design approach to formalize and verify system requirements and used the Arrowhead Framework to manage secure service-oriented communications. Verification of system behaviors against formalized safety assertions is done through Contract Analysis Tool (CAT) and 3D simulations in Blender, and it was finally validated through a prototype implementation involving real-time control of wheeled robot platoons. Arrowhead's token-based authorization was able to enable secure communication with latencies of up to 40ms, which addresses the trade-off between security and overhead.

III. Architecture

In this section, we describe the architecture of the Co-Simulation setup and its integration with the Arrowhead Framework. Moreover, we will delve into the case scenarios that will be developed for testing Vehicular Networks.

A. *System's Architecture*

Figure 1 displays a simplified architecture of our Co-Simulation Framework and its integration with Arrowhead. It integrates both ROS 2 and Gazebo with the Omnet++ and Arrowhead Frameworks. Omnet++ is a network simulation tool, and we use it together with the libraries from the Artery framework which allows us to simulate Vehicular Communications with protocols such as ITS-G5.

Our Framework leverages ROS2 and the underlying DDS as the foundation for the interaction between the different tools that compose our application. The Omnet++ Interface is composed by a Omnet++ transmitter that creates a publisher, whose topics are subscribed by the Omnet++ module, which then relays the information to the correct nodes within the Omnet++ simulation environment. The interface also has an Omnet++ receiver which captures the packets from the Omnet++ simulation environment and publishes them into the respective ROS topics.

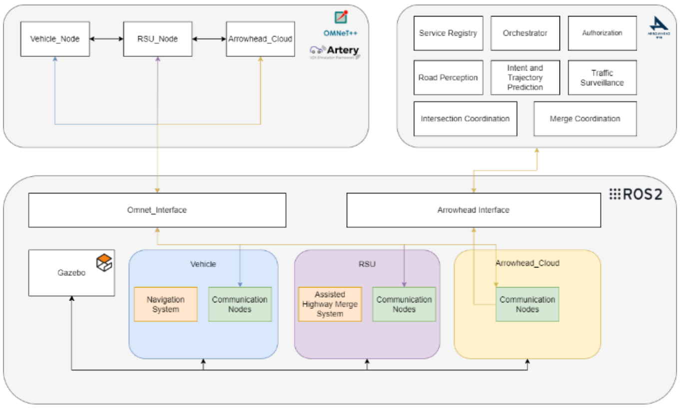

Figure 1: Architecture of the Simulation Framework.

In the context of simulation for connected vehicles, integrating real-time communications with simulation time can be quite challenging due to the Event-Driven nature of network simulation tools, while Simulation tools like Gazebo operate on real-time progression. To ensure that our Network Simulation tool, in this case Omnet++, is on par with real-time events, it's necessary to build a scheduler capable of synchronizing event times with the simulation times. To ensure both Omnet++ and Gazebo are on par with each other, we are using an event scheduler previously made for Omnet++ in [8].

The Arrowhead Interface is a module that sends and handles RESTful requests to the Arrowhead Framework from the nodes within our simulation environments. Within our network, Arrowhead allows for service discovery and authentication. As already mentioned, we intend to test the capabilities of Arrowhead, evaluating the impact services delays may have on the performance of cooperative driving tasks. In our network architecture we decompose services into smaller services capable of interoperating together in many scenarios. In this case, we have Arrowhead Core Services, Service Registry, Orchestrator and Authorization along with some services that are going to be provided by Roadside Units (RSU). The following services are deployed at Edges and discovered and orchestrated by Arrowhead's core services:

- **Road Perception –** Detects and classifies static and dynamic road agents, such as vehicles and pedestrians.
- **Intent and Trajectory Prediction –** Predicts the behavior and trajectory of nearby road agents using past motion and context information.
- **Traffic Surveillance –** Monitors the road and detects any unsafe or irregular behaviors or events, such as stalled vehicles or illegal maneuvers.
- **Merge Coordination** – Dedicated to the highway merge scenario, its function is to coordinate highway merging by calculating and suggesting safe gaps and speeds for entering vehicles.
- **Intersection Coordination –** Dedicated to the urban intersection scenario, its function is to coordinate and assign priorities to vehicles approaching and traversing intersections.

