AA 274A
Principles of Robot Autonomy I

Open-source Automated Driving Stack „Autoware“

Stanford University

ASL
Autonomous Systems Lab

Today's lecture

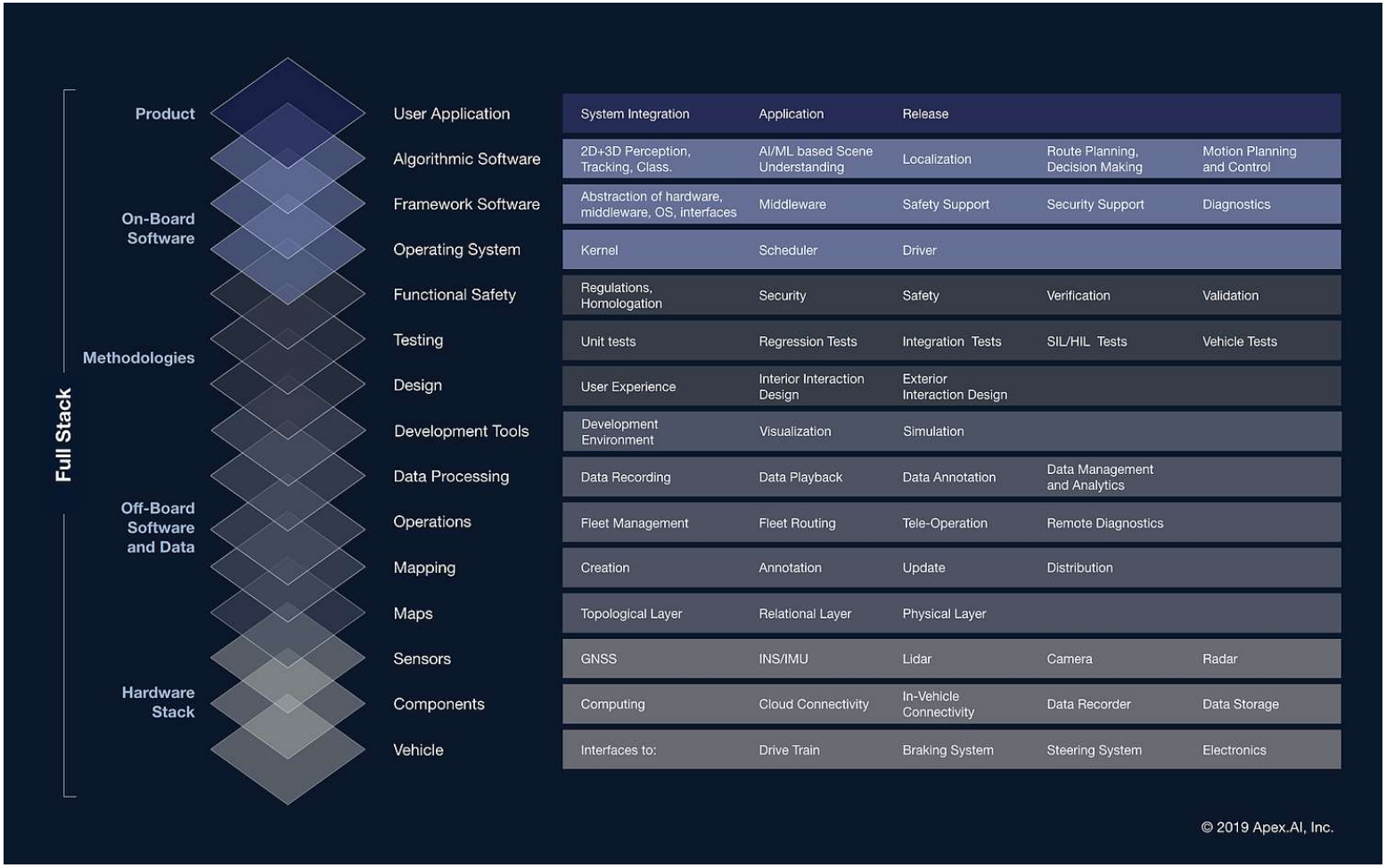
- Aim
  - Overview Autoware (Autoware.AI, Autoware.Auto), software architecture
  - Simulation environments for Autoware
  - Integration of Autoware into a research vehicle
  - Hands-on
    - Installation, development environment
    - Demos: Localization, object detection, path planning

- Readings
  - https://www.autoware.org/
  - https://gitlab.com/autowarefoundation/autoware.ai/autoware/wikis/Installation
  - https://gitlab.com/autowarefoundation/autoware.ai/autoware/wikis/home

Overview Autoware

- Other Automated driving stacks
- Autoware.AI (ROS1), Autoware.Auto (ROS2)
- Software architecture
  - General overview
  - Localization
  - Object detection
  - Path planning

Autonomous stack / Motivation



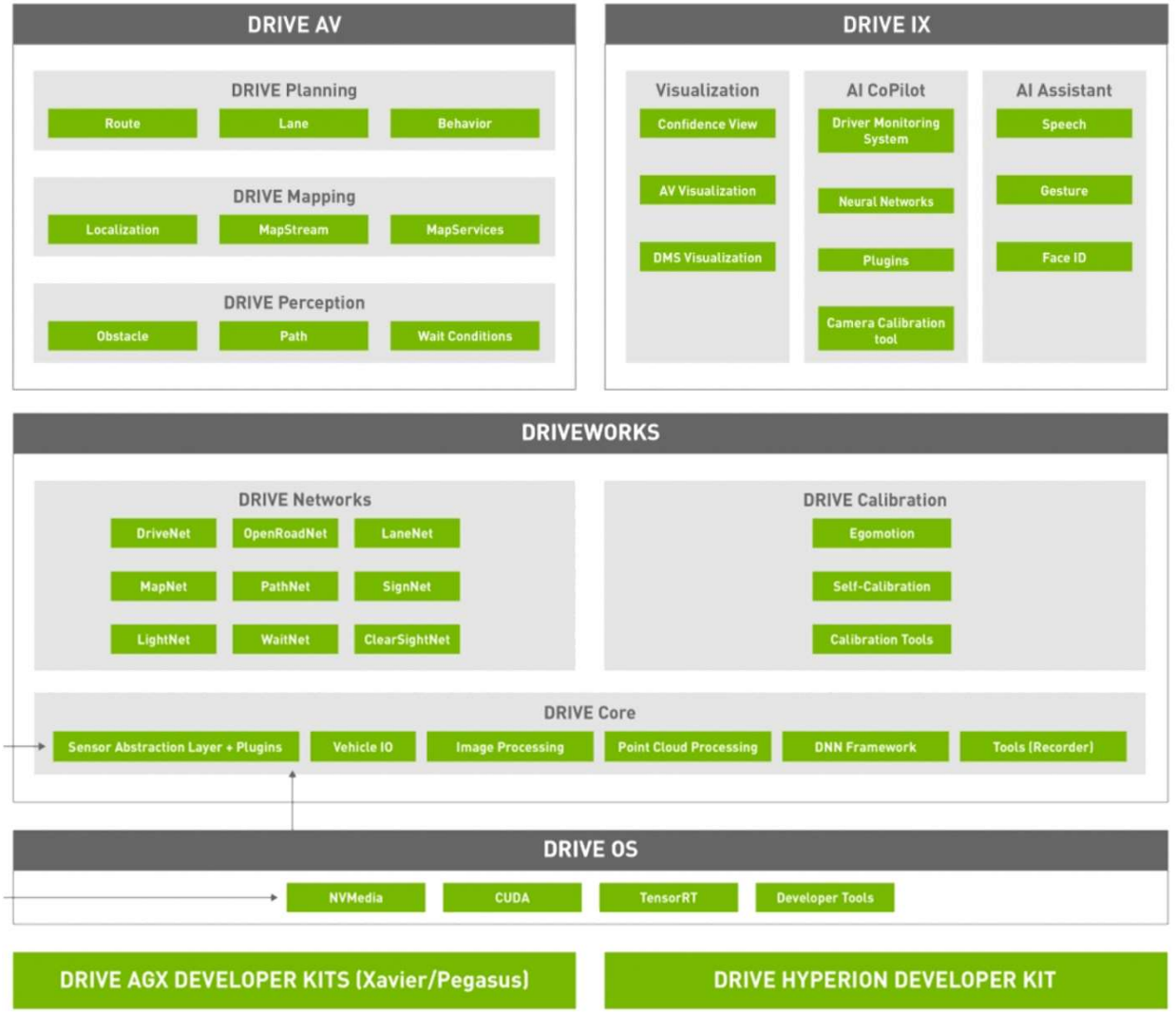
Autonomous software stack solutions

Most relevant software stacks in 2019

- DriveWorks (Nvidia)*
- Apollo*
- **Autoware**
- EB robinos & EB robinos Predictor (Elektrobit)
- OpenPilot (comma.ai)

