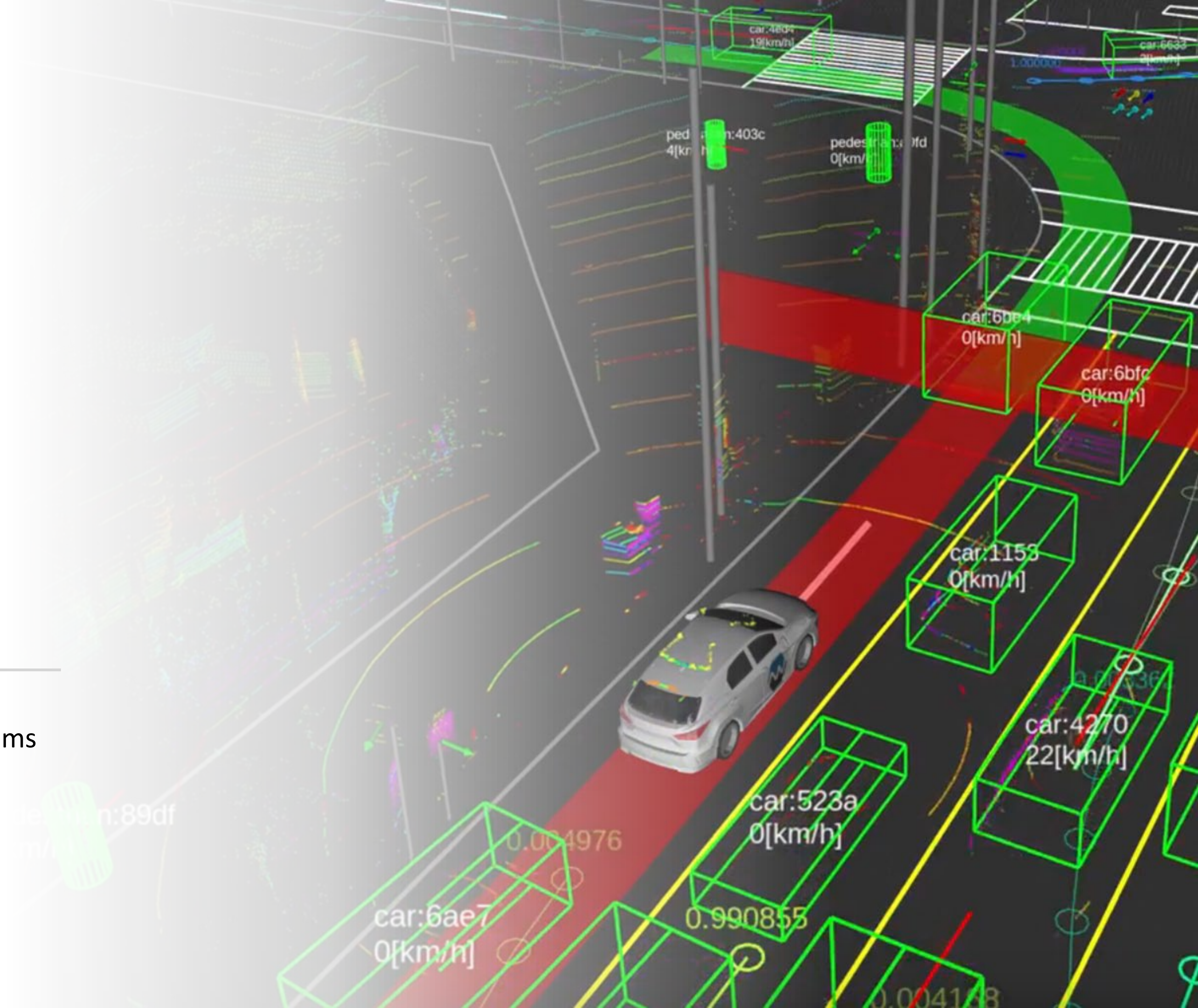
IACOS

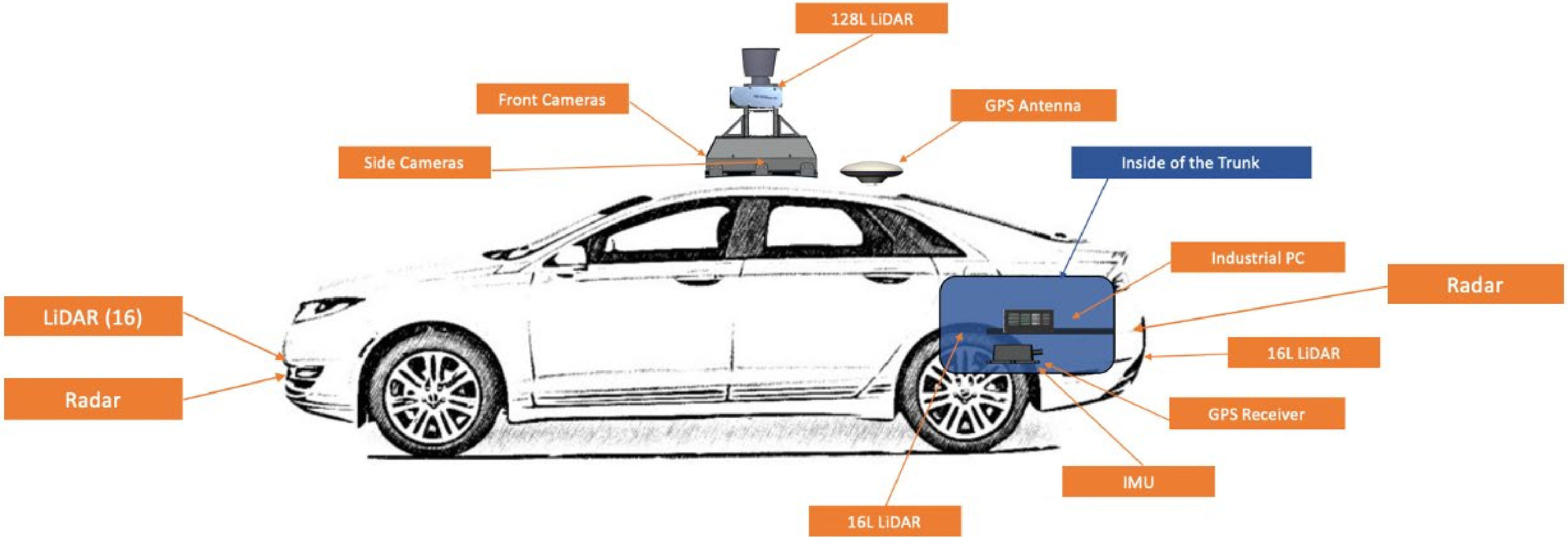
Intelligent Autonomous Cooperative Systems