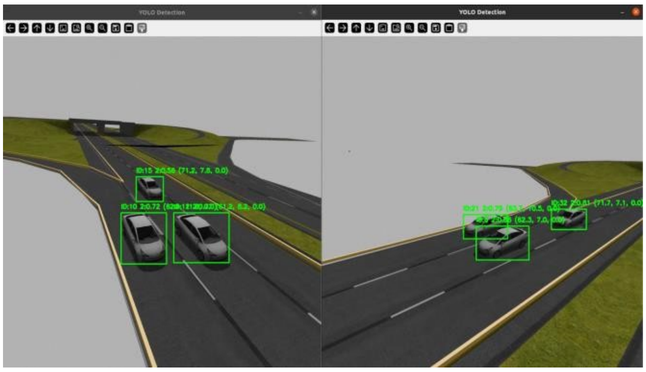
Figure 2: Vehicle Detection in Gazebo.

The Road Perception service relies on cameras that are distributed across the road positioned in a Bird's Eye View (BEV) configuration within the Gazebo simulation environment. YOLOv8 [9] was used for object detection and tracking, and trilateration was applied to fuse the different cameras' perspectives to provide more precise estimations of the positions of the road agents as shown in Figure 2.

B. Test Scenarios

To validate the proposed solution and assess the applicability of Arrowhead Framework within the context of cooperative autonomous driving, we design and implement two different test scenarios, High-Speed Highway Merge and Intersection Assistance. The interaction between vehicle and infrastructure requires the vehicle to first communicate with the RSU's requesting which services are available to use in that region. The RSU connected to an Edge forwards this request to an Arrowhead node deployed at a considerable distance, which will reply with the list of services available in that region. After receiving the list, the vehicle can finally choose what services it wants to subscribe to, through the same process, where then the Arrowhead will verify the vehicle's authenticity to consume those services.

The Highway Merge scenario serves as the most demanding test case, since it imposes lower latencies upon the performance of our services. In this setup, multiple vehicles will be deployed randomly, at speeds up to 120 km/h, simulating real world traffic. The cameras detect and track the vehicles' positions and speed, which will be used to inform incoming vehicles wanting to merge onto the highway. The Highway Merge service identifies suitable gaps and assigns them to incoming vehicles along with the recommended speeds for merging. Our primary goal is to evaluate the latency bounds and responsiveness of the Arrowhead Framework at higher speed cooperative scenarios. Through pushing the framework's capabilities, we can identify limitations and explore possible enhancements that could make Arrowhead more appropriate for automotive applications where low latency is a critical aspect.

The Urban Intersection Assistance scenario, while less sensitive to higher latencies than our previous scenario, offers a controlled environment that can be used to test communications under mixed traffic conditions and serve as a test bed for service scalability, orchestration and integration with other types of services, specifically Smart City applications. Furthermore, it provides an opportunity to explore intersections as convergence points between cooperative driving and urban infrastructure services, namely traffic monitoring, pedestrian safety, and emergency management. Figure 3 presents a more general and hierarchical view of the deployment scenarios. The vehicles, sensors, and RSU's are distributed throughout the environment. The RSUs are directly connected to the Edge nodes, providing them with sensorial data required for service execution. The Edges communicate with Arrowhead nodes that are deployed further from the urban environment usually in the form of Fog's deployed in the city where these services are provided.

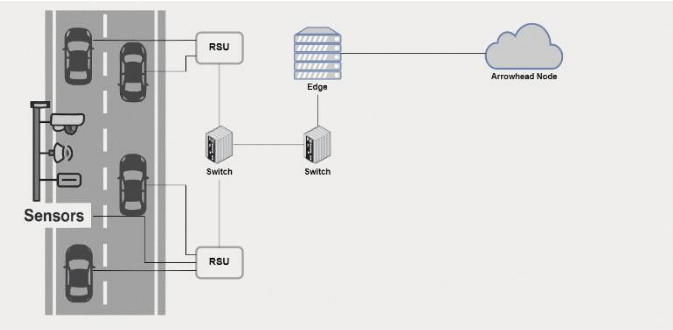

Figure 3: Hierarchical deployment Scenario.