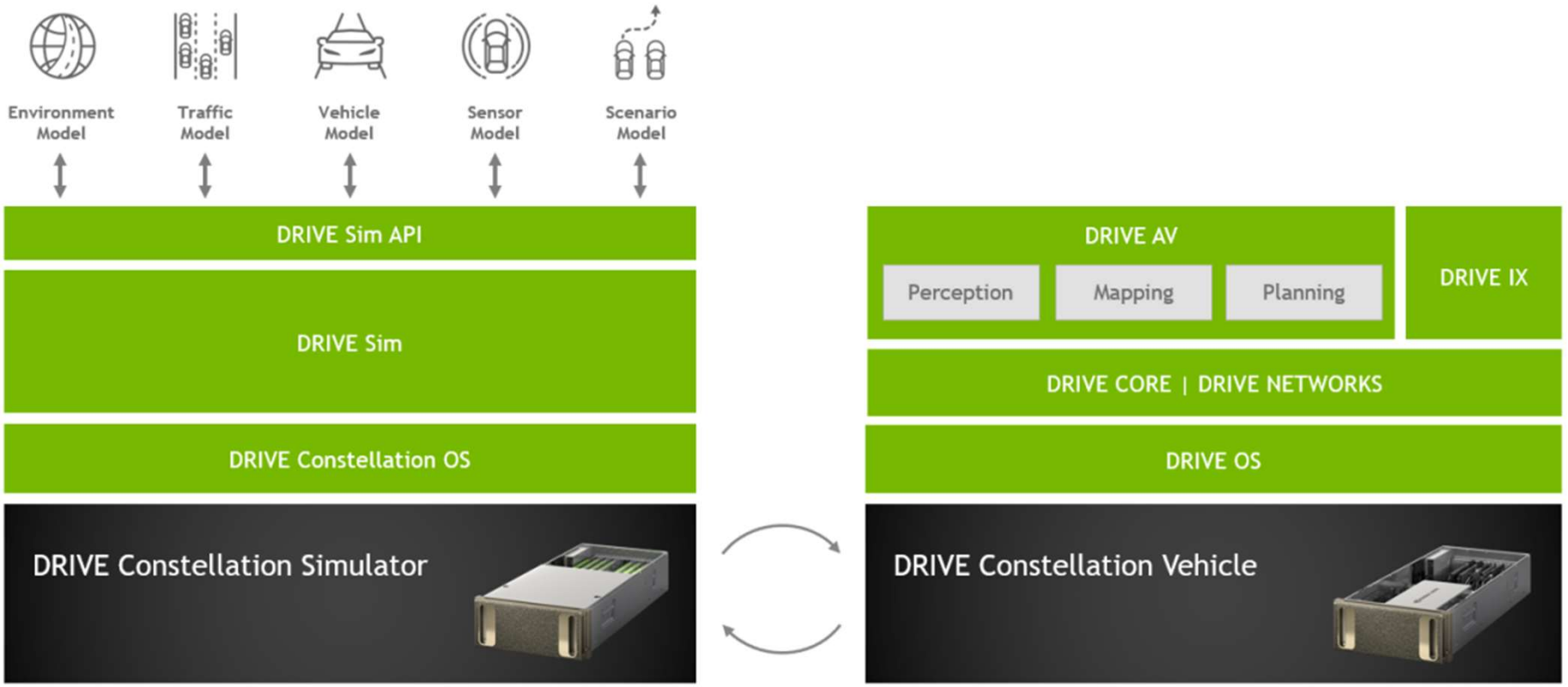
* will be briefly introduced

Nvidia DriveWorks



https://developer.nvidia.com/drive/drive-software

Nvidia Drive Constellation Architecture

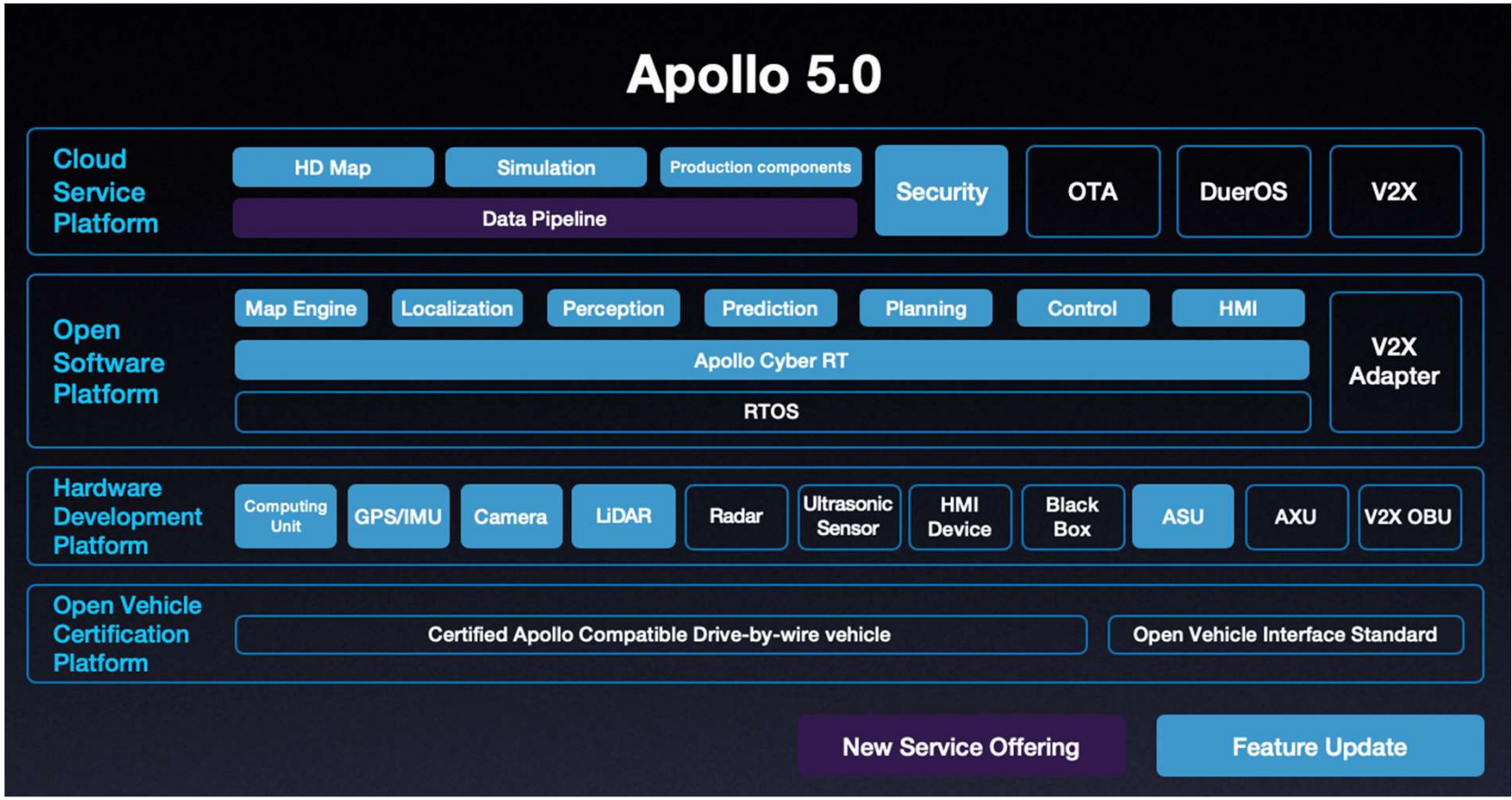


Apollo

Android of the autonomous driving industry, but more open and powerful.

Apollo / Software modules

- Data Pipeline
- Perception
- Planning
- Control
- Prediction
- Map Engine
- Simulation



Apollo in action

Autoware

- Autoware was started 2015 by Shinpei Kato at Nagoya University.
- "All-in-One" open-source software for autonomous driving technology.

- Autoware Foundation launched in 2018.
- Non-profit organization supporting open-source projects enabling self-driving mobility.

PREMIUM

Apex.AI, arm, AutoCore Intelligent Vehicle Platform, Autonomou-Tuff, Parkopedia, KALRAY, LG, Linaro 96Boards, Velodyne LiDAR, STREETDRONE, Tier IV Intelligent Vehicle, TRI-AD, eSOL, HUAWEI, MACNICA, ITDLab Intelligent Stereo Camera

INDUSTRY & GOVERNMENT

intel, robosense, SiFive, HITACHI Inspire the Next, XILINX, Brison, MANDLI COMMUNICATIONS, LeiShen Intelligent System, canoo, NEXTY Electronics, HESAI, AISAN TECHNOLOGY, TATA CONSULTANCY SERVICES, cadence, Quanta Computer, AXELL CORPORATION

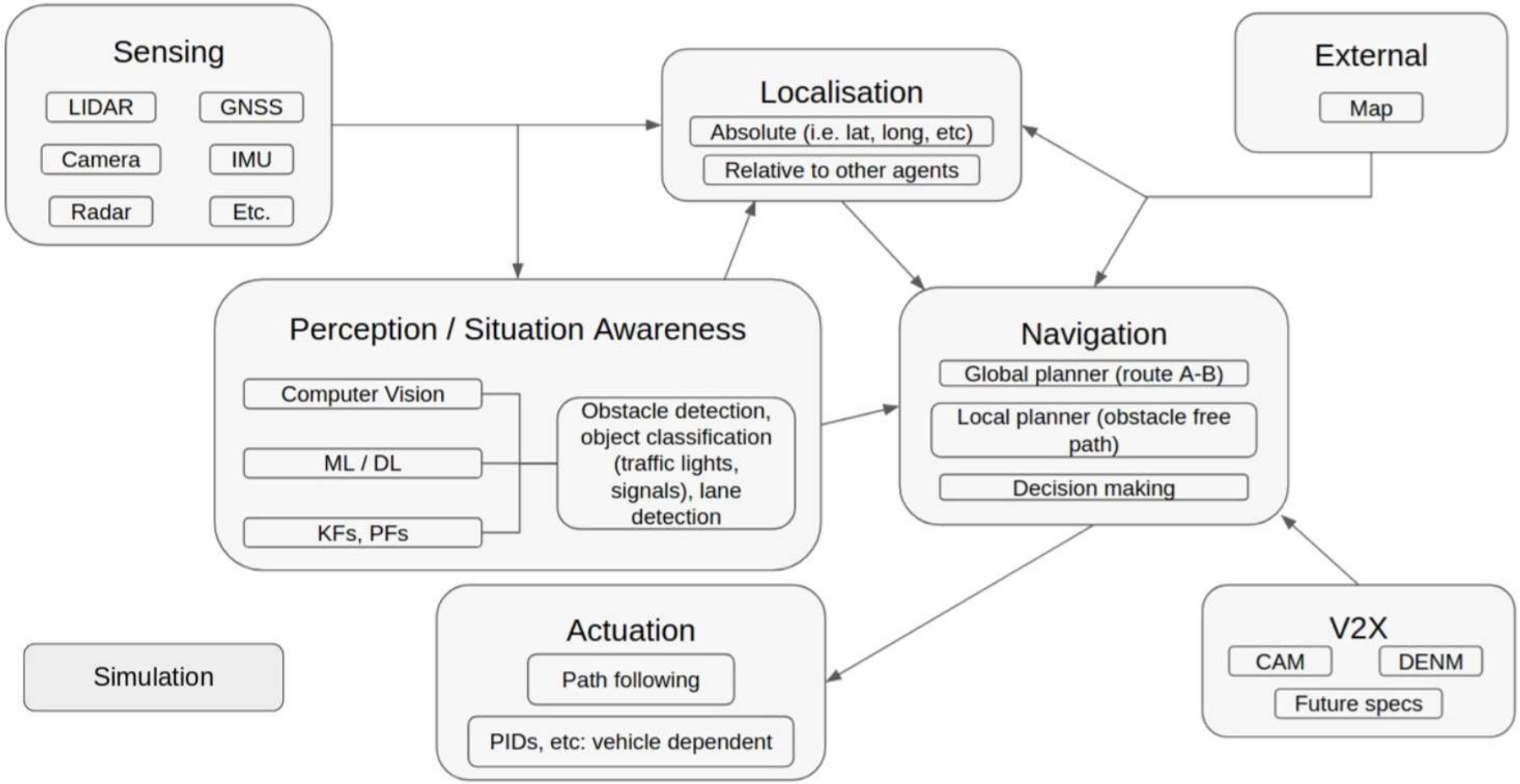
ACADEMIC & NON-PROFIT MEMBERS

NAGOYA UNIVERSITY, open robotics, semi, utbm, ITRI Industrial Technology Research Institute, Saitama University 埼玉大学

