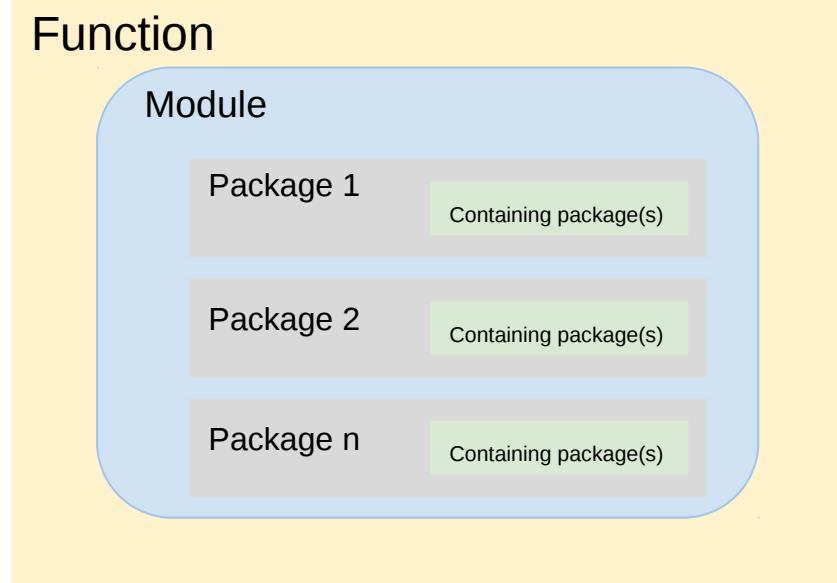


System Hierarchy



Sensing

Drivers

Camera

- baumer
- hexacam
- pointgrey
- vectacam

CAN

- kvaser

GNSS

- garmin
- javad_navsat_driver
- nmea_navsat

IMU

- memsic
- microstrain
- xsens

LiDAR

- hokuyo
- velodyne

Filters

- image_processor
- points_downsample
- points_preprocessor

Fusion

- calibration_camera_lidar
- points2image
- scan2image

Polygon

- points2polygon

Computing

Perception

Detection

- cv_tracker
- image_segmenter
- integrated_viewer
- lane_detector
- lidar_tracker
- road_wizard
- viewers

Localisation

- autoware_connector
- gnss_localizer
- icp_localizer
- ndt_localizer
- orb_localizer

Semantics

- object_map

Planning

Common

- libs

Decision

- decision_maker

Mission

- freespace_planner
- lane_planner
- way_planner

Motion

- astar_planner
- dp_planner
- ff_waypoint_follower
- lattice_planner
- op_local_planner
- op_simulator
- op_simulator_perception
- waypoint_follower
- waypoint_maker

Actuation

Vehicles

- as , - ymc

Data

Map

<code>map_db</code>
<code>map_file</code> <ul style="list-style-type: none">- <code>points_map_loader</code>- <code>vector_map_loader</code>
<code>obj_db</code> <ul style="list-style-type: none">- <code>can_uploader</code>- <code>obj_downloader</code>- <code>obj_uploader</code>
<code>pos_db</code> <ul style="list-style-type: none">- <code>pos_downloader</code>- <code>pos_uploader</code>
<code>vector_map</code>
<code>vector_map_msgs</code>
<code>vector_map_server</code> <ul style="list-style-type: none">- <code>vector_map_client</code>- <code>vector_map_server</code>

Socket

Socket

<code>mqtt_socket</code> <ul style="list-style-type: none">- <code>mqtt_receiver</code>- <code>mqtt_sender</code>	<code>udon_socket</code> <ul style="list-style-type: none">- <code>udon_receiver</code>- <code>udon_sender</code>
<code>oculus_socket</code> <ul style="list-style-type: none">- <code>oculus_sender</code>	<code>vehicle_socket</code> <ul style="list-style-type: none">- <code>vehicle_receiver</code>- <code>vehicle_sender</code>
<code>tablet_socket</code> <ul style="list-style-type: none">- <code>tablet_receiver</code>- <code>tablet_sender</code>	<code>tablet_socket_msg</code>

System

Sync

<code>cv_tracker</code> <ul style="list-style-type: none">- <code>kf_track</code>- <code>obj_reproj</code>- <code>range_fusion</code>
<code>lidar_tracker</code> <ul style="list-style-type: none">- <code>obj_fusion</code>