Week 3 – Autoware

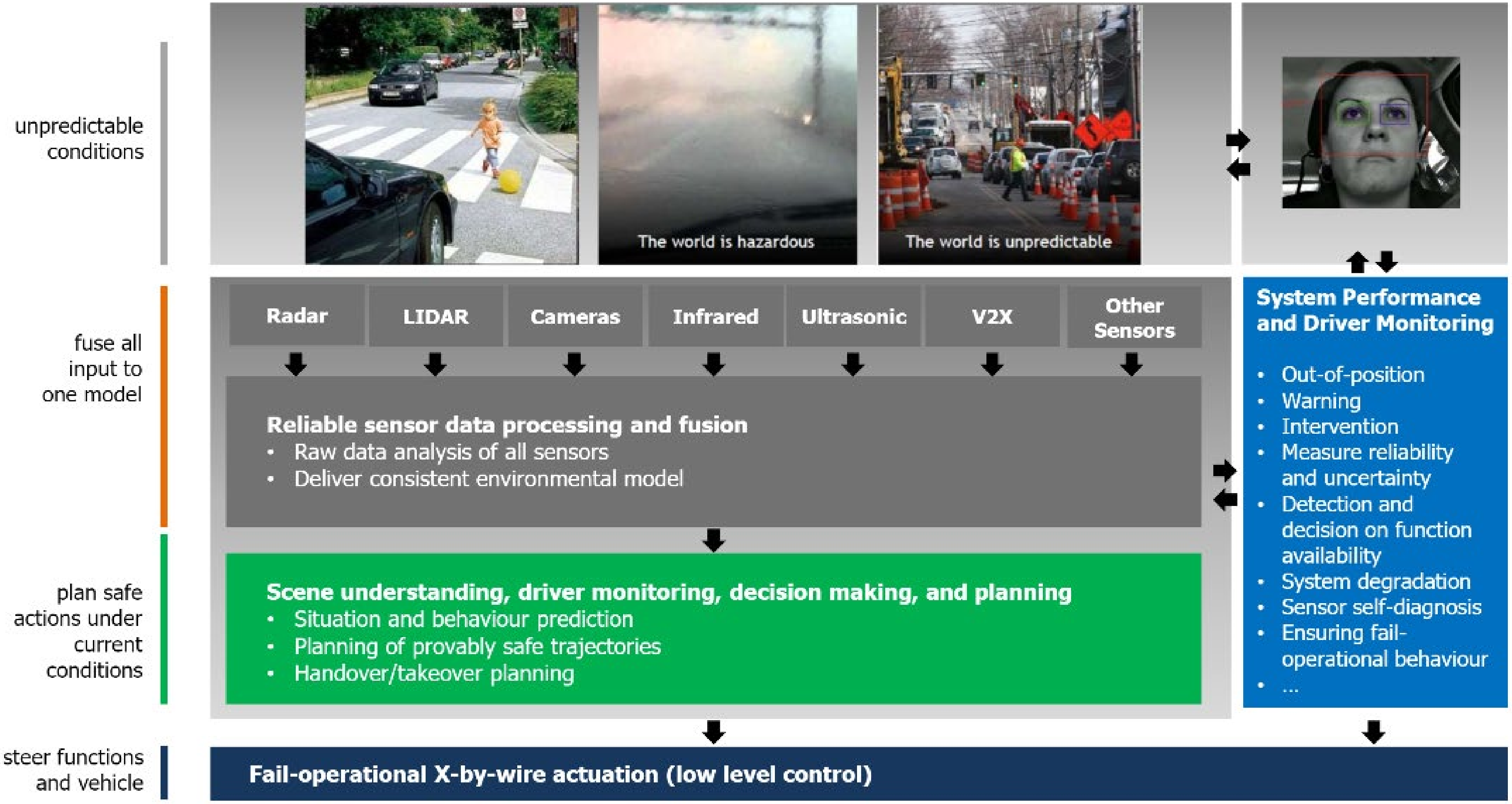
Ricardo Severino and Luis Miguel Pinho



Autonomous Driving



Autonomous Driving Architecture



unpredictable conditions

The world is hazardous

The world is unpredictable

fuse all input to one model

Radar | LIDAR | Cameras | Infrared | Ultrasonic | V2X | Other Sensors

Reliable sensor data processing and fusion

- Raw data analysis of all sensors
- Deliver consistent environmental model

plan safe actions under current conditions

Scene understanding, driver monitoring, decision making, and planning

- Situation and behaviour prediction
- Planning of provably safe trajectories
- Handover/takeover planning

System Performance and Driver Monitoring

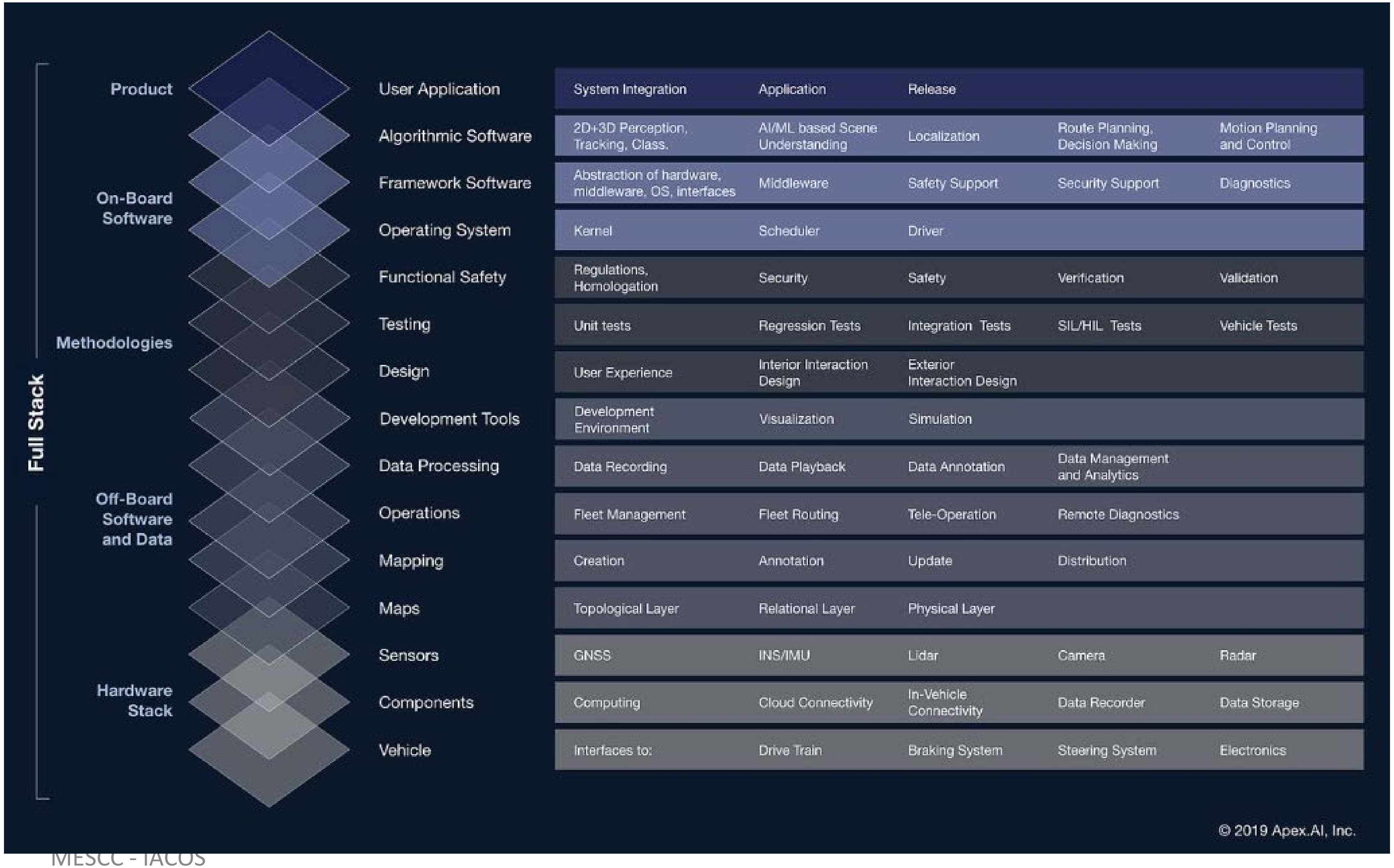
- Out-of-position
- Warning
- Intervention
- Measure reliability and uncertainty
- Detection and decision on function availability
- System degradation
- Sensor self-diagnosis
- Ensuring fail-operational behaviour
- ...

steer functions and vehicle

Fail-operational X-by-wire actuation (low level control)

[Source: based on ECSEL Project RobustSense, 2016]