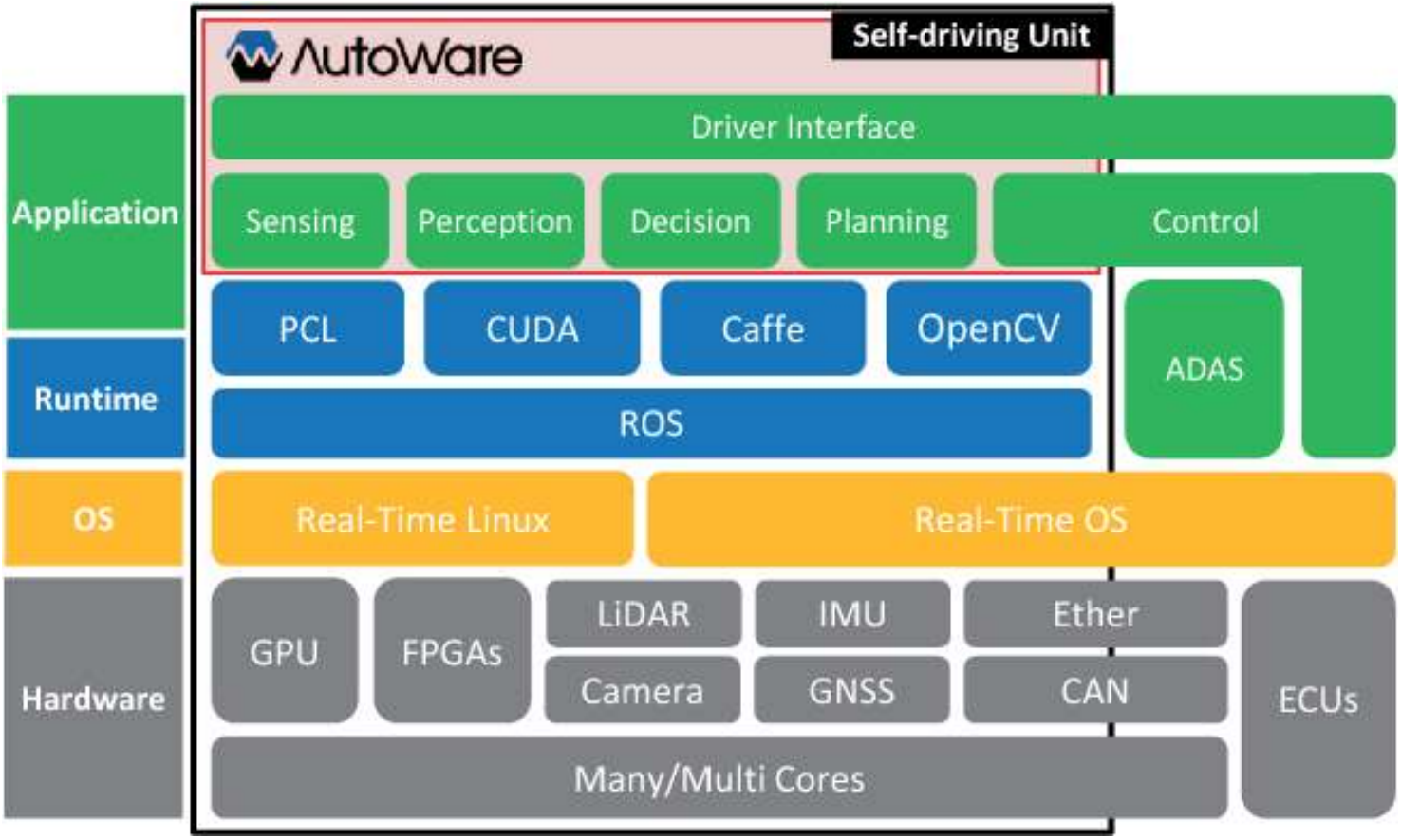
AUTOWARE.AI

- It is based on ROS 1 and available under Apache 2.0 license
- It contains the following modules:
  - **Localization** is achieved by 3D maps and SLAM algorithms in combination with GNSS and IMU sensors.
  - **Detection** uses cameras and LiDARs with sensor fusion algorithms and deep neural networks.
  - **Prediction** and **Planning** are based on probabilistic robotics and rule-based systems, partly using deep neural networks as well.
- The output of Autoware to the vehicle is a twist of velocity and angular velocity (yaw rate).

Software modules



Abstraction layers



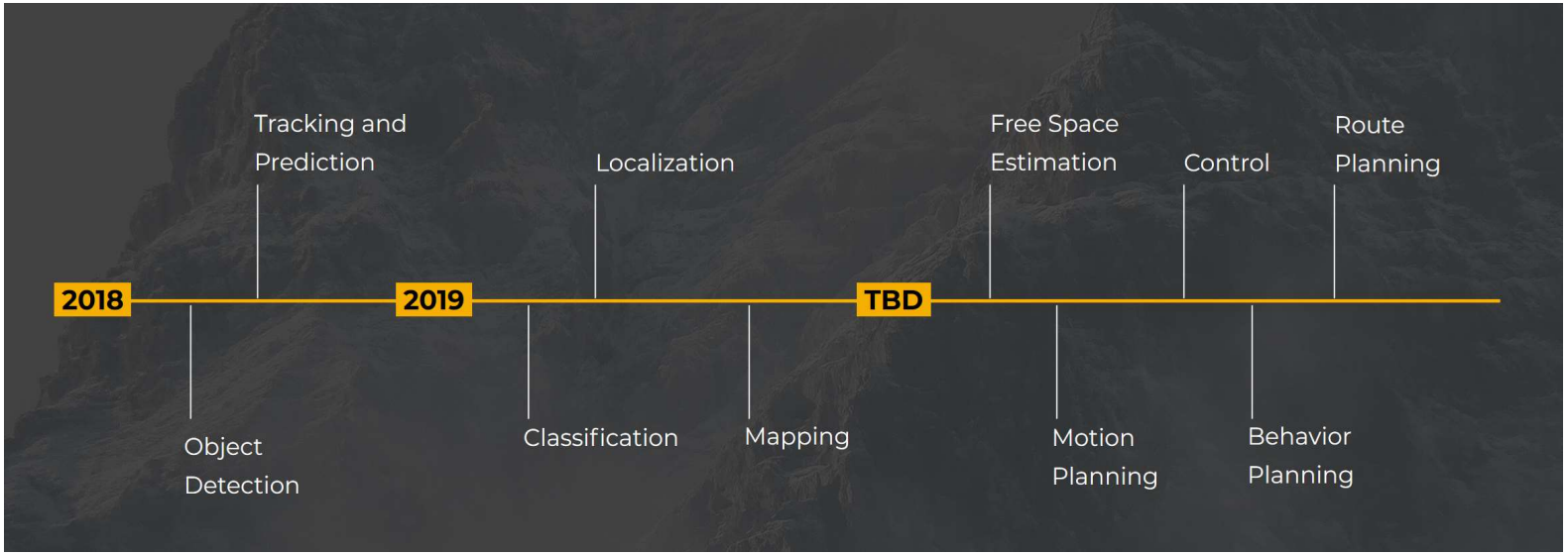
- An interface project for Autoware to be extended with proprietary software and third-party libraries in a reliable manner.
- Include device drivers for sensors, by-wire controllers for vehicles, and hardware-dependent programs for SoC boards.
- Provides a hardware reference platform with tools, unified interface design and test framework.
- Enables the integration of member company's solutions onto platforms which support the Autoware.Auto and Autoware.AI software stack.

AUTOWARE.AUTO

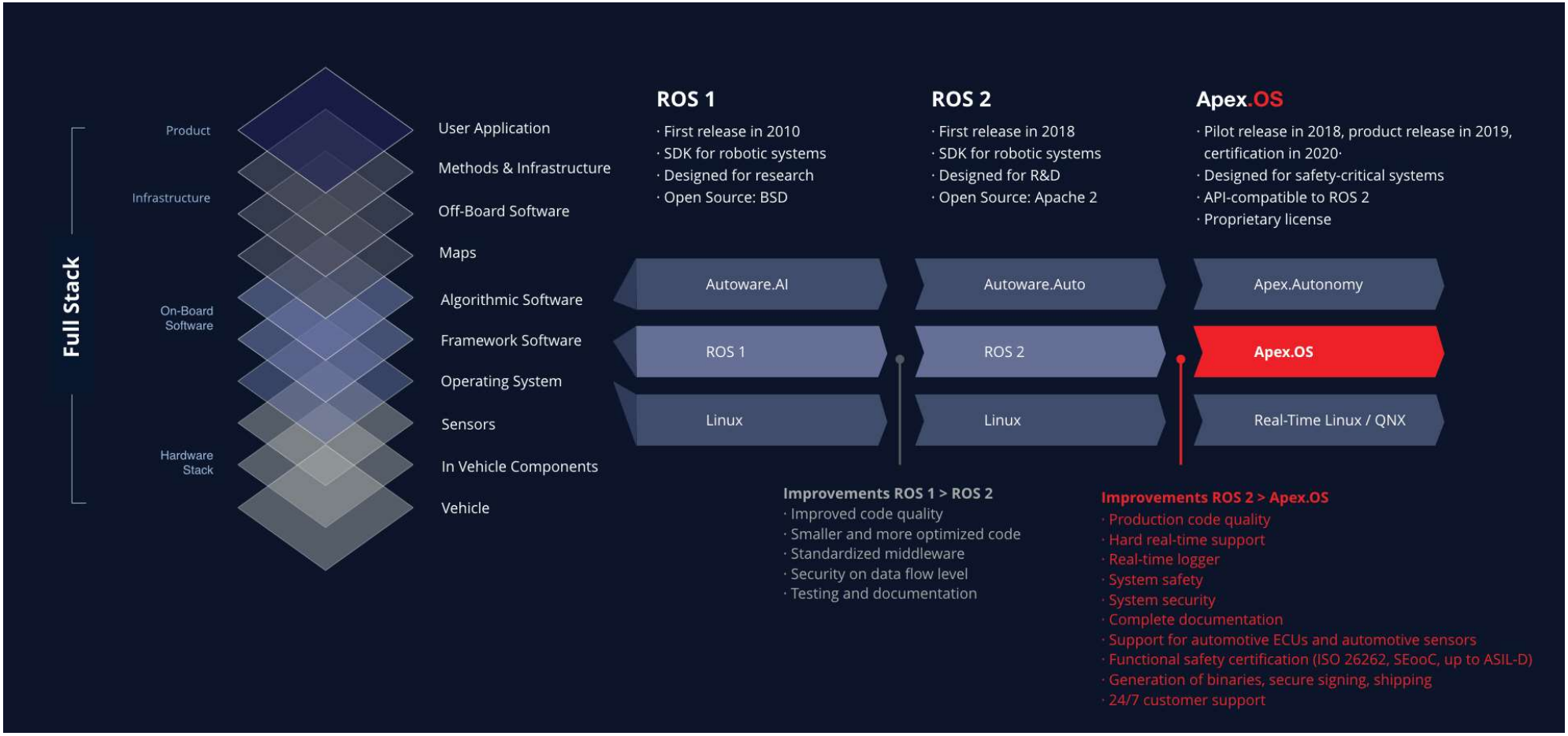
- Re-implementation of Autoware.AI
- ROS2 based
- Cleary defined APIs and interfaces for the different modules
- State of the Art development process CI/CD
  - Pull request reviews, pull request builds
  - Comprehensive documentation
  - 100% code coverage
  - Coding style guide
  - Managed by an open source community manager

