# Introduction to Autoware.Auto for AI Challenge Competitors

Ryohsuke Mitsudome

October 6st, 2021

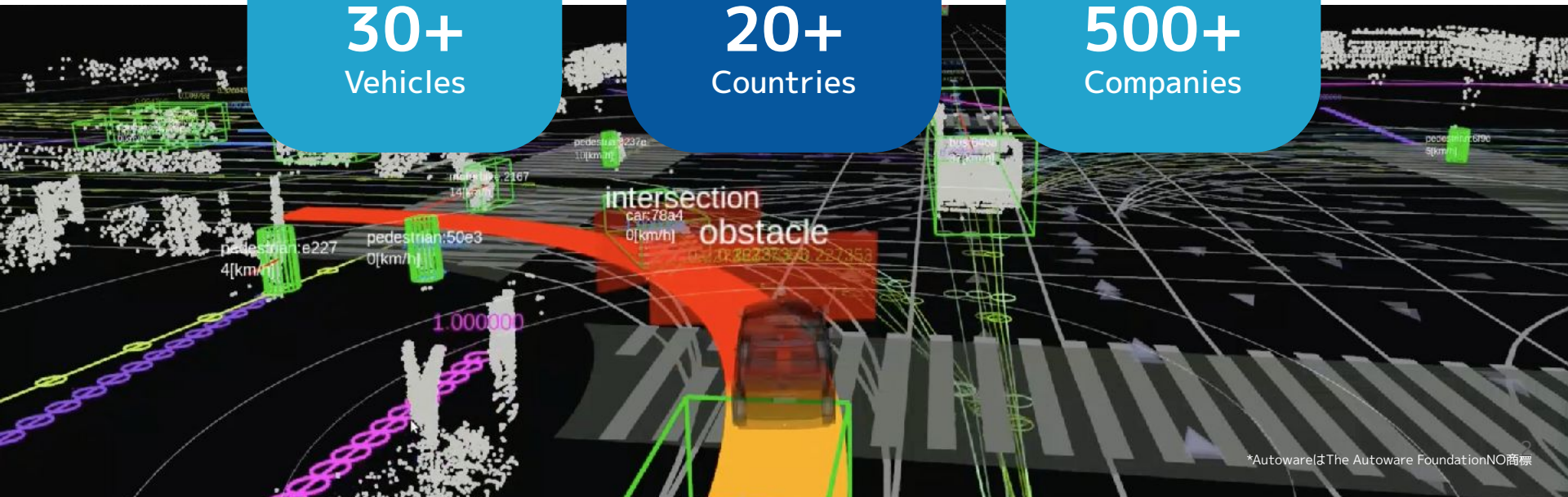
# What is Autoware?

The world’s first open-source autonomous driving software

**30+**
Vehicles

**20+**
Countries

**500+**
Companies




*AutowareはThe Autoware FoundationNO商標

# Autoware.AI and Autoware.Auto


Started in 2015

Based on ROS1

Used mainly for PoC & Research

More than 130 packages


Started in 2018

Based on ROS2

Following best engineering practices

More than 90% Test Coverage

# Autoware High Level Roadmap

2015-19 | 2020 | 2021 | 2022 | . . . | 2025*

Autoware.AI

EOL in 2022

**Campus/Bus***
Autoware.AI is being used in many Private Area Shuttle/Bus implementations
* Not AWF defined ODDs

Autoware.Auto V2.0 demo and development expect by end of 2021

**Cargo Delivery**

Autoware.Auto

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

Next ODD ?

**MaaS/Robotaxi**
Ultimate goal of Autoware.Auto is to achieve L4 AD in dense urban environments
* Timeframe is estimated

# Autoware.Auto

- Not a “ported” version of Autoware.AI, but a “rewrite” with best-in-class software engineering practices
- Autoware in ROS2
- Currently using Foxy ( https://docs.ros.org/en/foxy/index.html )