Autonomous Driving Stack



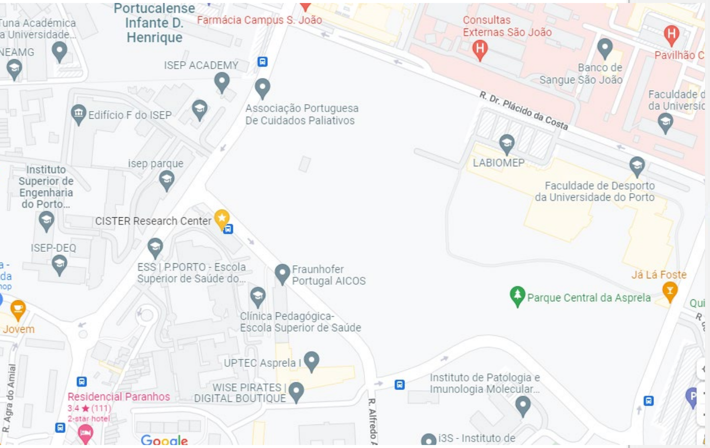
Full Stack

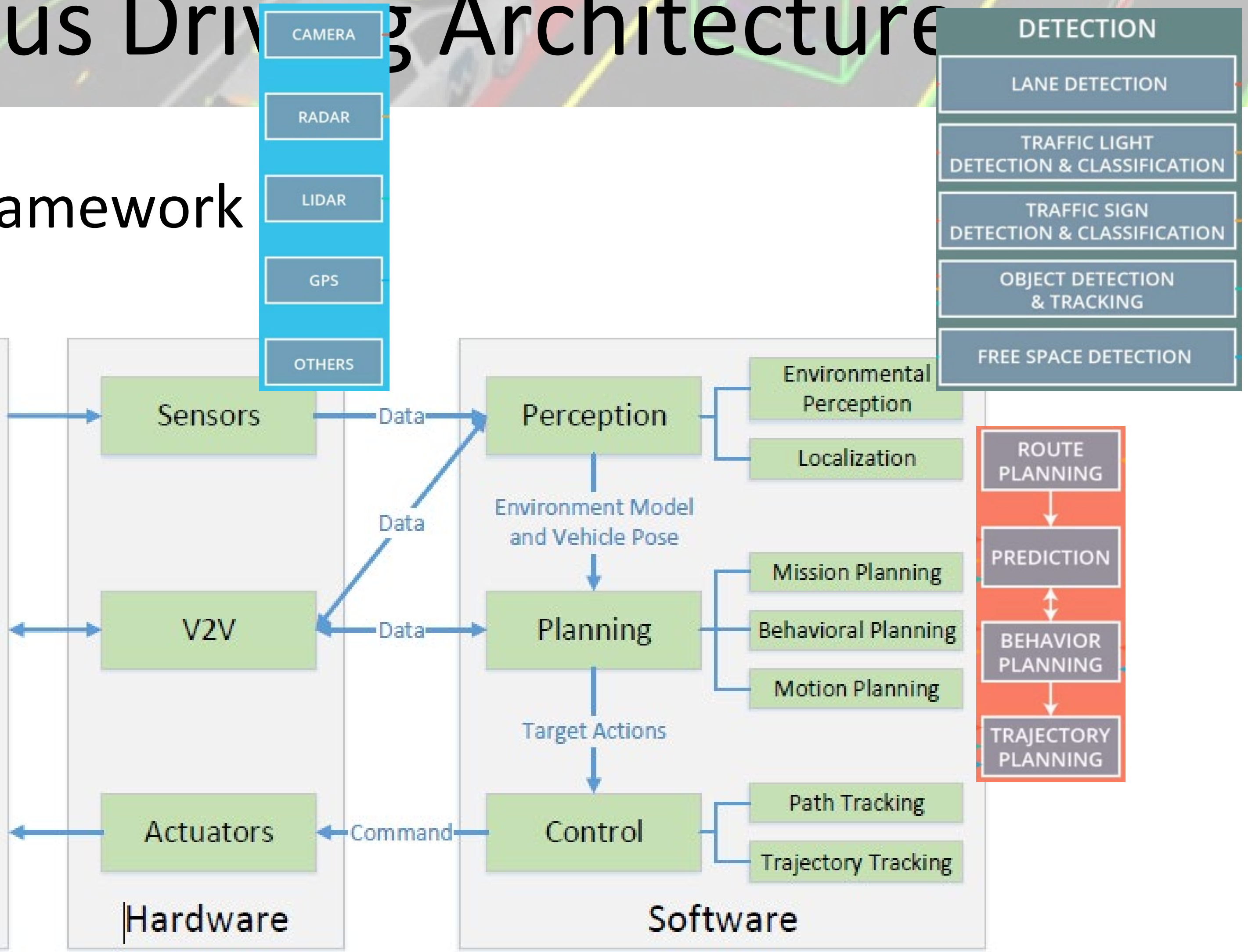
Product	User Application	System Integration	Application	Release		
On-Board Software	Algorithmic Software	2D+3D Perception, Tracking, Class.	AI/ML based Scene Understanding	Localization	Route Planning, Decision Making	Motion Planning and Control
	Framework Software	Abstraction of hardware, middleware, OS, interfaces	Middleware	Safety Support	Security Support	Diagnostics
	Operating System	Kernel	Scheduler	Driver		
Methodologies	Functional Safety	Regulations, Homologation	Security	Safety	Verification	Validation
	Testing	Unit tests	Regression Tests	Integration Tests	SIL/HIL Tests	Vehicle Tests
	Design	User Experience	Interior Interaction Design	Exterior Interaction Design		
	Development Tools	Development Environment	Visualization	Simulation		
Off-Board Software and Data	Data Processing	Data Recording	Data Playback	Data Annotation	Data Management and Analytics	
	Operations	Fleet Management	Fleet Routing	Tele-Operation	Remote Diagnostics	
	Mapping	Creation	Annotation	Update	Distribution	
	Maps	Topological Layer	Relational Layer	Physical Layer		
Hardware Stack	Sensors	GNSS	INS/IMU	Lidar	Camera	Radar
	Components	Computing	Cloud Connectivity	In-Vehicle Connectivity	Data Recorder	Data Storage
	Vehicle	Interfaces to:	Drive Train	Braking System	Steering System	Electronics

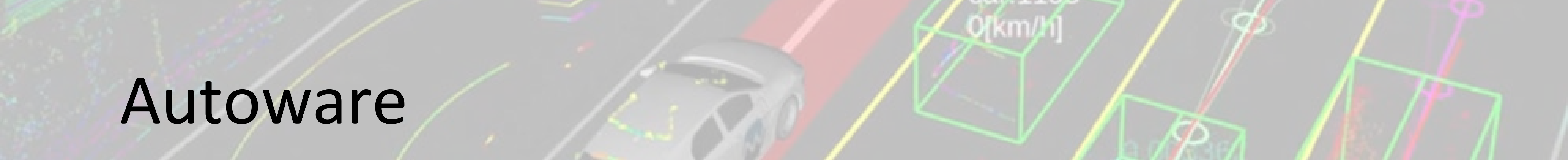
Autonomous Driving Architecture

- Generic AD framework

MAP

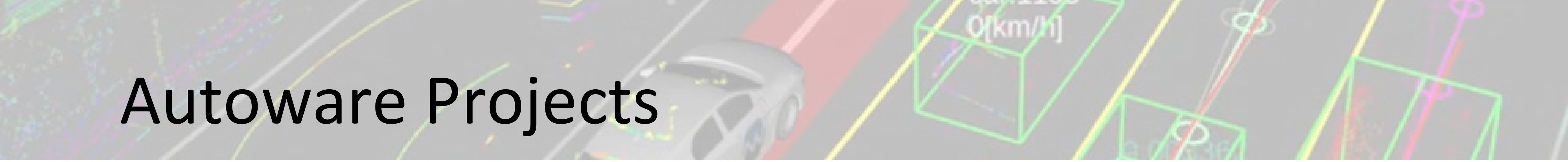

Portucalense Infante D. Henrique
Farmácia Campus S. João
Consultas Externas São João
Tuna Académica da Universidade
NEAMG
ISEP ACADEMY
Banco de Sangue São João
Pavilhão C
Edifício F do ISEP
Associação Portuguesa De Cuidados Paliativos
R. Dr. Plácido da Costa
Faculdade da Universid
Isep parque
LABIOMEP
Instituto Superior de Engenharia do Porto
Faculdade de Desporto da Universidade do Porto
CISTER Research Center
ISEP-DEQ
ESS | P.PORTO - Escola Superior de Saúde do...
Fraunhofer Portugal AICOS
Já Lá Foste
Parque Central da Asprela
Clínica Pedagógica-Escola Superior de Saúde
Jovem
UPTEC Asprela I
WISE PIRATES | DIGITAL BOUTIQUE
Instituto de Patologia e Imunologia Molecular...
Residencial Paranhos
3.4 ★ (111)
2-star hotel
R. Alfredo A
i3S - Instituto de
R. Agra do Amial
Google





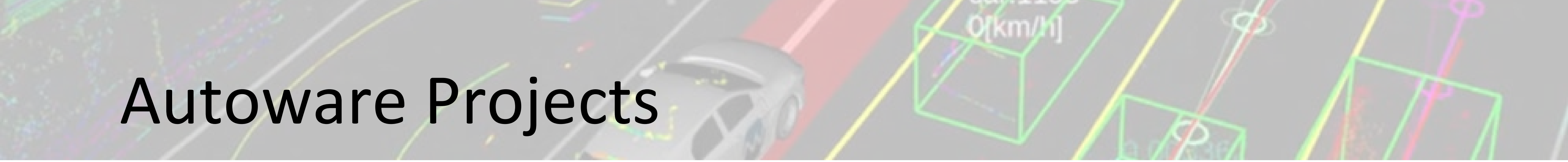
Autoware

- Autoware is an open-source software project for autonomous driving
    - Built on Robot Operating System (ROS)
    - Enables both research and commercial deployment of autonomous driving
- Autoware was started in 2015 by Professor Shinpei Kato at Nagoya University and Carnegie Mellon University
    - Today, it is supported by a Foundation
    - And a large autonomous driving open-source community
- Used by more than 100 companies and runs on more than 30 vehicles in more than 20 different countries
- Automotive OEMs are using Autoware for Mobility as a Service (MaaS) development
- Autoware has been qualified to run on driverless vehicles on public roads in Japan since 2017



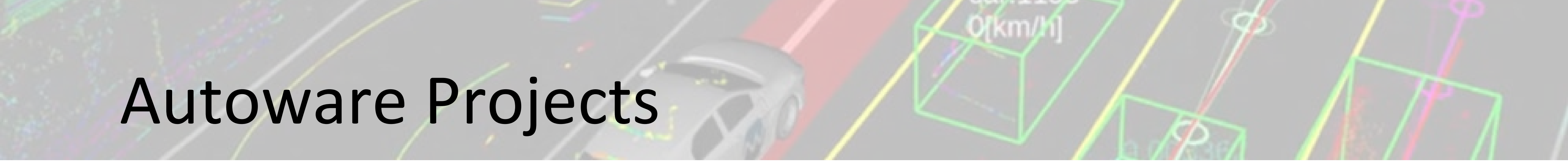
Autoware Projects

- First project was Autoware.AI
  - Officially released in August 2015
  - Development in Github
    - Mainly developed by students/researchers
  - Based on ROS 1
  - Intended to be full self-driving stack (SAE Level 3/4)
  - Supporting many vehicle platforms
  - Contains over 209,000 lines of code
  - Multiple implementations of algorithms in:
    - Control, Localization, Perception, Planning, Simulation
  - Supports multiple simulators (LGSVL, CARLA)



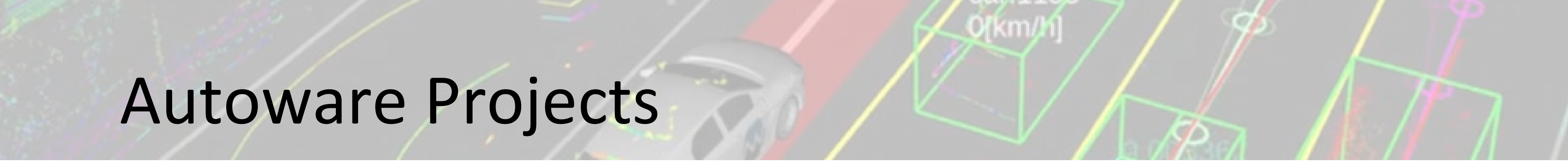
Autoware Projects

- First project was Autoware.AI
  - Not designed for functional safety
  - Grew organically - code quality and development best-practices not priorities
  - Development started early in Autonomous Vehicle revolution
  - R&D and testing
  - Since January 2021 on maintenance mode
    - End of Life December 2022