AUTOWARE.AUTO

- Will initially address Autonomous Valet Parking and Autonomous Depot Maneuvering as example uses cases.
- Autoware.Auto will allow mapping of a parking lot, creation of a map for autonomous driving and autonomous driving on the parking lot.



Autoware outlook - commercial use – APEX.AI



Simulation environments for Autoware

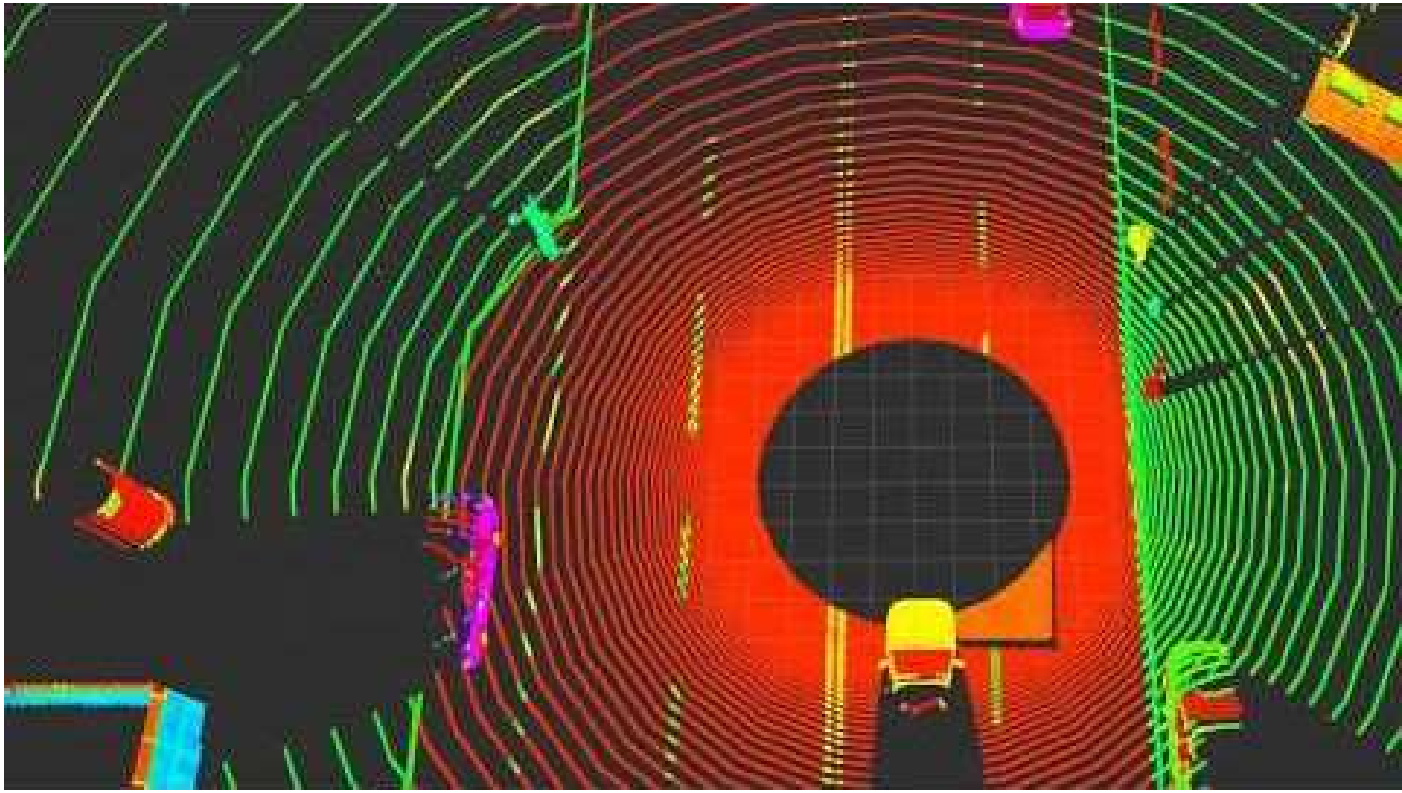
- Carla
- LGSVL
- Gazebo
- Autoware simulator

Carla

- Camera, lidar, GPS, ground truth
- Autoware, Apollo interface
- Unreal Engine
- Open Drive
- Scenario modelling
- Detailed camera model

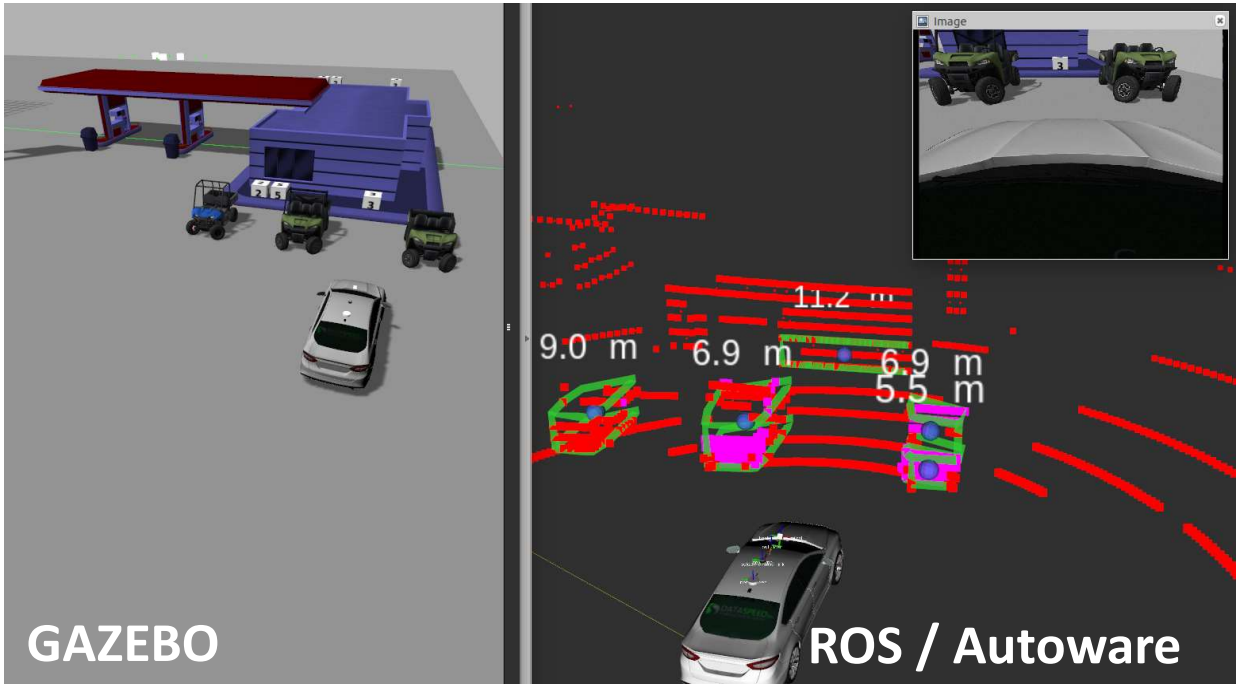
LG SVL Simulator

- Camera, lidar, GPS, ground truth
- Autoware, Apollo interface
- Unity engine
- Road editor
- Radar simulation
- GPU optimized lidar model



Gazebo

- Willow garage project, since 2012 Open Source Robotics Foundation
- ROS Interface
- Different sensor models
  - Camera
  - Lidar
- Vehicle model

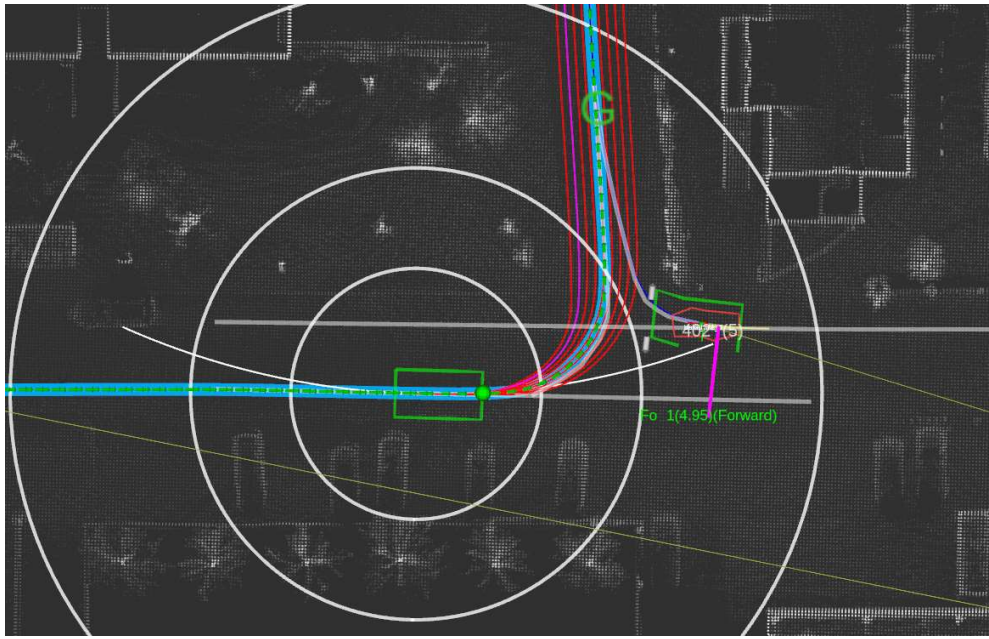


+ not automotive specific
+ large community
- custom engine
- modelling other traffic participants

