What are all these repos?

Our organization has more than 50 repos in it currently. It can get confusing what each of these repos does and where to look. In general the repos in our stack can be grouped into the following.

Meta

- art: Meta repository used to stage Github discussions

The Core Autonomy System

- race_common : ROS2 autonomy code written by the ART team for racing. Contains packages like our behavior planning, path planning and control packages. Launch files and parameters are also in here.

Simulation

- simulator : Private LGSVL fork, supporting Vehicle Physics Pro plugin for better vehicle dynamics.

Autoware Stack

- autoware.universe: Autoware experimental packages in all stages of urban driving
- autoware_common: Autoware common packages/libraries used by multiple packages in it's ecosystem
- autoware_msgs: ROS2 message definitions for the entire `autoware` stack
- autoware.core: Autoware's high quality stable packages
- autoware_auto_msgs: Old ROS2 message definitions of the Autowar which is still being used in the current `autoware` while they are porti

- tier4_autoware_msgs: Tier4 version of Autoware messages used by Tier4 (current maintainers of `autoware`) to develop packages internally before porting to the `autoware` stack

Autoware Dependencies

- grid_map: C++ library with ROS2 interface to manage two-dimensional grid maps with multiple data layers
- muSSP: muSSP (Minimum-update Successive Shortest Path) is an exact and efficient min-cost flow (MCF) solver for the global data association problem in multi-object tracking (MOT)
- ndt_omp: This package provides an OpenMP-boosted Normal Distributions Transform (and GICP) algorithm derived from PCL

Interfaces

- raptor_race_interface: An interface to convert vehicle odometry and other such DBW feedback to `race_msgs`. Specific to the Raptor ECU driver

Utils

- camera-utils: Utils for processing images written by Andrew Saba. Includes tools for compression to reduce ros2 bag sizes and debayering
- car_heading_generator: ROS2 package that takes a front and back GPS unit and derives heading in the ENU frame (used for RTK GNSS systems that don't support dual antenna)

External packages/tools

- image_pointcloud_detector: Private import of repository written by Michael Wu that was responsible for projecting 2D Camera detections onto a 3D Lidar space. Archived in favor of `obstacle_detector` which is a fork of Michael Wu's rewrite of this project
- obstacle_detector: Public fork of Michael's ROS2 package responsible for projecting 2D camera detections onto 3D Lidar data through the use of camera intrinsics and lidar-camera extrinsics

- SFA3D: Lidar 3D Deep Learning Detection Neural Network. Was being worked on until Autoware released their Centerpoint models which are now being used
- ros2_aruco: Private import of ROS2 Aruco Marker detection that was written when the evGrandPrix was using it (abandoned as teams determined that April Tag performance on moving cars was questionable for detections)
- offline-trajectory-tools: Public fork of Haoru Xue's offline TTL generation tool that is used to generate the TTLs currently being used in our stack
- raptor_ros2_generator: Public fork of Siddharth Saha's Raptor ECU ROS2 generator tool that reads in a `.dbc` file and a `.yaml` file to auto generate ROS2 Raptor ECU Drivers (used in evGrandPrix and Indy Autonomous Challenge)

Hardware Drivers

The remaining packages you will see are likely forks of ROS2 drivers for sensors or drivers we have written ourselves

External Repositories

There are some noteworthy external repositories that we don't fork in our organization but are good references

- IACBaseSoftware: Accessing IAC documentations and their versions of the drivers (authorization required).
- art-core: Accessing previous IAC autonomy stack (authorization required).
- OSSDC Simulator: Current efforts to make SVL last even after their sunset

Reach out to a technical lead if you want access to any of these external repos