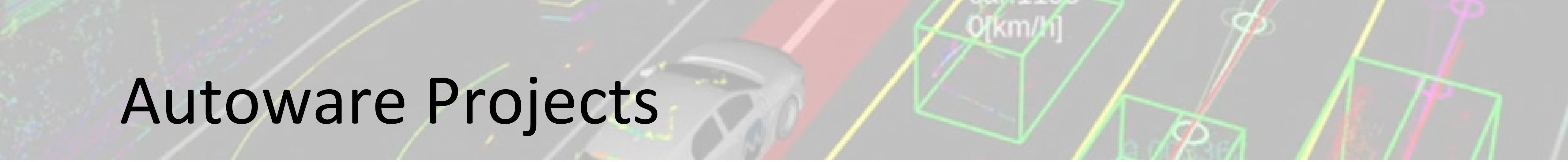
Autoware Projects

- Autoware.IO
  - Extensions to Autoware - members’ solutions in platforms which support Autoware.Auto and Autoware.AI software
  - Device drivers for sensors
  - By-wire controllers for vehicles
  - Heterogeneous hardware reference platforms



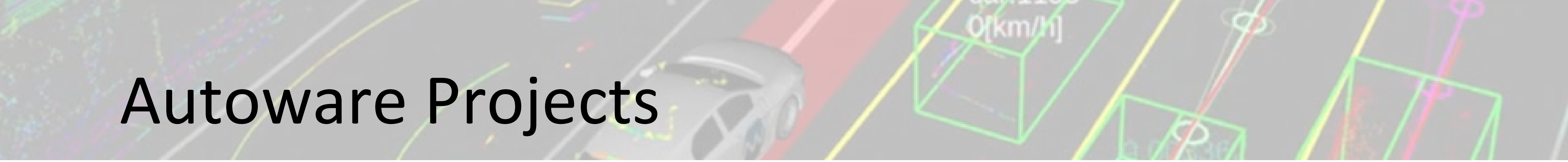
Autoware Projects

- Autoware.Auto
  - Autoware.Auto was introduced as Autoware reimagined
  - Based on ROS 2 and DDS
  - Open-Source Autonomous Vehicle Software Stack
  - Supported by software engineering practices
    - Intended to support specific Operational Design Domains (ODDs)
    - Provides a modular reference design with clearly-defined APIs
    - Importance given to documentation
  - Redesigned architecture
    - Allows extension with other (potentially proprietary) vendor-developed modules
  - Development in Gitlab



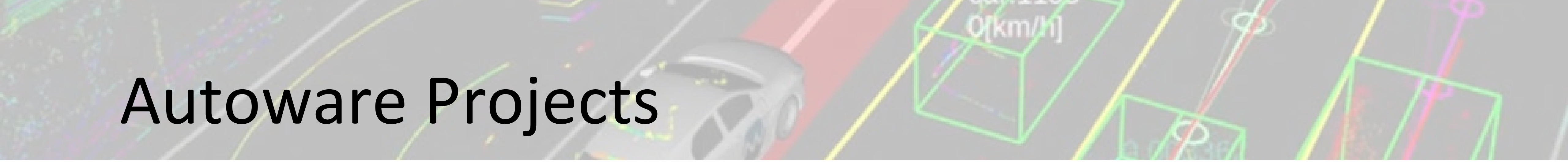
Autoware Projects

- Autoware.Auto
  - Importance given to autonomous driving important aspects
    - Safety
    - Code quality (multiple linters/static analyzers enforced)
    - Static memory allocation
    - Documentation
- Development began in August 2018
  - Initial framework and development by Apex.AI (in partnership with Tier IV)
- First release in August 2019
- Ongoing, active development by most AWF member organizations
  - Development managed by the Autoware Software Working Group (ASWG)



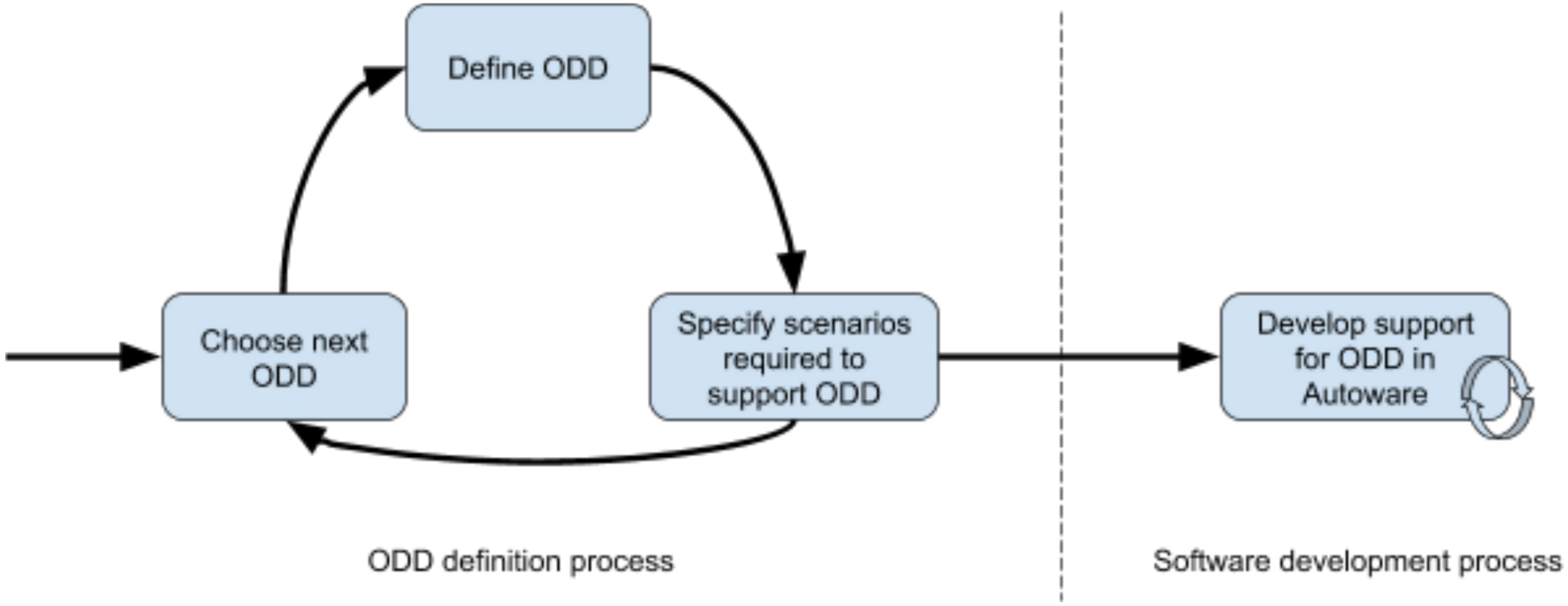
Autoware Projects

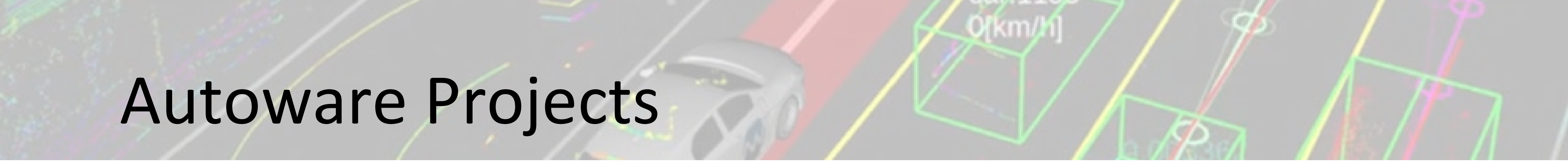
- Autoware.Auto
  - Currently targets Foxy ROS 2 release
  - Docker-based development/runtime environment, ADE
    - Used in IACOS laboratories
    - Supports the addition of simulation environments as Docker volumes in ADE
  - Supports the following functions
    - Sensing
      - E.g. Cameras, GPS, sensors, …
      - Additional drivers from ROS2
    - Localization
    - Ground Filtering (Ray Classifier)
    - Object Detection (Voxel Grid / Euclidean Clustering)
    - Vehicle Interface (currently through Linux SocketCAN or simulation-specific interfaces)
    - Motion Control (Pure Pursuit / MPC)



