I-VICS R4 Architecture Document

Blueprint overview/Introduction

Zhuming Zhang Hechun Zhang Hao Zhongwang

The AI Edge is an Akraino approved blueprint family and part of Akraino Edge Stack, which intends to provide an open source MEC platform combined with AI capacities at the Edge, and could be used for safety, security, and surveillance. The MEC platform, which named ote-stack, targets on shielding the heterogeneous characteristics through underlying hardware virtualization and providing an unified access for cloud edge, mobile edge and private edge. In addition, the AI Edge utilizes the cluster management and intelligent scheduling of multi-tier clusters to enable low-latency, high-reliability and cost-optimal computing support for running AI applications at the edge. At the same time, it makes device-edge-cloud collaborative computing possible.

Application scope of Intelligent Vehicle-Infrastructure Cooperation System(I-VICS): including airport and station passenger flow grooming system, urban traffic intelligent dispatching system, expressway intelligent dispatching system, operation vehicle dispatching management system, motor vehicle automatic control system, etc. The function of intelligent transportation system: it improves the efficiency of transportation, alleviates traffic congestion, improves the capacity of road network, reduces traffic accidents, reduces energy consumption and environmental pollution through the harmonious and close cooperation of people, vehicles and roads.

Use Case

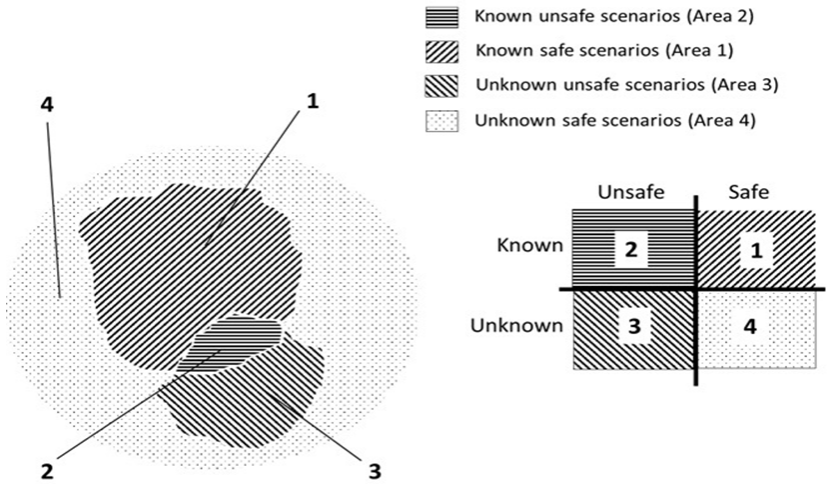
Use case 1: Safety Of The Intended Functionality (SOTIF)

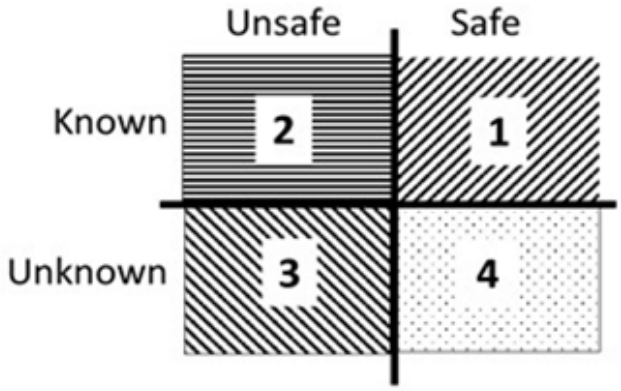
SOTIF(ISO/PAS 21448) emphasizes to avoid unreasonable risks due to expected functional performance limitations.

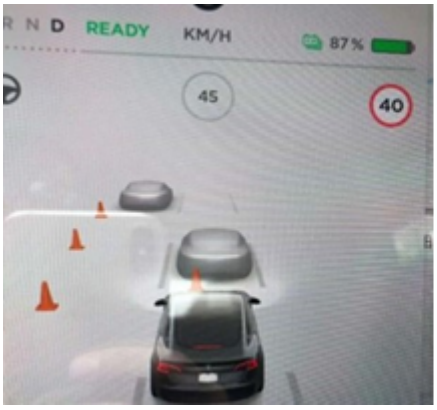
The background of the birth of SOTIF is the development of intelligent driving

If classified according to the functional chain of intelligent driving: perception-decision-execution, the "functional performance limitation" is reflected in three aspects:

- Sensor perception limitations lead to scene recognition errors (including missed recognition of driver misoperation)
- Insufficient deep learning causes the decision algorithm to judge the scene incorrectly (including the wrong response to the driver's misoperation)
- Actuator function limitations lead to deviation from the ideal target







- For Area2 (known unsafe scenarios), the basic idea of SOTIF is to identify risk scenarios through safety analysis, and develop countermeasures against risk scenarios.
- For Area3 (unknown unsafe scenarios), various scenarios that a car may encounter under various road conditions need to be identified (in theory) in the early stage of development

Use case 2: Cooperative intelligent transportation systemvehicular communication data exchange system

Vehicle road collaboration platform integrates sensing, communication, computing, control and other technologies, based on standardized communication protocol, realizes mutual mapping between physical space and information space, including "vehicle, traffic, environment" and other elements, standardized interaction and efficient collaboration, and uses cloud computing big data capabilities to solve systematic resource optimization and configuration problems.

The platform provides dynamic basic data such as vehicle operation, infrastructure, traffic environment and traffic management for intelligent vehicle and its users, management and service institutionsApplication scenario development, verification and commercial application of vehicle communication system based on various communication modes.

The vehicular communication data exchange system to support following functions:

No.	Classification	Communication	Application
1	Safety	V2V	Forward collision warning
2		V2V/N2I	Intersection collision warning
3		V2V/N2I	Left turn assis
4		V2V	Blind area warning / lane change assistance
5		V2V	Reverse overtaking warning
6		V2V-Event	Emergency braking warning
7		V2V-Event	Warning of abnormal vehicles
8		V2V-Event	Vehicle out of control warning
9		V2I	Warning of road danger
10		V2I	Speed limit warning
11		V2I	Red light warning
12		V2P/V2I	Collision warning for vulnerable traffic participants

13	Efficiency	V2I	`Green Wave Speed Guidance`
14		V2I	`In-vehicle signs`
15		V2I	`Congestion alert ahead`
16		V2V	Emergency vehicle reminder
17	Information service	V2I	Car near field payment

Use case 3: Autonomous Valet Parking

Autoware is the world's first "all-in-one" open-source software for self-driving vehicles hosted under the Autoware Foundation.

The Autoware.Auto project, based on ROS 2, is the next generation successor of the Autoware.AI project, based on ROS 1.

The major differentiators of Autoware.Auto compared to Autoware.AI are:

1. Modern software engineering best practices including code reviews, continuous integration testing, thorough documentation, thorough test coverage, style and development guides
2. Improved system architecture and module interface design (including messages and APIs)
3. An emphasis on reproducibility and determinism at the library, node, and system levels

The Autonomous Valet Parking (AVP) demonstration uses Autoware.Auto to provide a valet parking service in the parking lot ODD.

blocked URL

Autonomous valet parking

The AVP demonstration uses Autoware.Auto to provide the following functions:

1. Automatically drive a car from a pre-defined drop-off zone (e.g. the entrance to a carpark) to a parking spot indicated by an external system.
2. Park a car in a parking spot, starting in a lane near that parking spot.
3. Drive out of a parking spot.
4. Drive to a pre-defined pick-up zone (e.g. the exit from a carpark).
5. Automatically stop for obstacles while achieving the above.