IV. Conclusion and Future Work

This work aims at assessing the performance of the Arrowhead Framework as a middleware in enabling and managing services in the context of cooperative autonomous driving. To this end, we integrated the Arrowhead Framework over a co-simulation framework relying on ROS 2, Gazebo, and OMNeT++ and devised a set of cooperative autonomous driving scenarios. By leveraging Arrowhead's service-oriented architecture, we enable dynamic service discovery, orchestration, and secure communication between distributed vehicular and infrastructure nodes. Our initial implementation focuses on validating the feasibility and responsiveness of Arrowhead microservices in two representative scenarios: high-speed highway merging and urban intersection coordination. These scenarios allow us to examine the framework's behavior under both strict latency constraints and complex, scalable service environments. Early results suggest that while Arrowhead offers promising mechanisms for structuring and managing ITS services, its performance in real-time vehicular contexts—especially under high-speed conditions—must be thoroughly assessed. The proposed co-simulation setup provides a flexible and extensible environment to iteratively develop, test, and refine distributed services for automotive and smart city applications. It serves as a valuable platform to simulate real-world conditions encompassing multiple cyber-physical dimensions, while maintaining controlled test parameters for the intended evaluation.

Ongoing and future work will focus on profiling service delays, optimizing orchestration logic, and exploring enhancements to the Arrowhead Framework to better meet the stringent demands of vehicular networks and general QoS demanding ITS services. Ultimately, this research contributes to the broader goal of enabling safer and more efficient cooperative autonomous systems by evaluating service-oriented solutions within realistic, simulation-driven development pipelines.

Acknowledgment

This research is supported by the OPEVA project funded within the Chips Joint Undertaking (Chips JU) from the European Union's Horizon Europe Programme and the National Authorities (France, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. It is also funded by the EU Chips-JU project Arrowhead fPVN with grant agreement no. 101111977 and the national funding authorities from the countries involved. Views and opinions expressed are however those of the authors only and do not reflect those of the European Union or Chips JU.

References

[1] J. Amaral, A. Figueiredo, P. Rito and S. Sargento, "Microservice-based Approach to Integrate V2X with Autonomous Mobility," 2024 IEEE Future Networks World Forum (FNWF), Dubai, United Arab Emirates, 2024, pp. 221-225, doi: 10.1109/FNWF63303.2024.11028714.

[2] Zarie, Mira M., Abdelhamied A. Ateya, Mohammed S. Sayed, Mohammed ElAffendi, and Mohammad Mahmoud Abdellatif. 2024. "Microservice-Based Vehicular Network for Seamless and Ultra-Reliable Communications of Connected Vehicles" Future Internet 16, no. 7: 257. https://doi.org/10.3390/fi16070257

[3] Muhammad Salah Ud Din, Muhammad Atif Ur Rehman, Muhammad Imran, Byung Seo Kim, Evaluating the impact of microservice-centric computations in internet of vehicles, Journal of Systems Architecture, Volume 150, 2024, 103119, ISSN 1383-7621, https://doi.org/10.1016/j.sysarc.2024.103119.

[4] Eclipse Foundation, "The Arrowhead Framework," https://projects.eclipse.org/projects/iot.arrowhead.

[5] Varga, Pal & Blomstedt, Fredrik & Ferreira, Luis & Eliasson, Jens & Johansson, Mats & Delsing, Jerker & Martínez de Soria, Iker. (2016). Making system of systems interoperable – The core components of the arrowhead framework. Journal of Network and Computer Applications. 81. 10.1016/j.jnca.2016.08.028.

[6] J. Jokinen, T. Latvala and J. L. Martinez Lastra, "Integrating smart city services using Arrowhead framework," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 2016, pp. 5568-5573, doi: 10.1109/IECON.2016.7793708.

[7] Passerone, Roberto & Cancila, Daniela & Albano, Michele & Mouelhi, Sebti & Plósz, Sándor & Jantunen, Erkki & Ryabokon, Anna & Emine, Laarouchi & Hegedus, Csaba & Varga, Pal. (2019). A Methodology for the Design of Safety-Compliant and Secure Communication of Autonomous Vehicles. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2937453.

[8] Teper, Harun & Bayuwindra, Anggera & Riebl, Raphael & Severino, Ricardo & Chen, Jian-Jia & Chen, Kuan-Hsun. (2022). AuNa: Modularly Integrated Simulation Framework for Cooperative Autonomous Navigation. 10.48550/arXiv.2207.05544.

[9] Yaseen, Muhammad. (2024). What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. 10.48550/arXiv.2408.15857