Autoware Projects

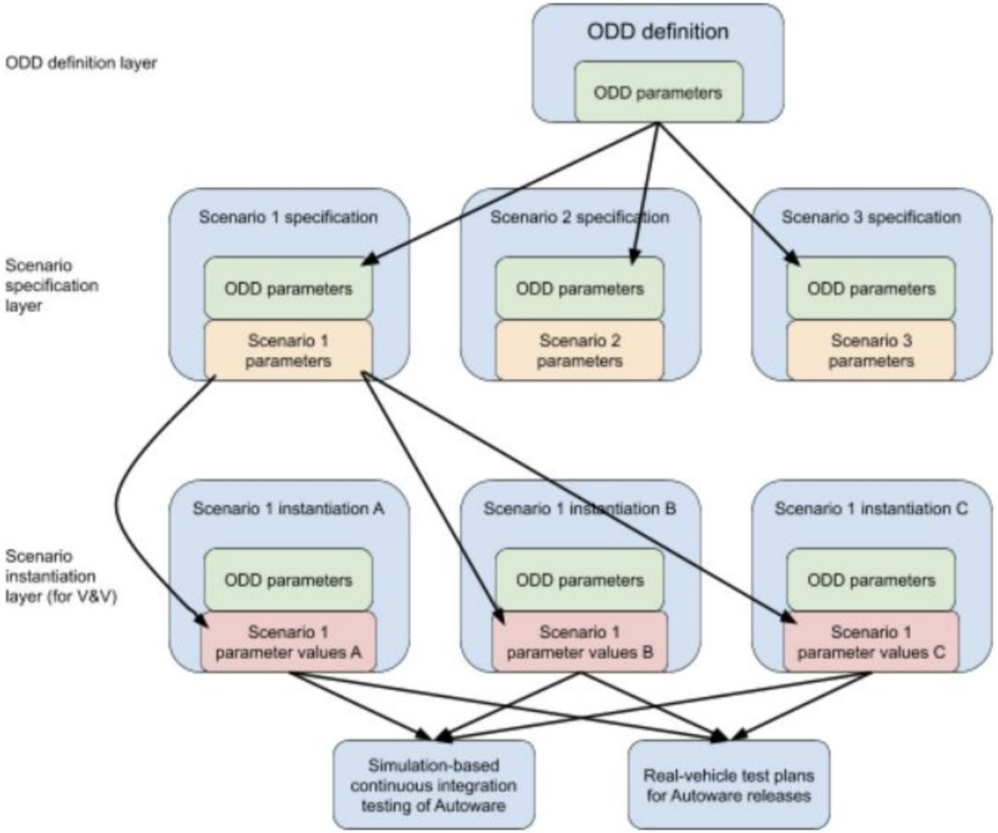
- Autoware.Auto shifted towards a use-case-based development approach
  - Use-cases: Operational Design Domain (ODD)
  - Determined by Autoware foundation leadership, approved by members

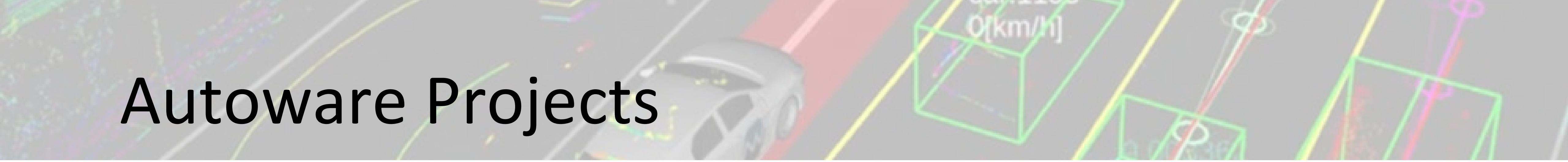

Define ODD
Choose next ODD
Specify scenarios required to support ODD
Develop support for ODD in Autoware
ODD definition process
Software development process



Autoware Projects

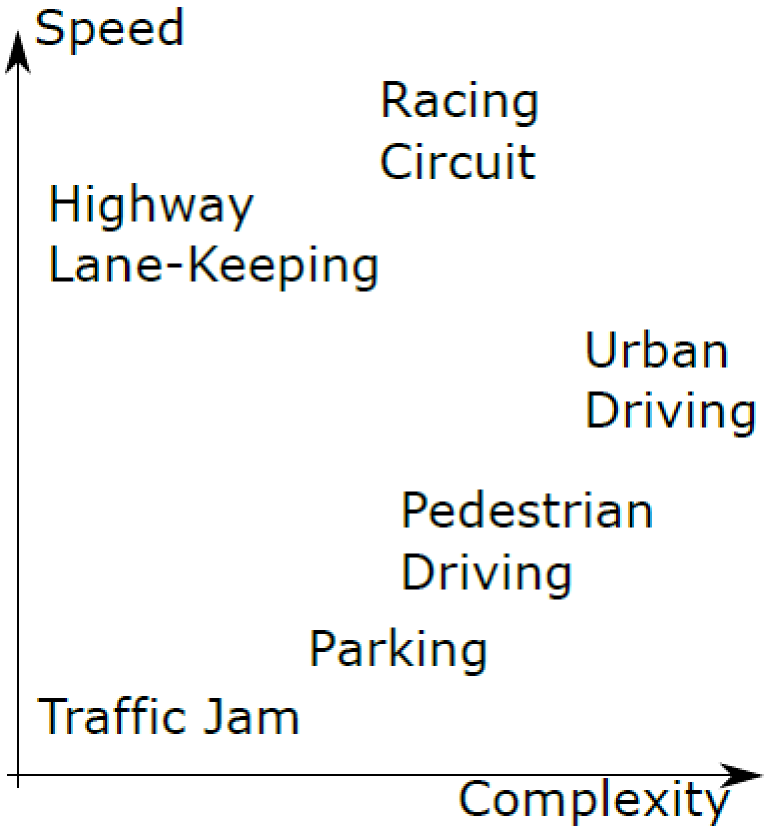
- Operational Design Domain (ODD)
  - Technical Steering Committee determines next ODD
  - ODD Workgroup scopes and refines ODD
    - Creates tests scenarios
  - Autonomy Software WG implements requirements in Autoware.Auto
  - Autonomy Hardware WG creates HW reference configuration and verifies HW compatibility
  - V&V Task force uses scenarios to validate implementation


ODD definition layer
ODD definition
ODD parameters
Scenario specification layer
Scenario 1 specification
ODD parameters
Scenario 1 parameters
Scenario 2 specification
ODD parameters
Scenario 2 parameters
Scenario 3 specification
ODD parameters
Scenario 3 parameters
Scenario instantiation layer (for V&V)
Scenario 1 instantiation A
ODD parameters
Scenario 1 parameter values A
Scenario 1 instantiation B
ODD parameters
Scenario 1 parameter values B
Scenario 1 instantiation C
ODD parameters
Scenario 1 parameter values C
Simulation-based continuous integration testing of Autoware
Real-vehicle test plans for Autoware releases



Autoware Projects

- Operational Design Domain (ODD)
  - Autonomous Valet Parking
  - Cargo Delivery
  - Autonomous Racing
  - Robo-Bus/Shuttle
  - Robo-taxi



Speed

Racing Circuit

Highway Lane-Keeping

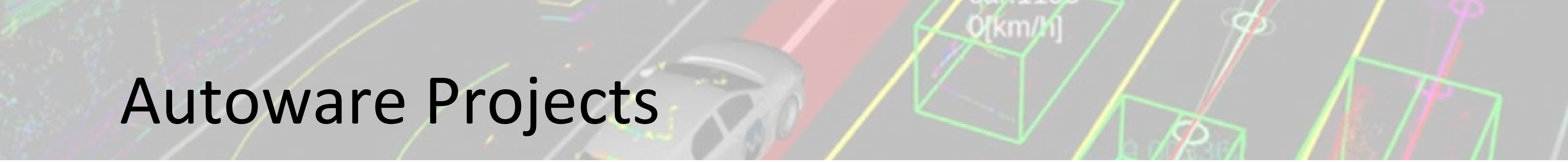
Urban Driving

Pedestrian Driving

Parking

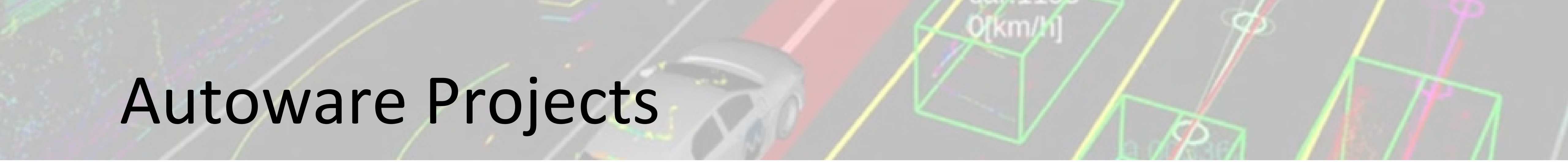
Traffic Jam

Complexity



Autoware Projects

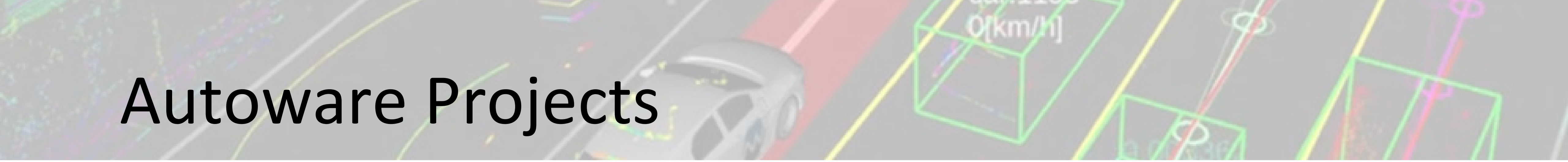
- Autoware Core/Universe
  - Autoware.Auto approach allowed stringent requirements such as
    - best-in-class software engineering practices
    - 100% code coverage
    - coding style guidance
    - exhaustive testing
  - However, it restricted evolutions and scaling
    - And commercialization