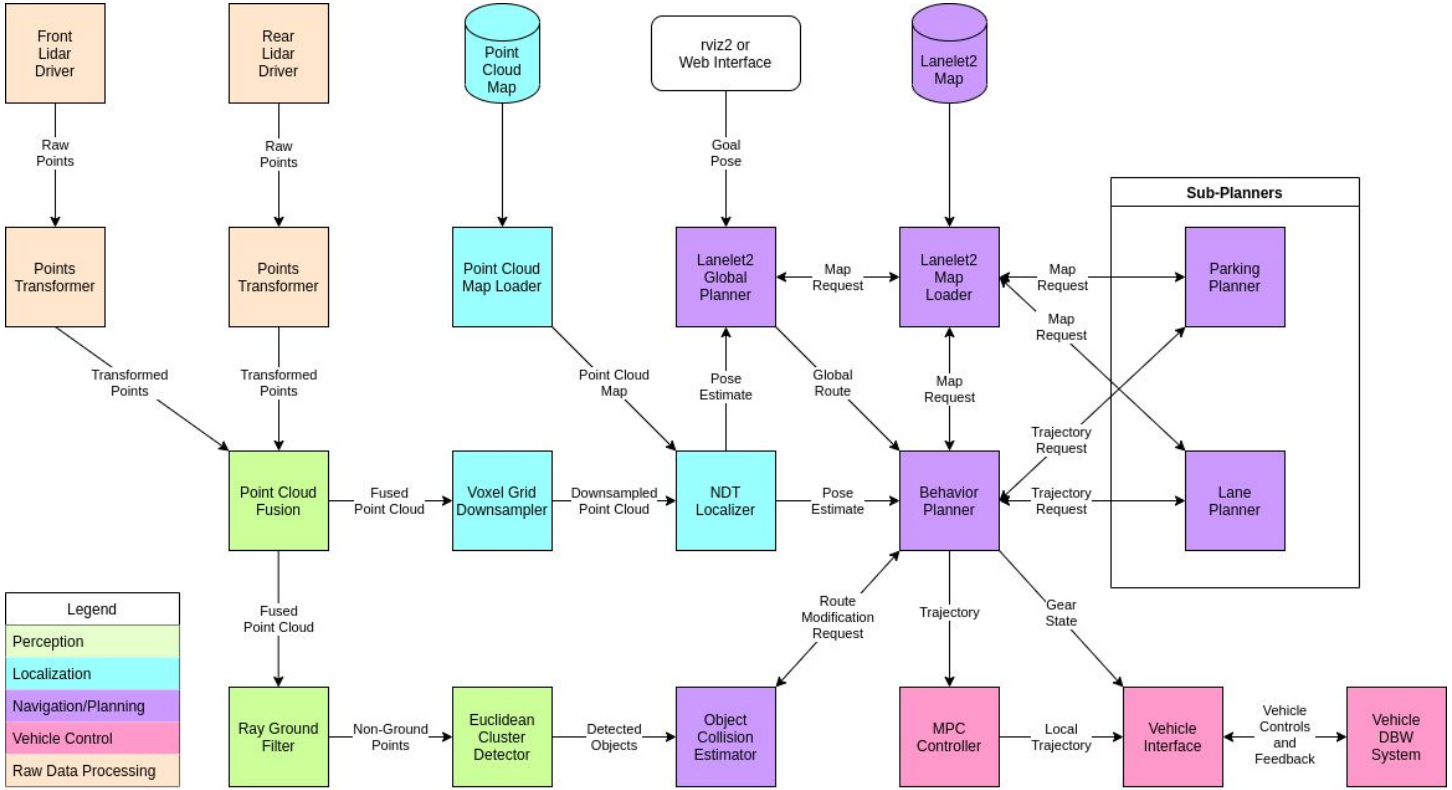
# Getting Started with Autoware.Auto

**Source Code** **Readme**

- Autoware.Auto Installation link
  - Recommended to follow ade installation
- Setup LGSVL Simulator link
- Run AVP Demo link
  - AVP Demo launch captures current capabilities of Autoware.Auto pretty well
  - You may also check individual features by following “General Demos” in the Usage Page for better understanding
  - You may want to checkout v1.0.0 during installation

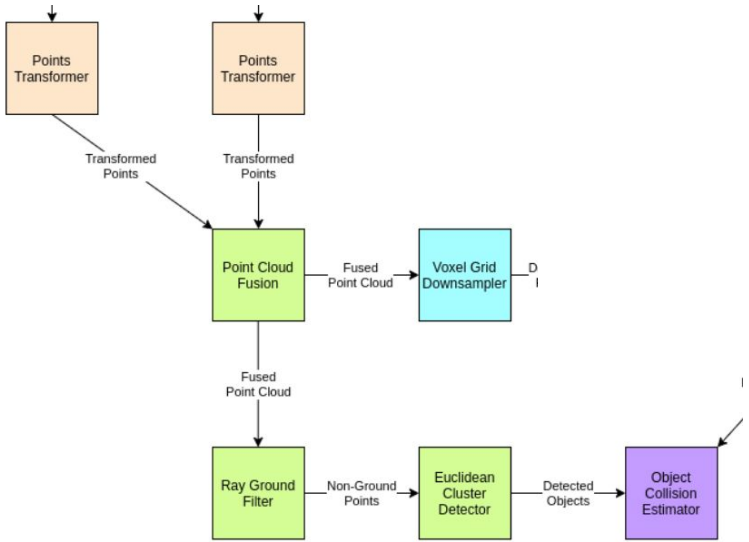
# AVP Architecture Walkthrough & Tips for AI Challenge