Autoware Simulator

- Focus: Path Planning algorithms
- Function development only based on ground truth data
- Simulation of 5 other vehicles

- Simple vehicle models



Integration of Autoware into a research vehicle

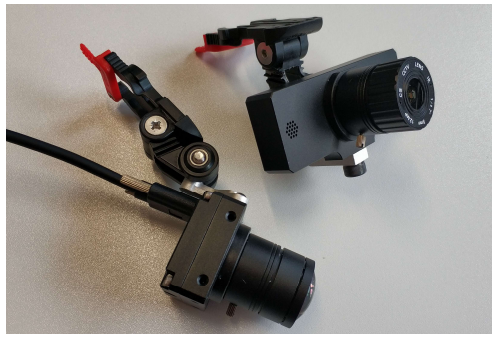
Stanford University

ASL
Autonomous Systems Lab

Integration of Autoware into a research vehicle

- Vehicle hardware components
- Mapping of road network
- Required software components from Autoware

Virtual Vehicle – Automated Drive Demonstrator



Hardware components

x86 Computer
Ubuntu 16.04 / ROS Kinetic

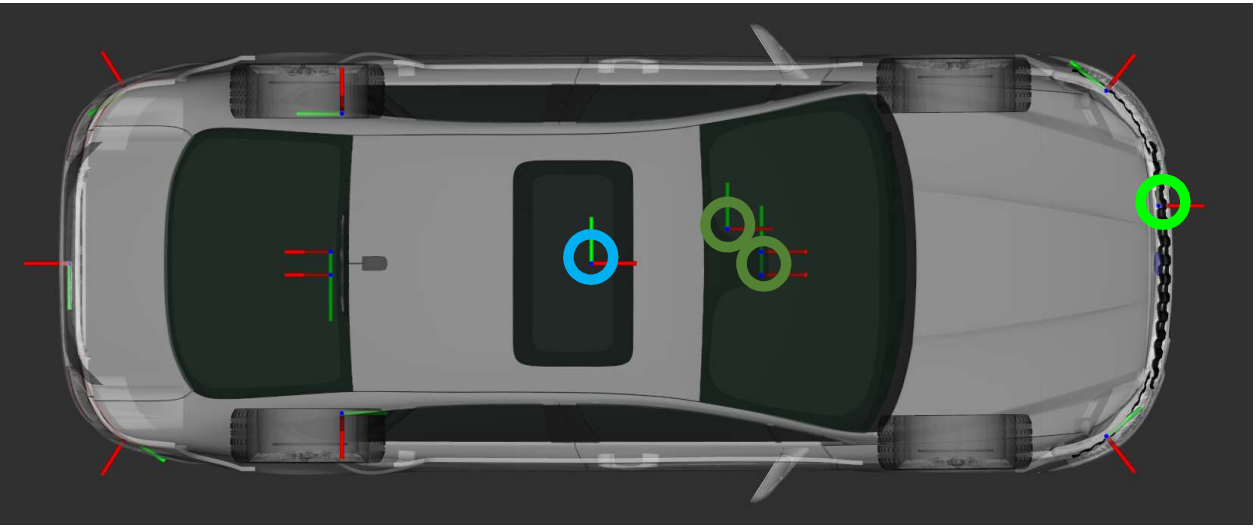
Localization:
Novatel RTK-GPS / ProPak6
Positioning < 5cm, 100Hz via TCP/IP