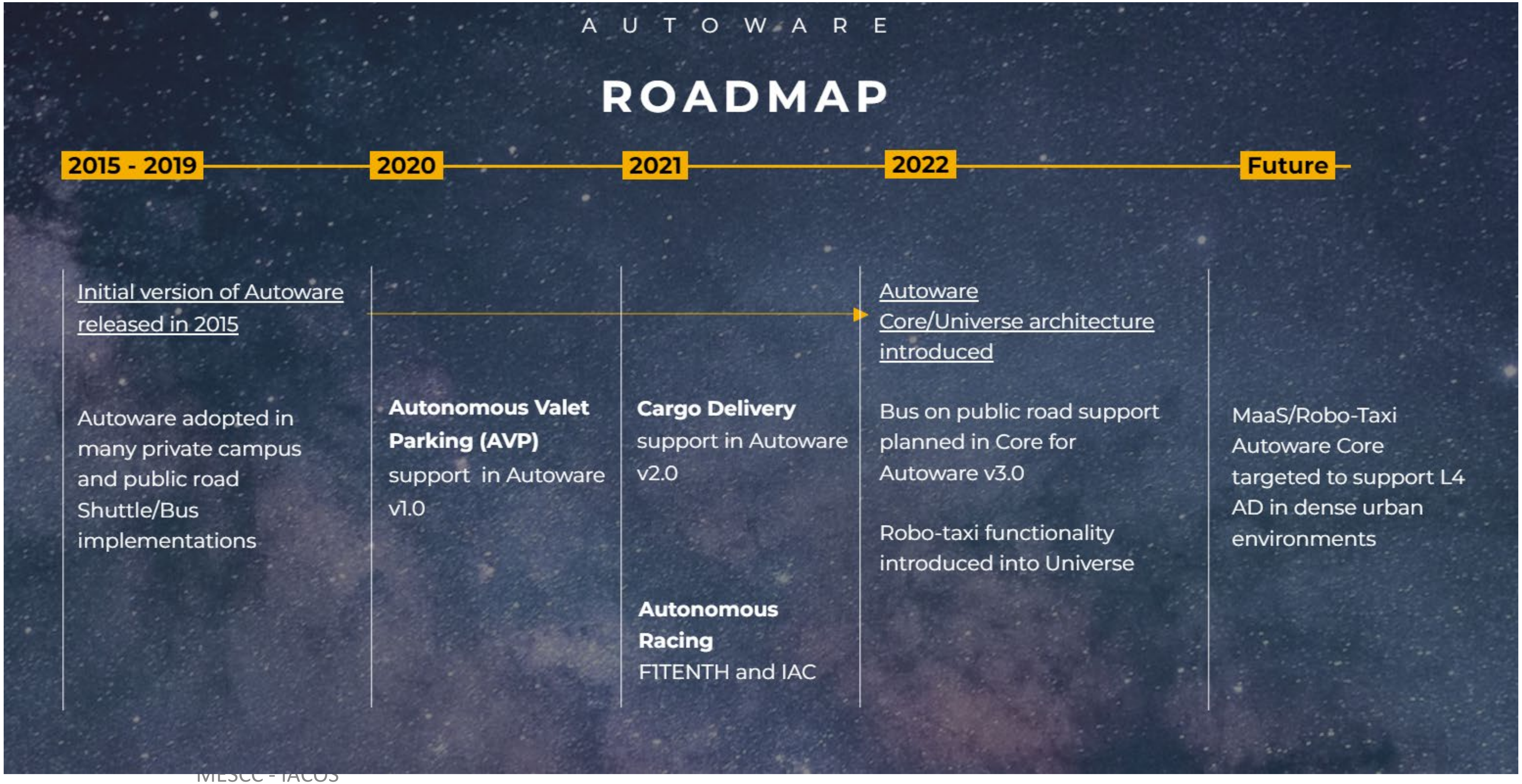


Autoware Projects

- Autoware Core/Universe
  - The new Core/Universe paradigm addresses this concern
  - Autoware Core will be an enhancement on Autoware.Auto project
    - Fundamental core following the same process as Autoware.Auto
  - Autoware Universe extends Autoware's scope broader
    - Enabling contributors to develop additional components on top of Autoware Core
    - Without being bound by stringent development requirements
    - Relaxed development, easy prototype
    - Lower the barriers for autonomous vehicle development and facilitates commercial solutions
  - Complemented by the OpenAD Kit initiative
    - Reference Platform for development of Software Defined Vehicle (SDV)

Autoware Projects





AUTOWARE

ROADMAP

2015 - 2019

Initial version of Autoware released in 2015

Autoware adopted in many private campus and public road Shuttle/Bus implementations

2020

Autonomous Valet Parking (AVP) support in Autoware v1.0

2021

Cargo Delivery support in Autoware v2.0

Autonomous Racing F1TENTH and IAC

2022

Autoware Core/Universe architecture introduced

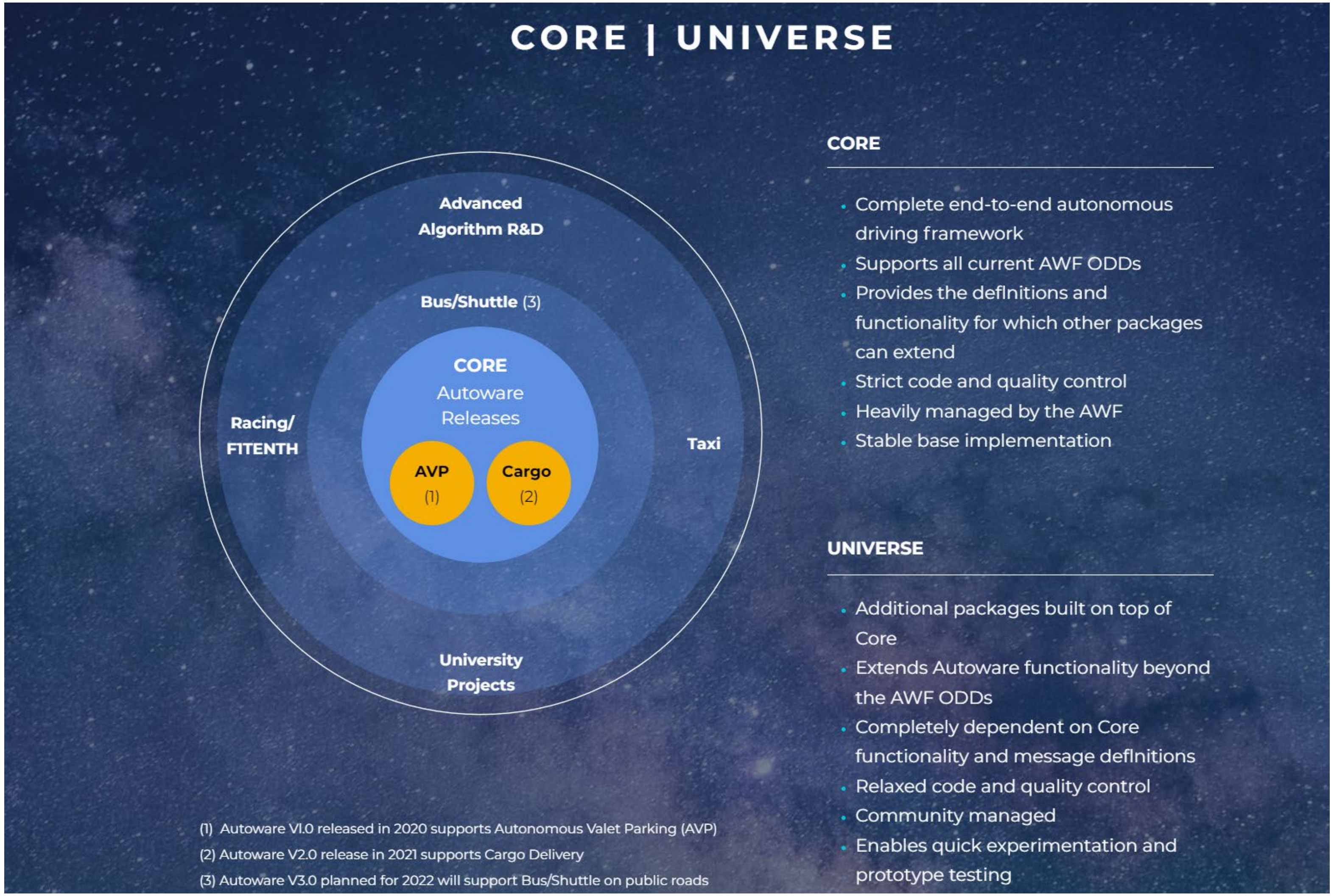
Bus on public road support planned in Core for Autoware v3.0

Robo-taxi functionality introduced into Universe

Future

MaaS/Robo-Taxi Autoware Core targeted to support L4 AD in dense urban environments

Autoware Projects



CORE | UNIVERSE

Advanced Algorithm R&D

Bus/Shuttle (3)

CORE

Autoware Releases

AVP (1)

Cargo (2)

Racing/ FITENTH

Taxi

University Projects

(1) Autoware V1.0 released in 2020 supports Autonomous Valet Parking (AVP)

(2) Autoware V2.0 release in 2021 supports Cargo Delivery

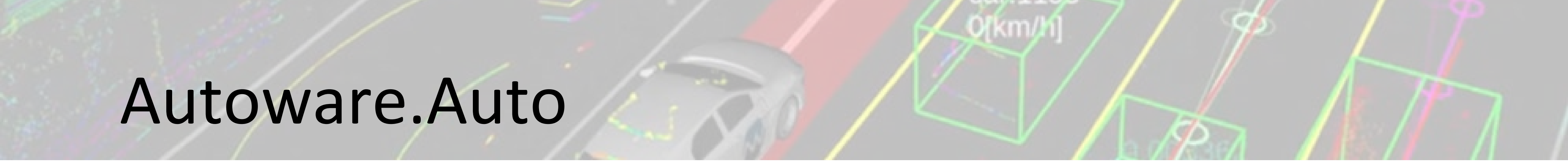
(3) Autoware V3.0 planned for 2022 will support Bus/Shuttle on public roads