# Architecture Diagram of AVP Demo



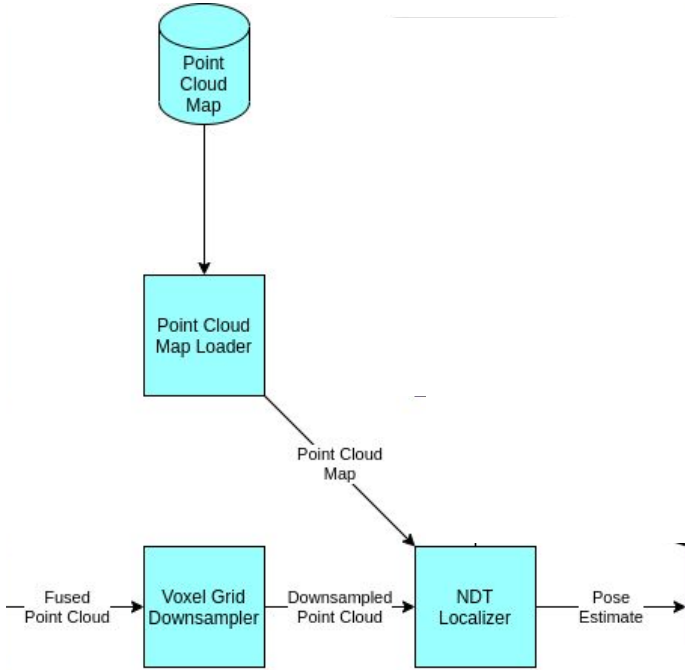
# AVP Demo - Perception Stack

- Only uses pointcloud for detecting obstacles
- Very basic ground removal + euclidean clustering
- New image + lidar detection pipeline is under development
  - tracker
  - DNN image detection
  - DNN pointcloud detection



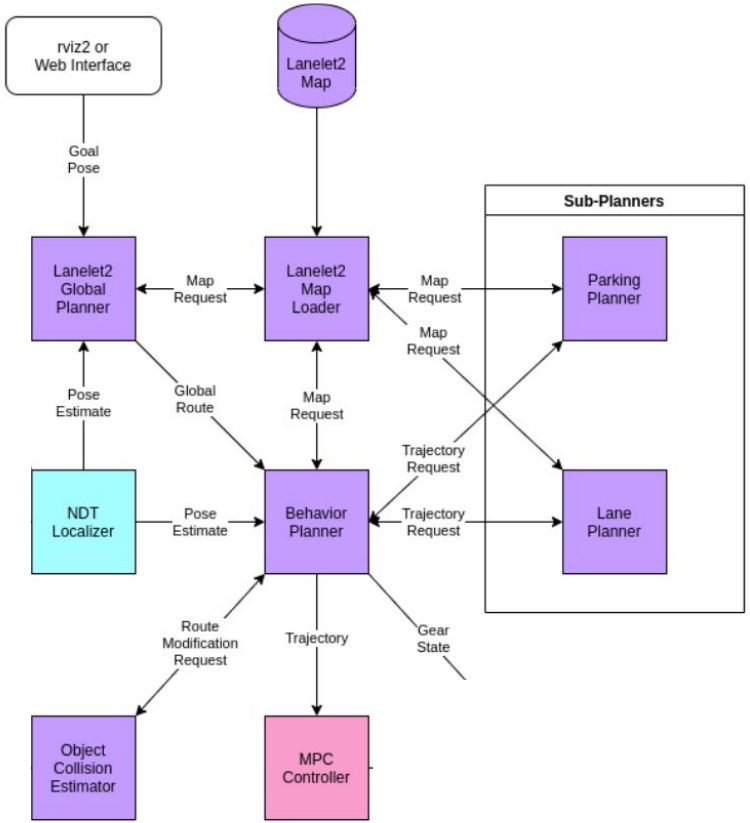
# AVP Demo - Localization Stack

- AVP Demo only uses NDT Matching for localization
- Localization using state estimator (ekf) is under development to fuse odometry/IMU/lidar/GNSS inputs.