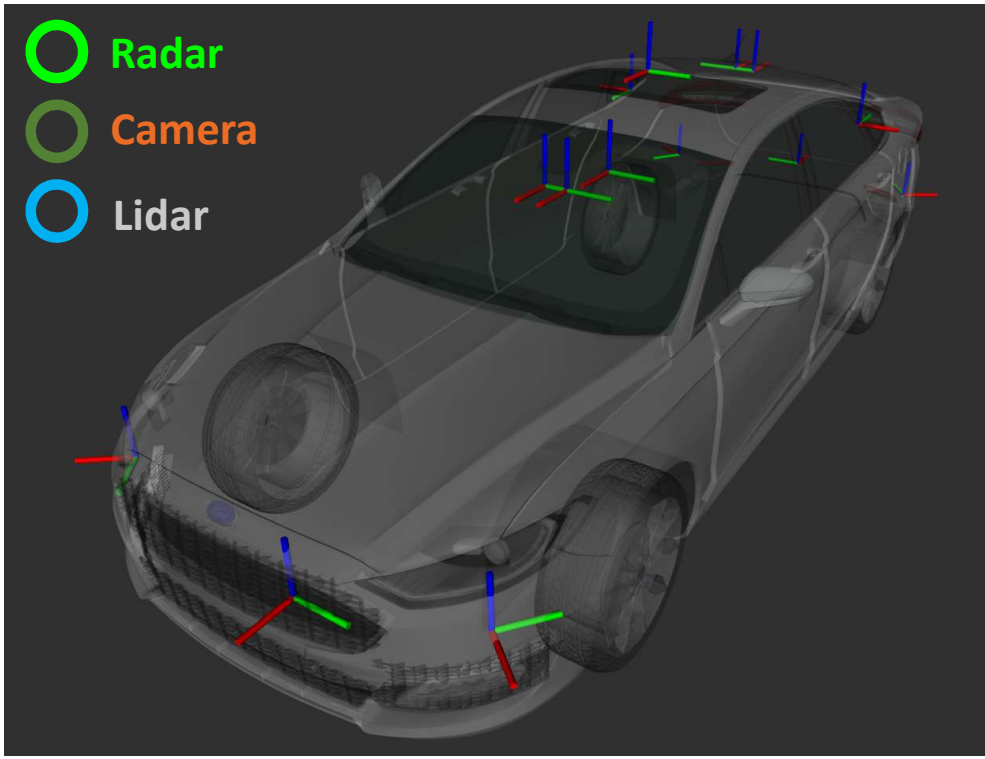
1x Long Range Radar
Continental ARS408

1x Lidar
Ouster OS1-64

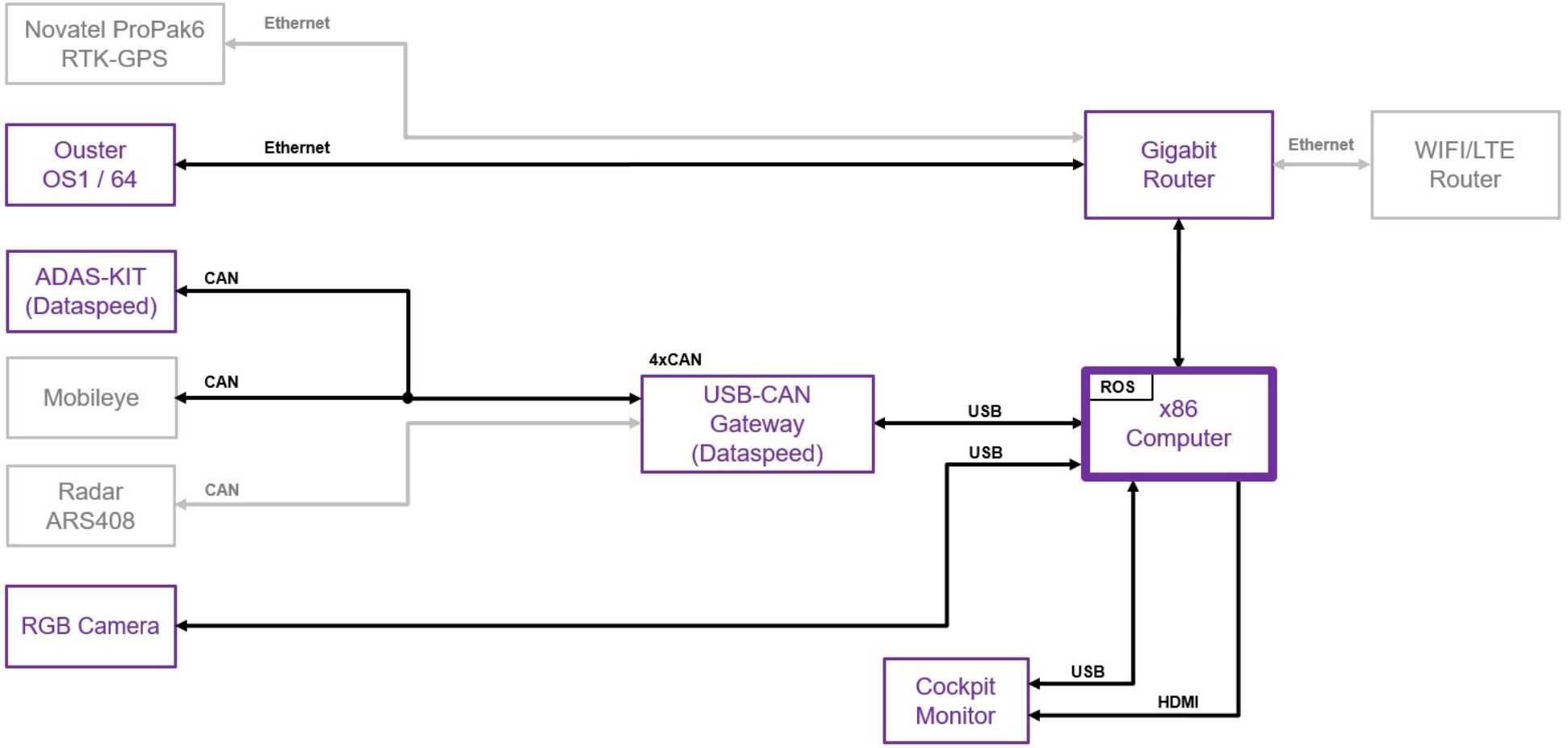
1x Mobileye

2x FLIR Cameras





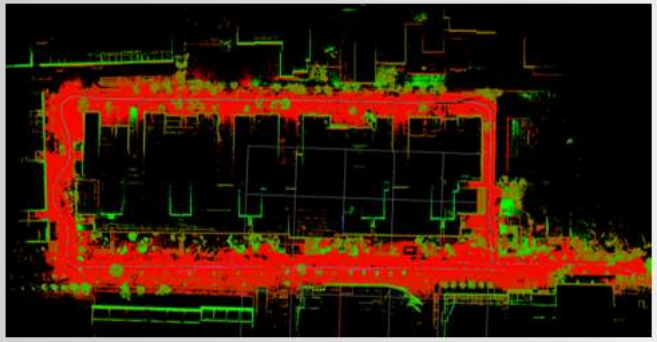
Hardware architecture



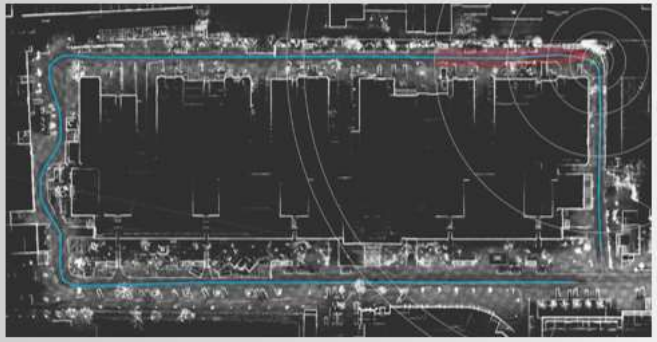
Software components

Map | Localization | Object Detection | Path Planning

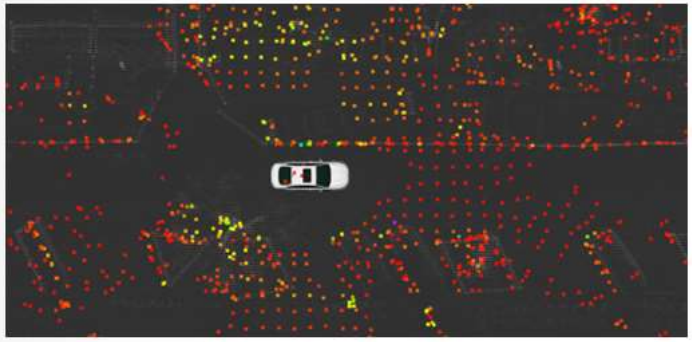
HD map creation



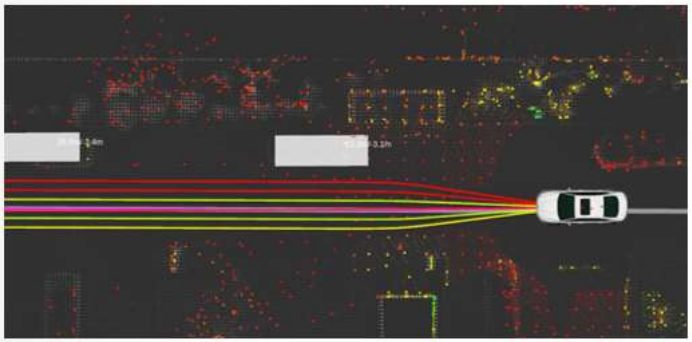
Global planning

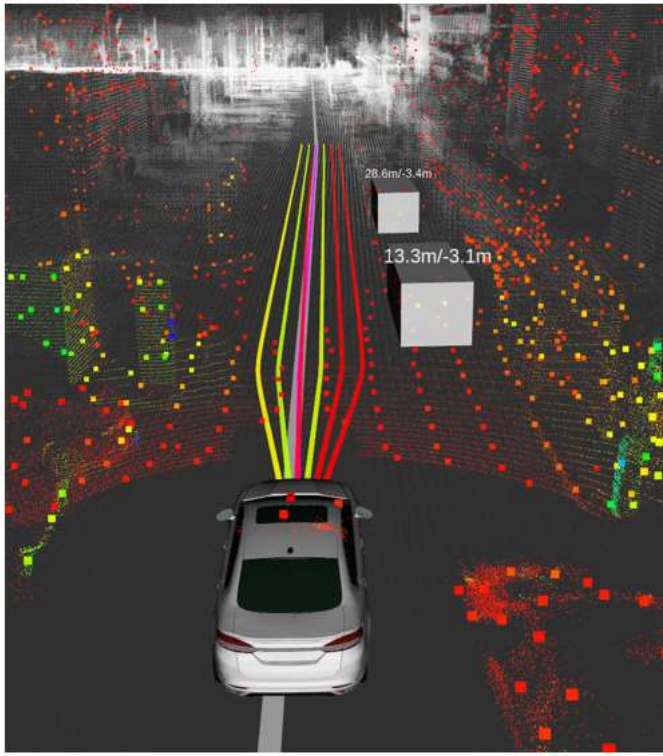


Lidar localization



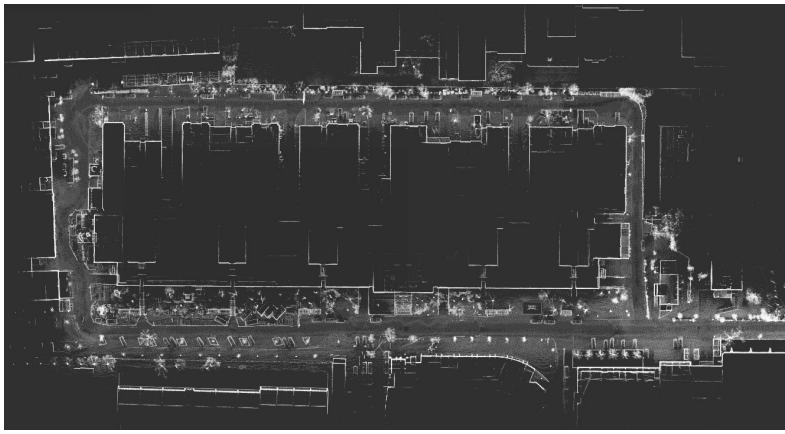
Local Planning



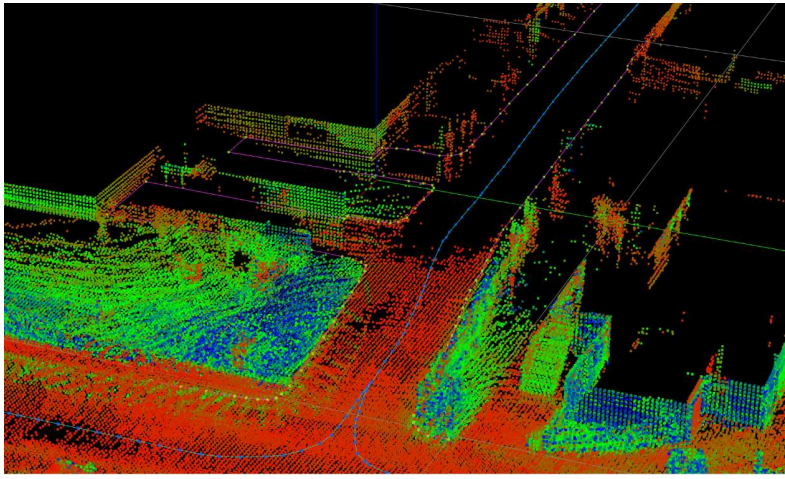


Mapping of Road Network

- Mapping for localization (without RTK-GPS)
  - Lidar with 64 layers (Ouster OS1-64)
  - NDT mapping
  - https://tools.tier4.jp/feature/vector_map_builder/