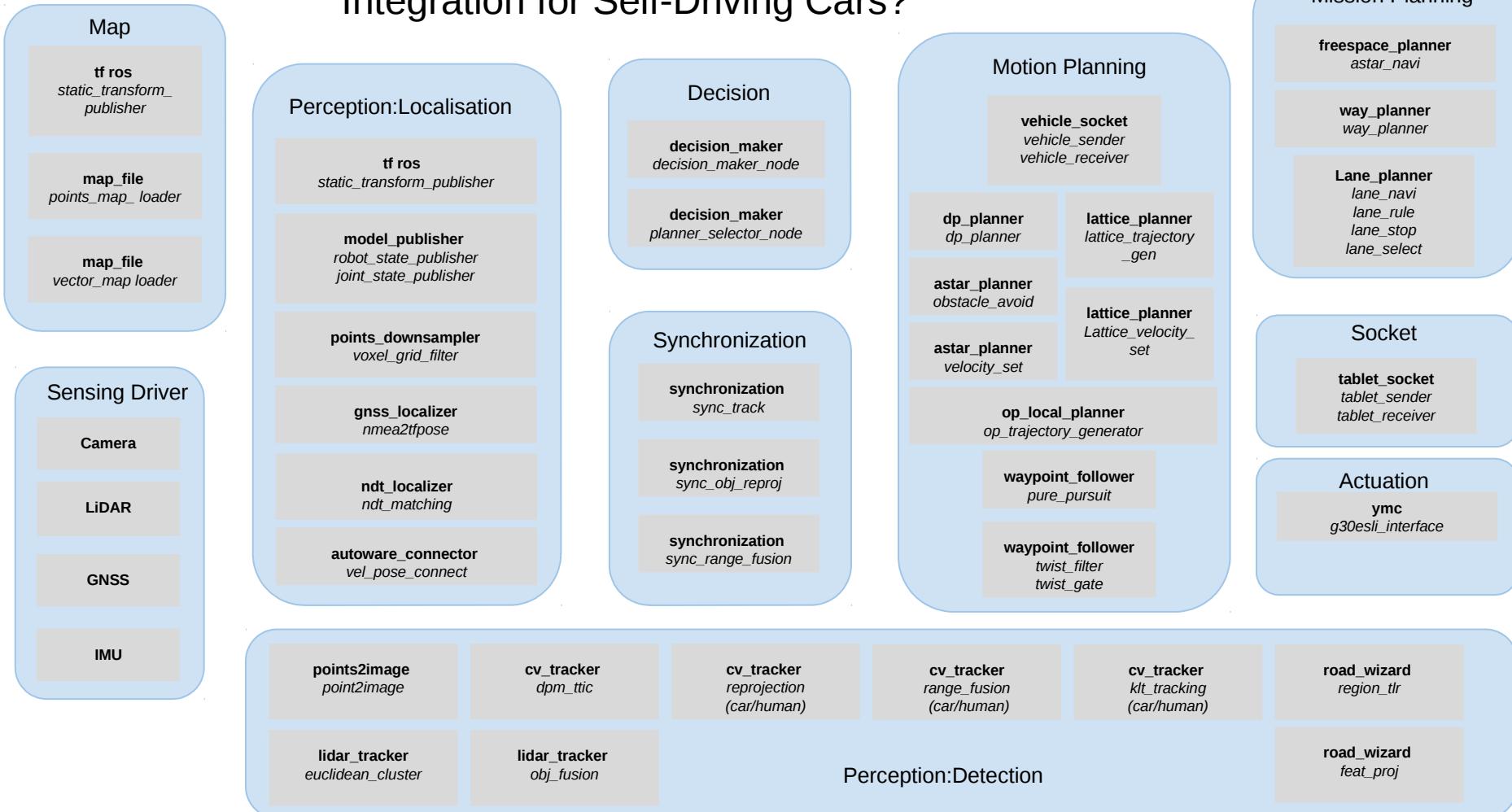
gazebo

<code>catvehicle</code>	<code>point_cloud_converter</code>
<code>laser_scan_converter</code>	<code>twist_cmd_converter</code>