CORE

- Complete end-to-end autonomous driving framework
- Supports all current AWF ODDs
- Provides the definitions and functionality for which other packages can extend
- Strict code and quality control
- Heavily managed by the AWF
- Stable base implementation

UNIVERSE

- Additional packages built on top of Core
- Extends Autoware functionality beyond the AWF ODDs
- Completely dependent on Core functionality and message definitions
- Relaxed code and quality control
- Community managed
- Enables quick experimentation and prototype testing

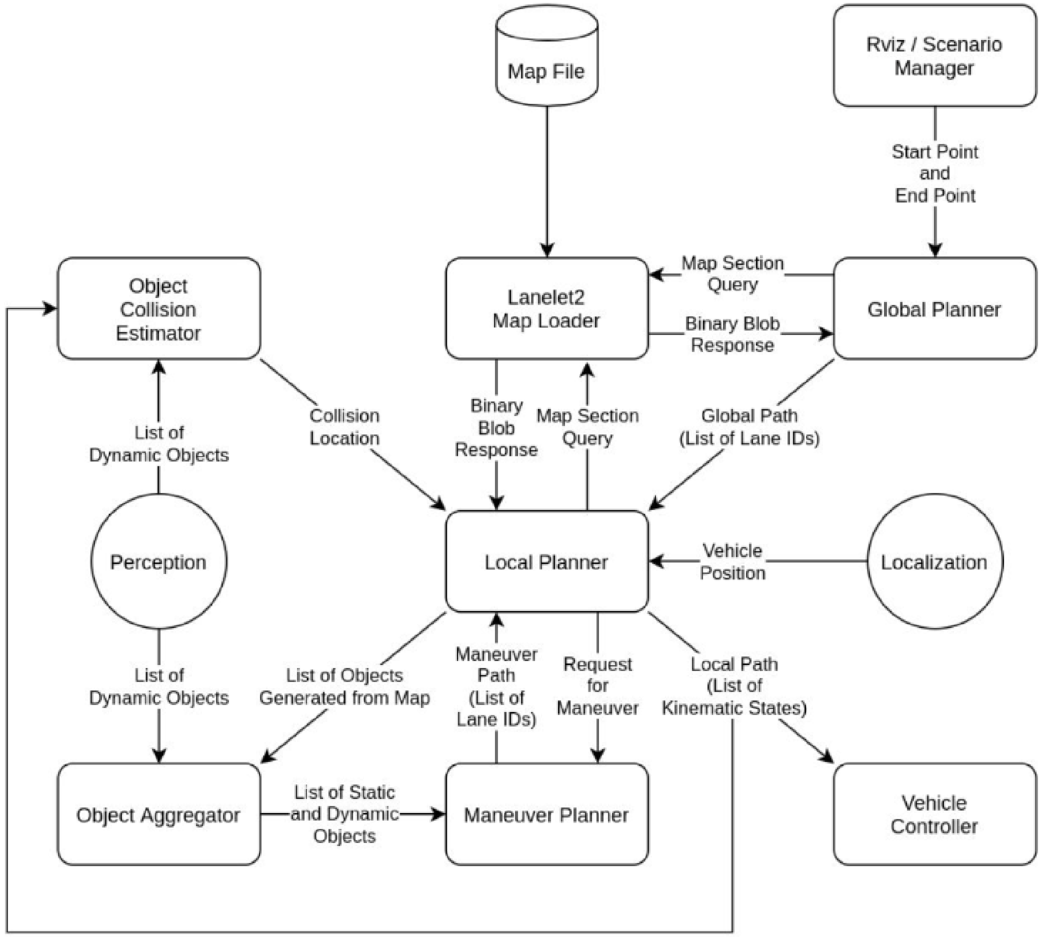


Autoware.Auto

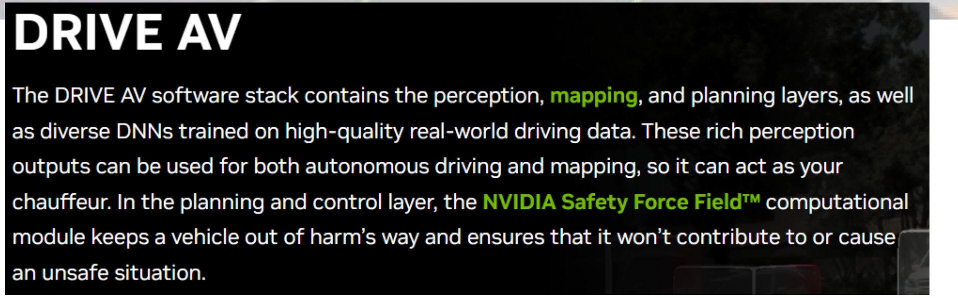
- IACOS will address Autoware.Auto
  - Stable version 1.0.0
    - Gitlab
    - Autonomous Valet Parking ODD
    - https://gitlab.com/autowarefoundation/autoware.auto/AutowareAuto
  - Version 2 moved back to github
    - Basis of Autoware Core
    - Autoware.AI is also in github (stalled branch of Autoware core)

Autoware.Auto

- AVP

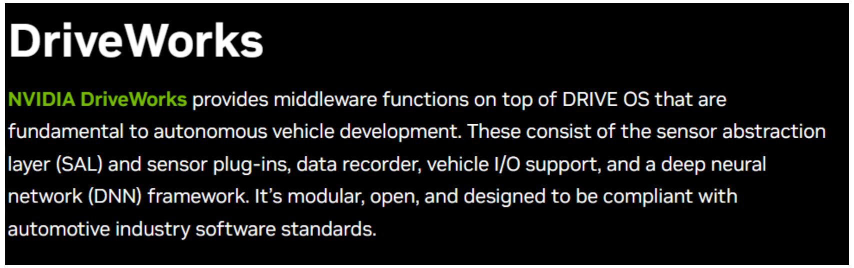


Other AD Frameworks



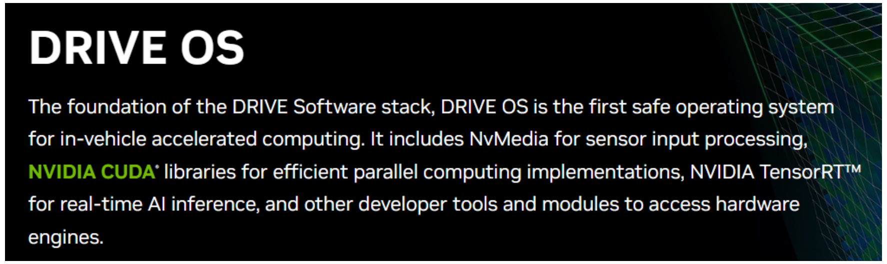
DRIVE AV

The DRIVE AV software stack contains the perception, mapping, and planning layers, as well as diverse DNNs trained on high-quality real-world driving data. These rich perception outputs can be used for both autonomous driving and mapping, so it can act as your chauffeur. In the planning and control layer, the NVIDIA Safety Force Field™ computational module keeps a vehicle out of harm's way and ensures that it won't contribute to or cause an unsafe situation.