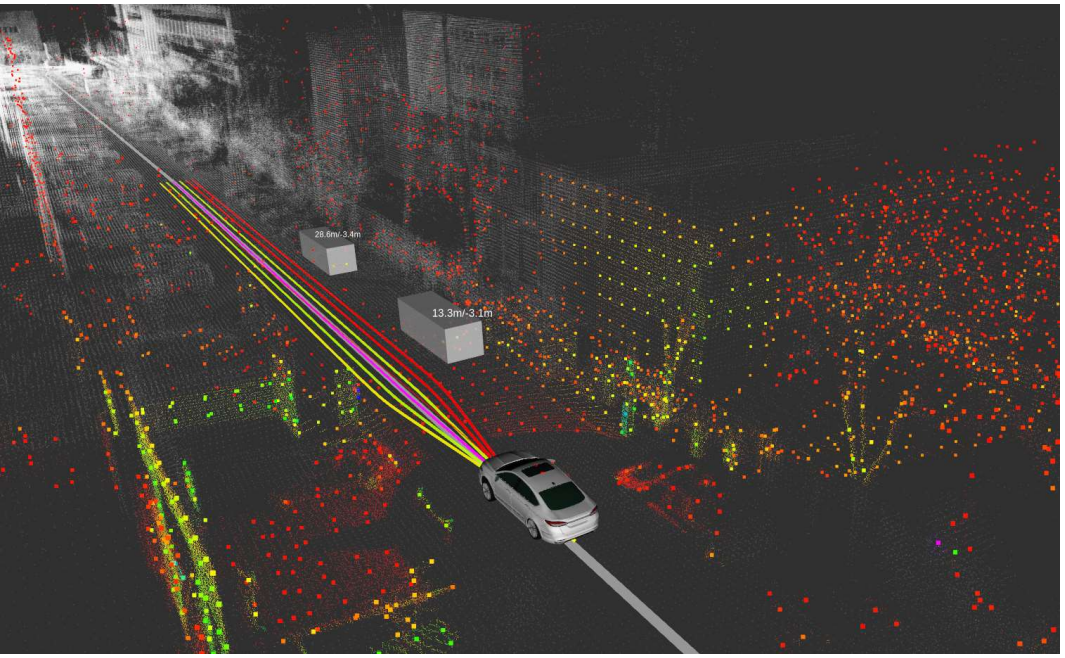


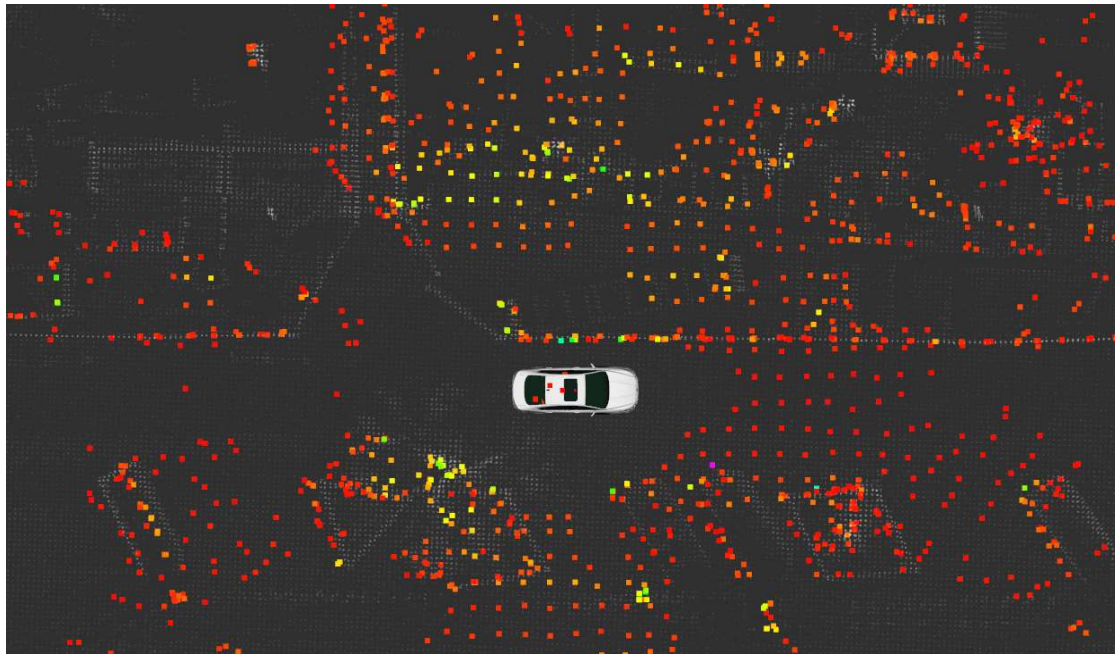
- HD map for path planning
  - Browser based tool for mapping (Tier IV)
  - Current data format: Asian Vector Map
  - Future data format: Lanelet2, OpenDrive



Localization

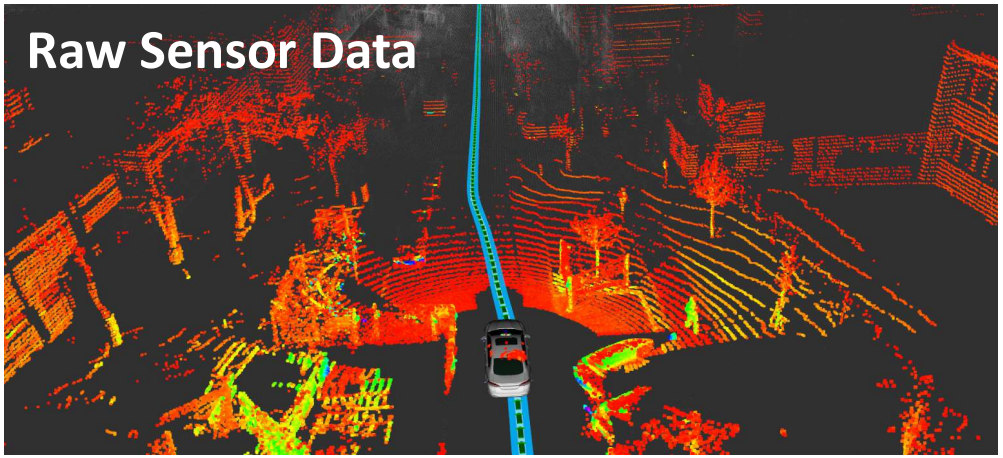
- Normal distributions transform (NDT) matching
- Lidar based / 64 Layers / 20 Hz / Voxel Grid 1m

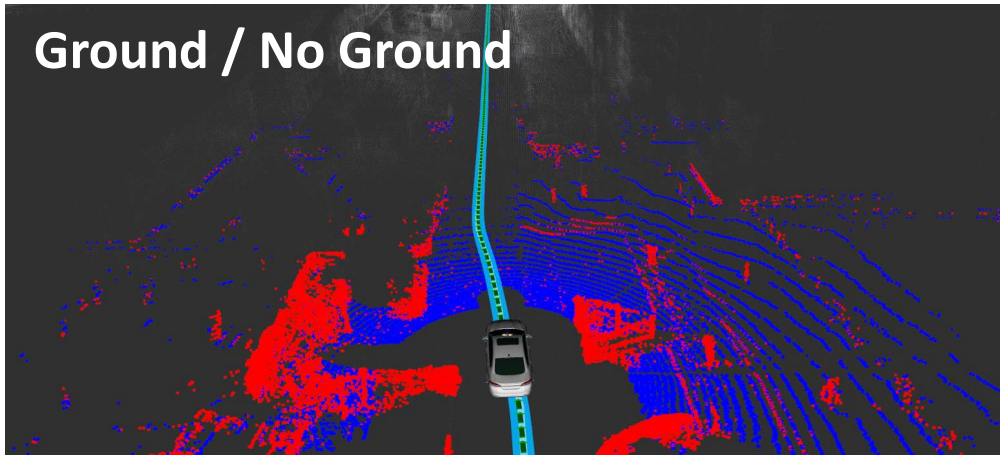


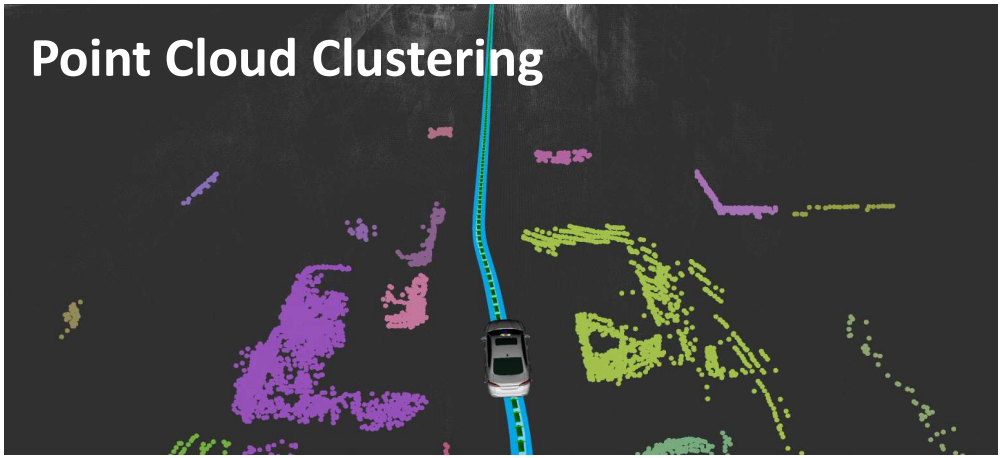


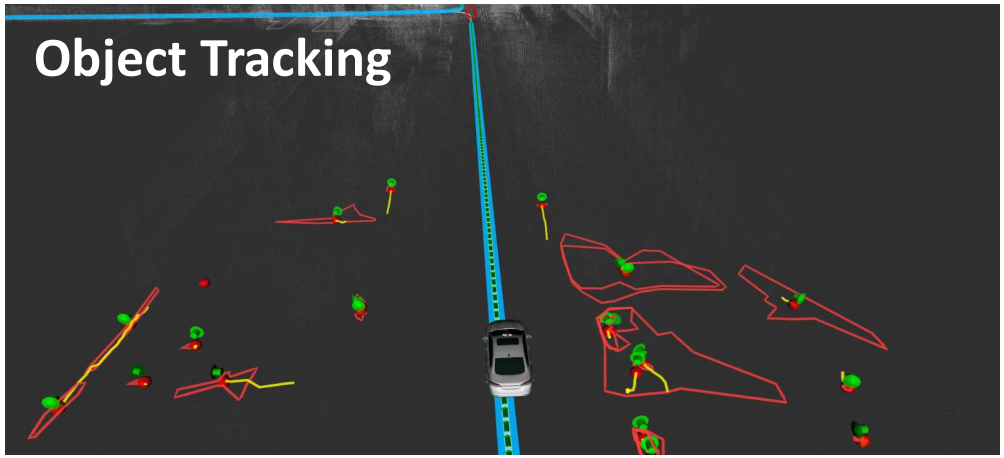
https://gitlab.com/autowarefoundation/autoware.ai/core_perception/tree/master/lidar_localizer/nodes/ndt_matching