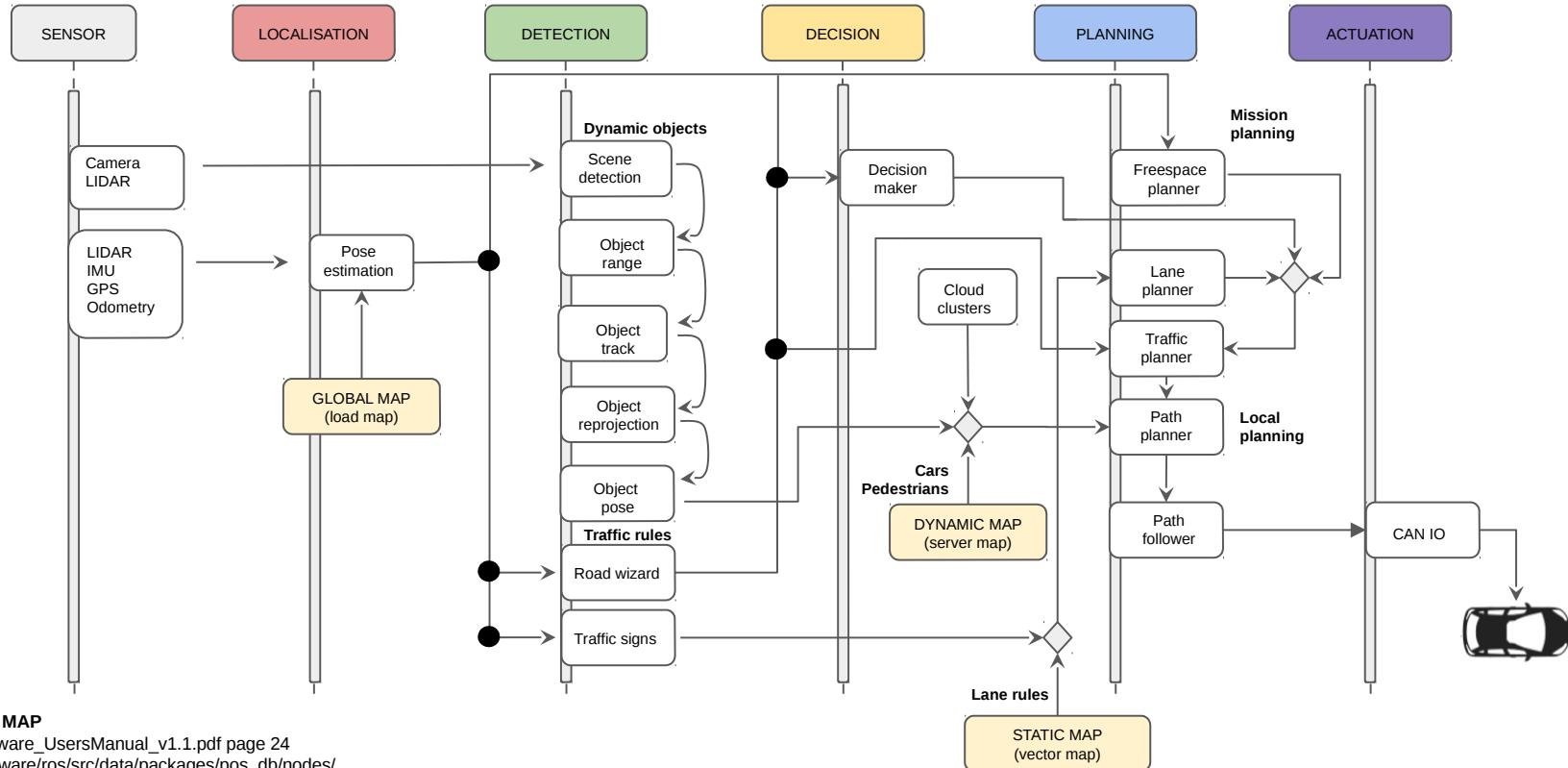
Util

<code>autoware_rviz_plugins</code>	<code>data_preprocessor</code>	<code>fake_drivers</code>	<code>graph_tools</code>	<code>kitti_pkg</code>
<code>marker_downsampler</code>	<code>map_tools</code>	<code>log_tools</code>	<code>model_publisher</code>	<code>sound_player</code>
<code>pc2_downsampler</code>	<code>RobotSDK</code>	<code>runtime_manager</code>	<code>sample_data</code>	
<code>requirements_version_checker</code>				

Integration for Self-Driving Cars?