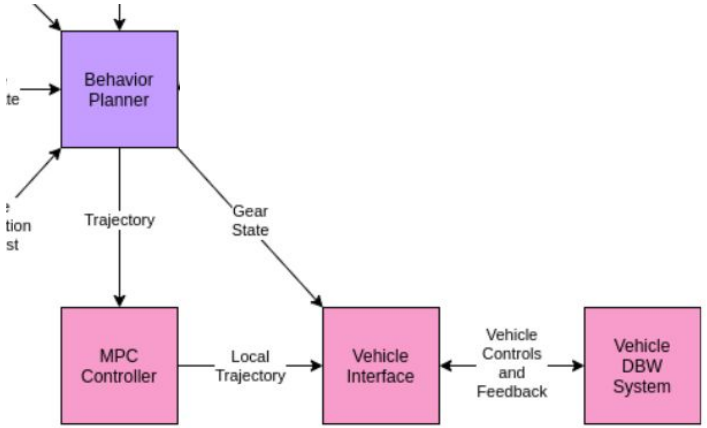
# AVP Demo - Planning Stack

- Receives goal, localization output, and perception output to generate trajectory
- Assumes to drive center of the lane
- It can stop before hitting obstacles, but no replanning of trajectory for obstacle avoidance



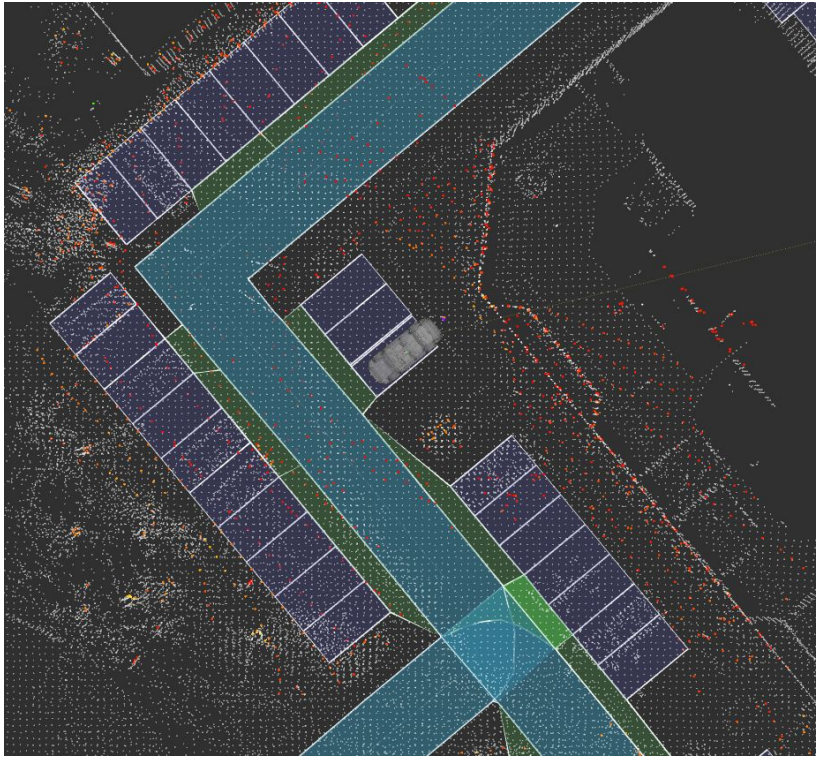
# AVP Demo - Control Stack

- Consumes trajectory from the planner and outputs control command to the vehicle interface (lgsvl_interface when simulation)
- Currently supports MPC and Pure Pursuit
  - MPC higher performance but difficult to tuning (new implementation is under development for easier tuning)
  - Pure_pursuit has easier tuning but less accuracy
- Not designed for racing
  - Motion model, calculation cost, etc



# AVP Demo - Maps

- PointCloud map
  - Geometry information in PCD format
  - Used for localization
- Lanelet2 Map
  - Contains lane information of the environment
  - See here for the details
  - For C++ API, check the official examples or checkout some of the helper functions in Autoware.Auto (e.g., had_map_utils)



# Available Tutorials/Videos

- Self-Driving Cars with ROS and Autoware
  - Online course to learn about ROS and Autoware
- ROS WORLD 2020: AUTOWARE PARALLEL SESSION
  - Parallel session presented by the core developers involved in AVP Demo project.