Object Detection





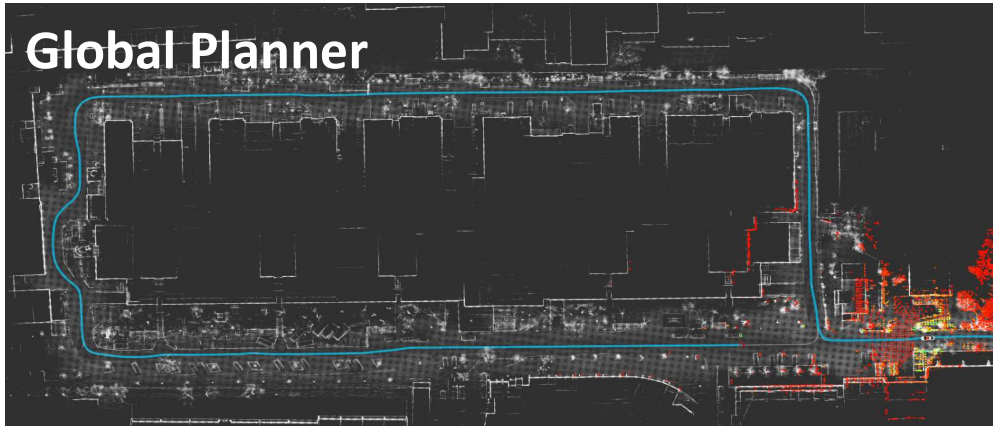


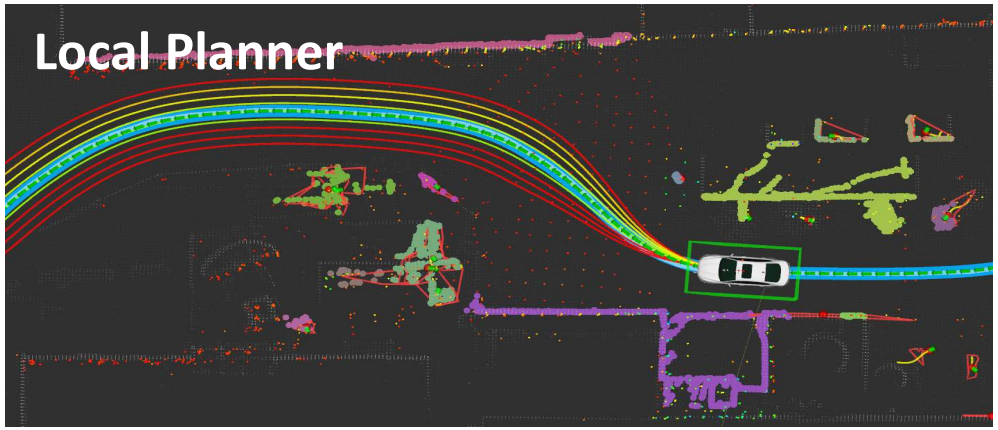


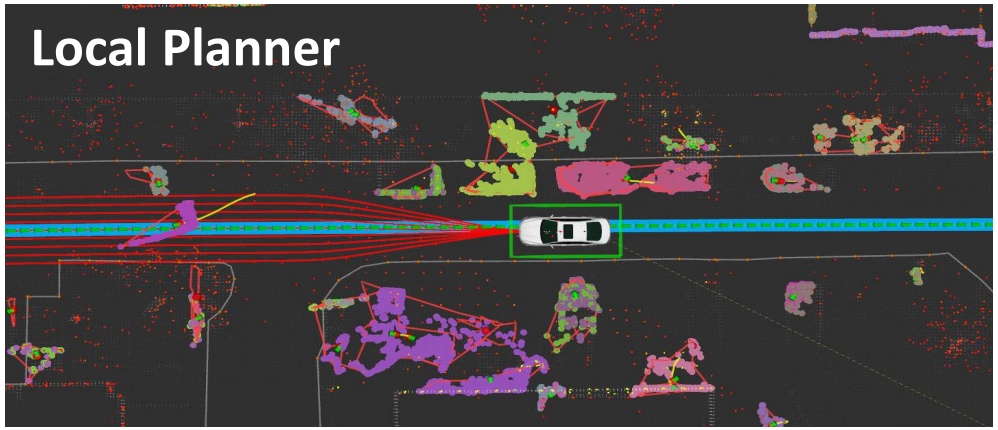
Object Detection / Classification

Full HD RGB Camera / Yolo3: Neural Network for Object Detection

Path Planning / Trajectory Tracking







Vehicle Interface

- ADAS-Kit Dataspeed Inc.
- Universal Lat/Lon Controller Interface

➢Controller for drive-by-wire interface (exectution on Dataspeed ECUs)

➢Velocity / yaw or curvature control (/twist_cmd)


autodrive-project.eu

Hands-on

- Installation on PC
- Development environment
- Autoware Demos
  - Localization (with recorded data from TU/Stanford campus)
  - Object detection (with recorded data)
  - Path planning (with Autoware simulator)

Installation

- Recommended System Specifications for complete stack
  - Number of CPU cores: 8, Nvidia GPU
  - RAM size: 32GB, Storage size: 64GB+

  Depends extremely which components are used from the stack. Runs also in a virtual machine.

- Source Build

- Docker (recommended)

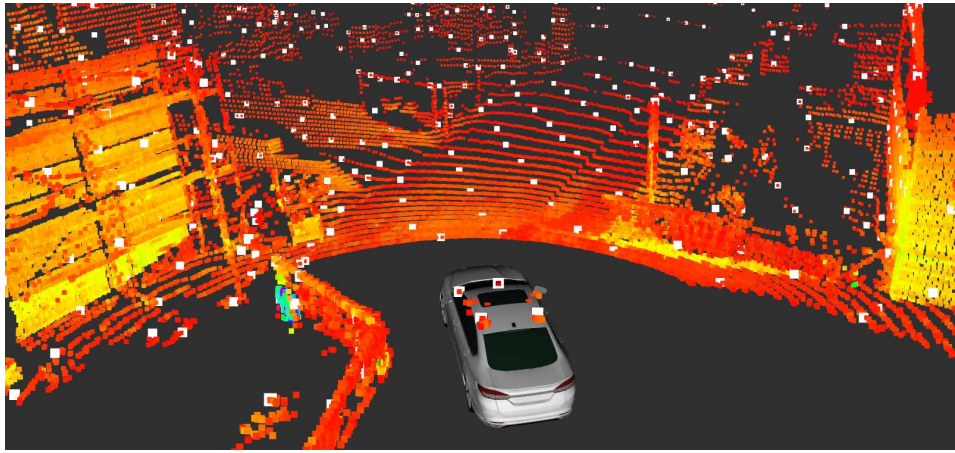
  https://gitlab.com/autowarefoundation/autoware.ai/autoware/wikis/Installation

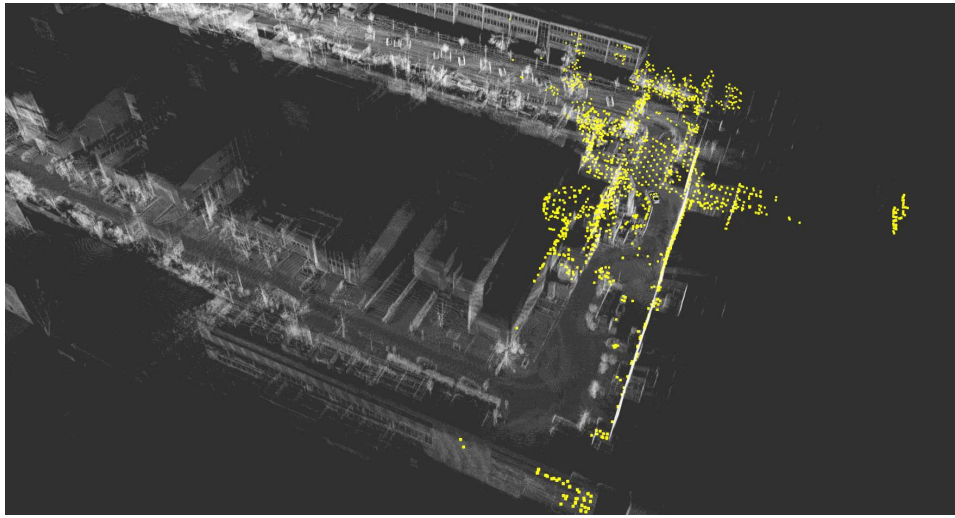
Path planning

- Global planner **[op_global_planner]**
- Local planner **[op_trajectory_generator, op_motion_predictor, op_trajectory_evaluator, op_behavior_selector]**
  - Input: /tracked_objects, /global_path
  - Output: /final_waypoints
- Trajectory Tracking **[pure_pursuit or mpc_follower, twist_filter]**
  - Input: /final_waypoints
  - Output: /twist_cmd
- Autoware Simulator **[wf_simulator]**
  - Input: /twist_cmd
  - Output: /simulated_objects

Localization pipeline

- Map loader **[points_map_loader]**
  - PCD loader from map
- Voxel Grid Filter **[voxel_grid_filter]**
  - Downsampling lidar data
  - Leaf size: 2m (60MB/s → ~1MB/s)
- Lidar based localization **[ndt_matching]**
  - NDT matching
  - Input: /filtered_points, /vehicle/twist
  - Output: /ndt_pose
- EKF Localization Fusion **[ekf_localizer]**
  - Input: /ndt_pose, /vehicle/twist
  - Output: /ekf_pose_with_covariance

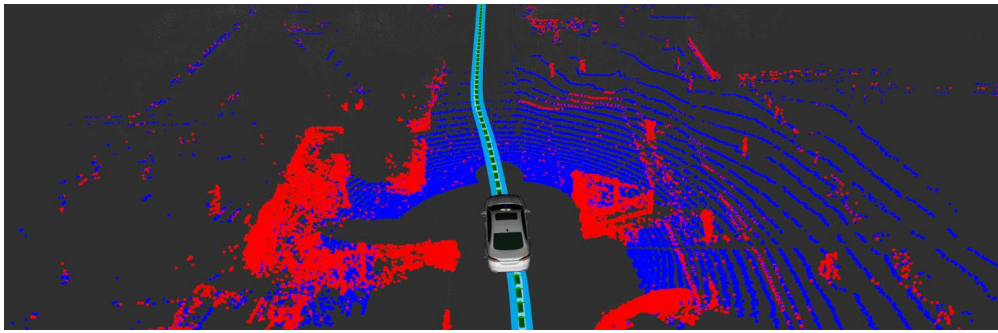


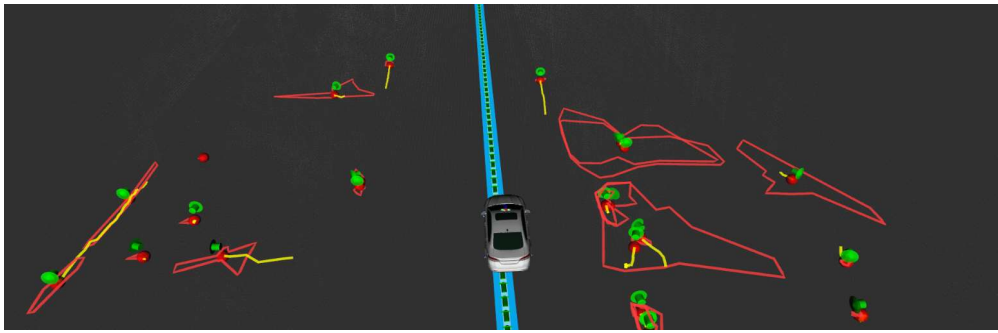


Localization / Autonomous Racing Graz

Object detection

- Ray ground filter **[ray_ground_filter]**
  - Separation ground / no ground
- Point cloud clustering **[lidar_euclidean_cluster_detect]**
  - NDT matching
  - Input: /points_no_ground
  - Output: /points_cluster
- Cluster tracker **[lidar_kf_contour_track]**
  - Input: /points_cluster
  - Output: /tracked_objects





Thanks for your attention!
Questions?

Daniel Watzenig, Markus Schratter
daniel.watzenig@v2c2.at
markus.schratter@v2c2.at

Stanford University

ASL
Autonomous Systems Lab