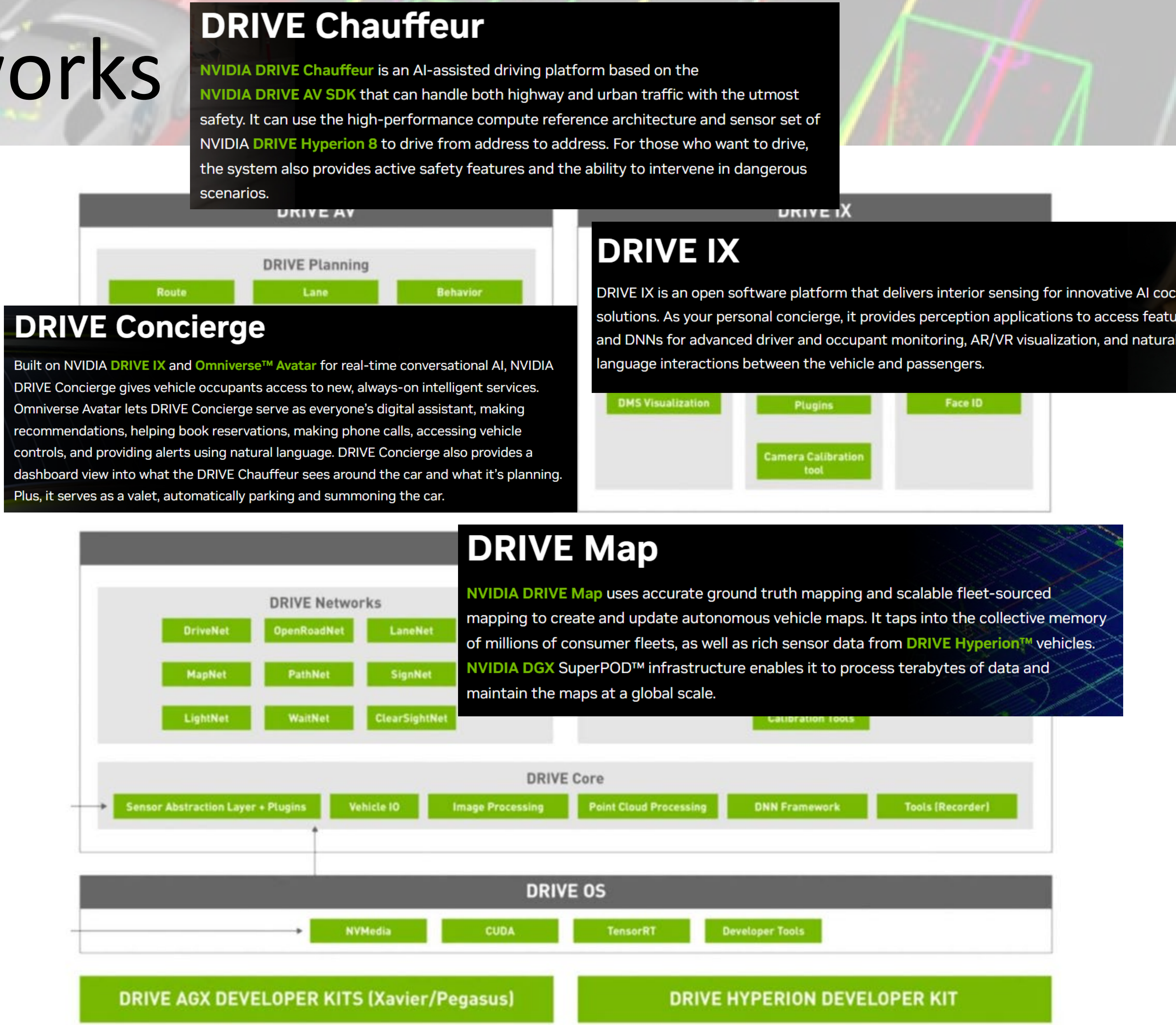
DriveWorks

NVIDIA DriveWorks provides middleware functions on top of DRIVE OS that are fundamental to autonomous vehicle development. These consist of the sensor abstraction layer (SAL) and sensor plug-ins, data recorder, vehicle I/O support, and a deep neural network (DNN) framework. It's modular, open, and designed to be compliant with automotive industry software standards.



DRIVE OS

The foundation of the DRIVE Software stack, DRIVE OS is the first safe operating system for in-vehicle accelerated computing. It includes NvMedia for sensor input processing, NVIDIA CUDA' libraries for efficient parallel computing implementations, NVIDIA TensorRT™ for real-time AI inference, and other developer tools and modules to access hardware engines.



DRIVE Chauffeur

NVIDIA DRIVE Chauffeur is an AI-assisted driving platform based on the NVIDIA DRIVE AV SDK that can handle both highway and urban traffic with the utmost safety. It can use the high-performance compute reference architecture and sensor set of NVIDIA DRIVE Hyperion 8 to drive from address to address. For those who want to drive, the system also provides active safety features and the ability to intervene in dangerous scenarios.

DRIVE IX

DRIVE IX is an open software platform that delivers interior sensing for innovative AI coc solutions. As your personal concierge, it provides perception applications to access featu and DNNs for advanced driver and occupant monitoring, AR/VR visualization, and natura language interactions between the vehicle and passengers.

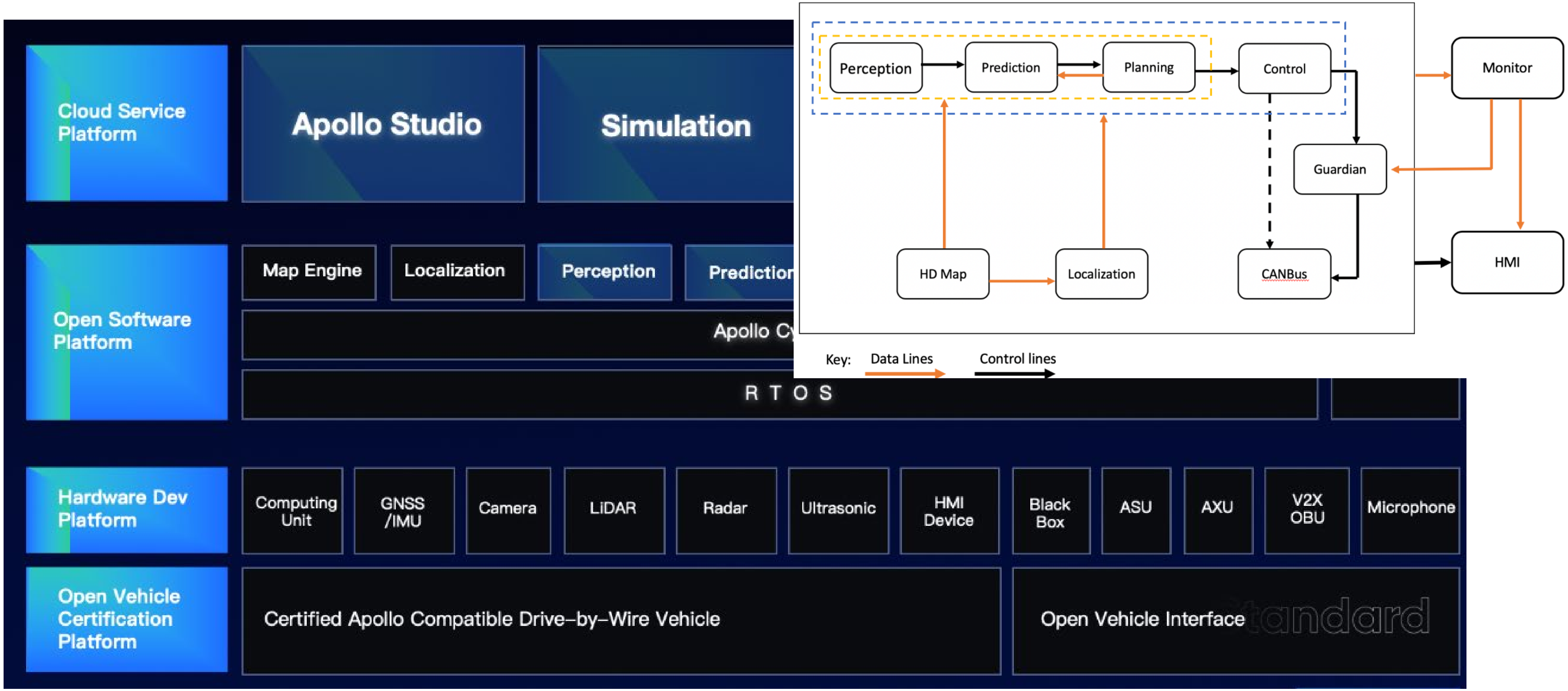
DRIVE Concierge

Built on NVIDIA DRIVE IX and Omniverse™ Avatar for real-time conversational AI, NVIDIA DRIVE Concierge gives vehicle occupants access to new, always-on intelligent services. Omniverse Avatar lets DRIVE Concierge serve as everyone's digital assistant, making recommendations, helping book reservations, making phone calls, accessing vehicle controls, and providing alerts using natural language. DRIVE Concierge also provides a dashboard view into what the DRIVE Chauffeur sees around the car and what it's planning. Plus, it serves as a valet, automatically parking and summoning the car.

DRIVE Map

NVIDIA DRIVE Map uses accurate ground truth mapping and scalable fleet-sourced mapping to create and update autonomous vehicle maps. It taps into the collective memory of millions of consumer fleets, as well as rich sensor data from DRIVE Hyperion™ vehicles. NVIDIA DGX SuperPOD™ infrastructure enables it to process terabytes of data and maintain the maps at a global scale.

Other AD Frameworks



Cloud Service Platform

Apollo Studio

Simulation

Open Software Platform

Map Engine

Localization

Perception

Predictior

Apollo Cy

RTOS

Hardware Dev Platform

Computing Unit

GNSS /IMU

Camera

LiDAR

Radar

Ultrasonic

HMI Device

Black Box

ASU

AXU

V2X OBU

Microphone

Open Vehicle Certification Platform

Certified Apollo Compatible Drive–by–Wire Vehicle

Open Vehicle Interface Standard

Perception

Prediction

Planning

Control

Monitor

Guardian

HD Map

Localization

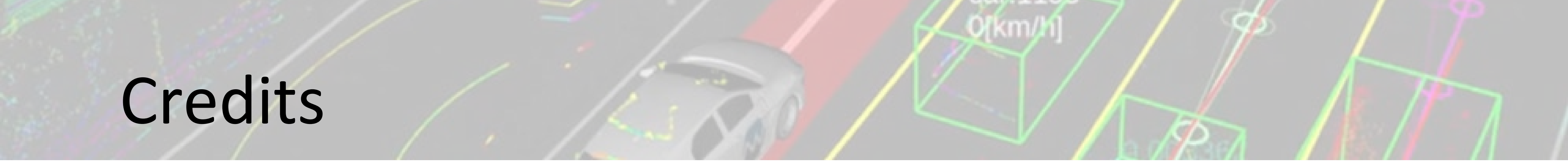
CANBus

HMI

Key: Data Lines Control lines

Autonomoums Driving

- Not everything is perfect
  - https://www.dailymotion.com/video/x154tbt



Credits

- Version 1.0, Ricardo Severino, Luis Miguel Pinho, based on
  - Autoware courseware https://www.autoware.org/training
  - NVIDIA Drive SKD, https://www.nvidia.com/en-us/self-driving-cars/drive-platform/software/
  - Apollo Autonomous Driving Solution, https://github.com/ApolloAuto/apollo
  - Pendleton, S.D.; Andersen, H.; Du, X.; Shen, X.; Meghjani, M.; Eng, Y.H.; Rus, D.; Ang, M.H. Perception, Planning, Control, and Coordination for Autonomous Vehicles. Machines 2017, 5, 6. https://doi.org/10.3390/machines5010006