<use case 3>

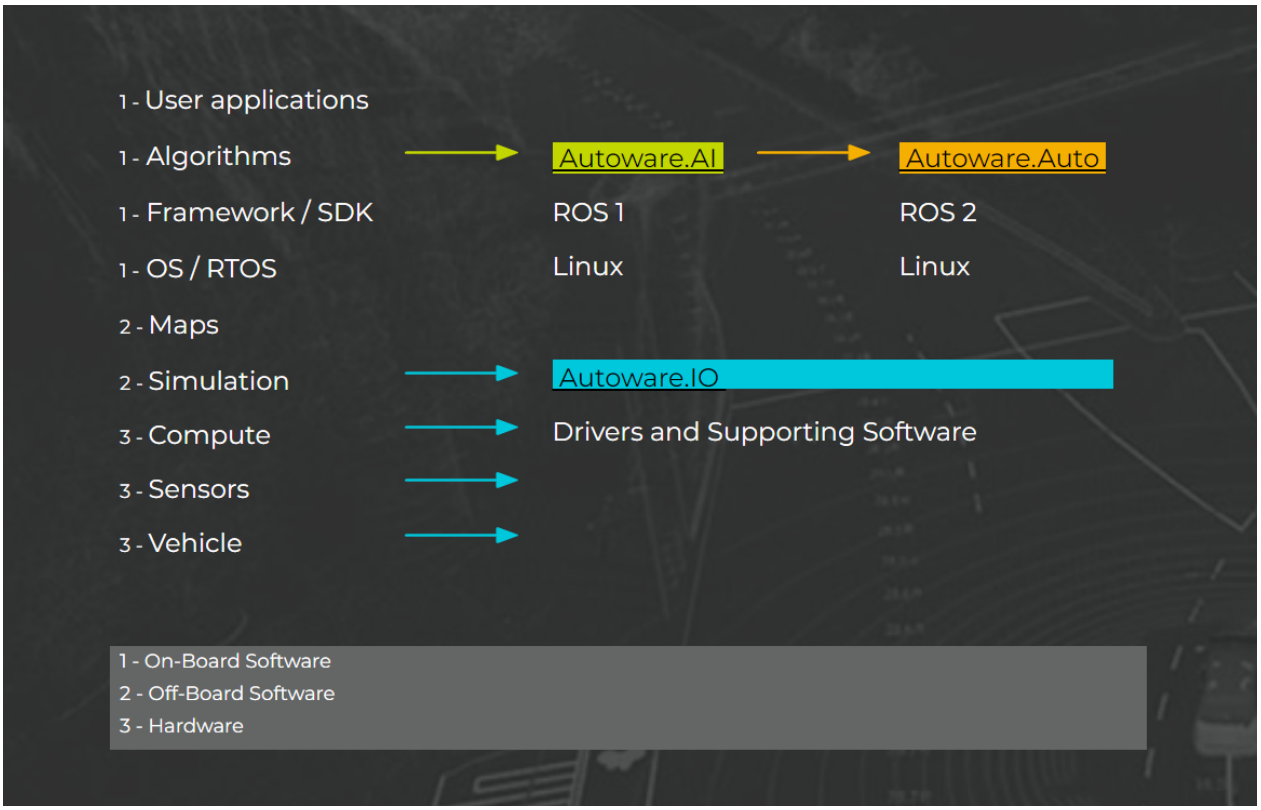
Where on the Edge

In the aspect of road edge computing, the new road side systemRSU in the future will integrate a variety of communication methods such as lte-v / 5G, provide a variety of sensor interfaces and local map system, provide signal timing information and surrounding moving target information, and provide vehicle collaborative decision-making and other technologies and capabilities to build a road side edge computing node. The coordinated driving of workshop (V2V) and vehicle road (V2I) can reduce the probability of accidents by means of accident warning and avoidance. The car needs to interact the data obtained by local radar and camera with the surrounding vehicles and road infrastructure through the edge gateway, and improve the perception range, so as to achieve the cooperation between vehicles and between vehicles and roads, provide the driver with collision warning, lane change warning, adaptive cruise and other assistance, and take over the car to prevent accidents when necessary.

Overall Architecture

<This could inform the non-technical audience, but now is more geared towards a more engaged, technical audience>

< Blue print's relation to Akraino generic architecture, how it relates to it >



< This section will use the Akraino architecture document as reference>

System architecture for the AVP ODD

The system architecture that was developed to address the AVP ODD in Autoware.Auto is given below:

blocked URL
Autoware.Auto AVP Architecture

Prerequisites

To run this demo, you will need to provide the following.

- A point cloud map of a carpark, for localisation.
- A HDMap (vector map) of a carpark, for navigation.
- A simulated world of a carpark, if the demo will be run on a simulator.
- A simulated version of your car.

A sample carpark is available, based on a real single-level carpark in San Jose, California.

A sample car is available, based on a real-life vehicle used by many members of the Autoware Foundation.

Platform Architecture

<Hardware components should be specified with model numbers, part numbers etc>

The physical AVP demo was tested with a Lexus RX 450h with

- the Pacmod 3.0 DBW interface
- 2 Velodyne VLP-32Cs
- an AutonomouStuff Spectra industrial PC.
- the AutonomouStuff Speed and Steering Control (SSC) software.

Software Platform Architecture

<Software components with version/release numbers >

AVP Simulation

The simulation AVP demo was tested with hardware that satisfies requirements for the LGSVL simulator. To run the LGSVL simulator, you will need an NVIDIA graphics card. Additional information about requirements can be found here.

<EDGE Interface>

<ETSI MEC Interaction>

APIs

APIs with reference to Architecture and Modules

High Level definition of APIs are stated here, assuming Full definition APIs are in the API documentation

Hardware and Software Management

Licensing

- GNU/common license