Use Case: Autoware Basic Self-Driving Integration



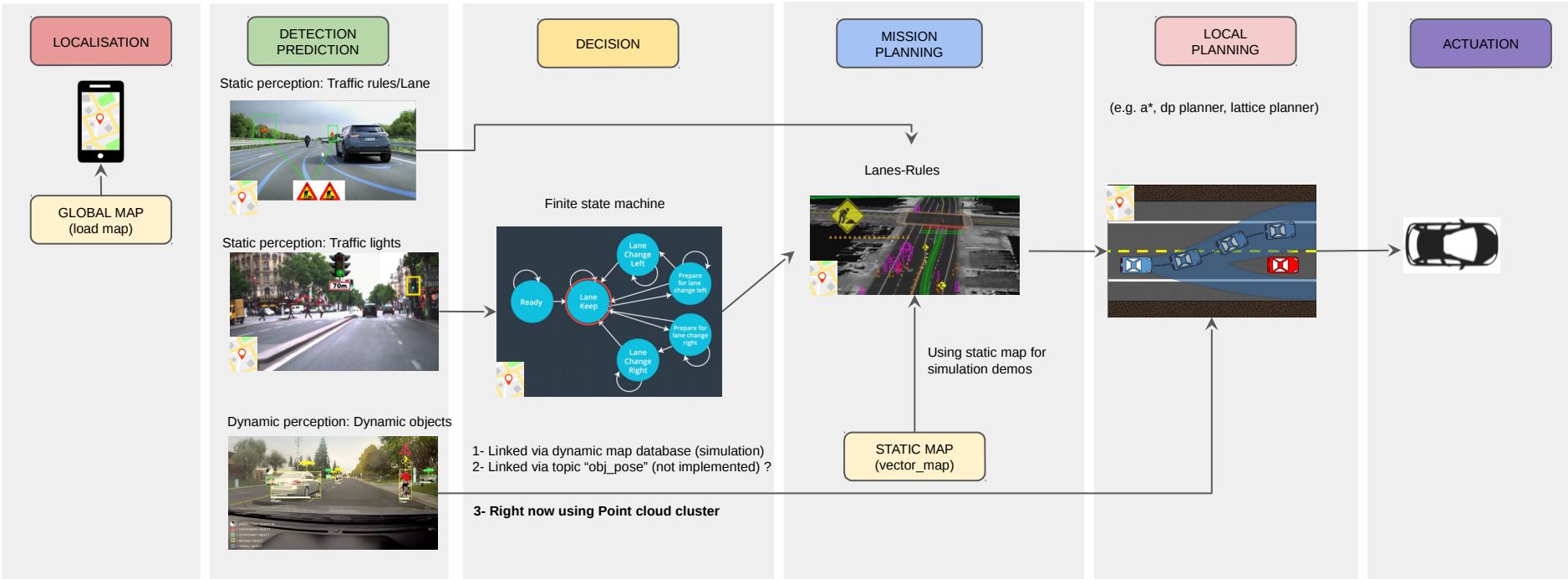
DYNAMIC MAP

Doc: Autoware_UsersManual_v1.1.pdf page 24
Files: Autoware/ros/src/data/packages/pos_db/nodes/..

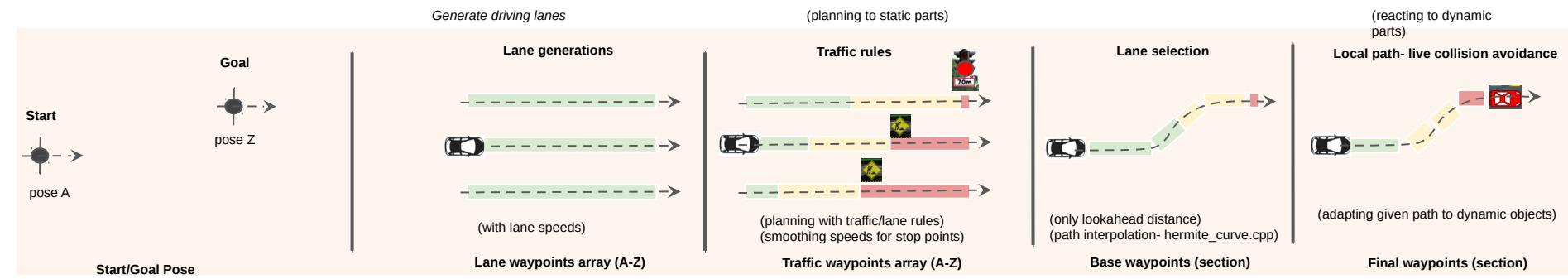
STATIC MAP

Doc: Autoware_UsersManual_v1.1.pdf page 27,39
Files: Autoware/ros/src/data/packages/vector_map/lib/vector_map

Autoware Autonomous Driving (Simplified Idea)



Overview of Autoware Autonomous Path Planning Strategy



Examples: Closed Loop Autonomous Driving using

A* planning
DP planning
Lattice planning

Node messaging

