ROS AUTONOMOUS CARS

Level 3 of autonomy → all tasks performed autonomously but human driver expected to intervene whenever required
↳ conditional automation

Sensors → Data in → Sensor array

- Laser → not used in navigation like traditional service robots
- Camera
- Lidar
- GPS
  ↳ localization

↳ impractical for outdoor localization
↳ only for obstacle detection mainly
↳ but fast, simple and reliable
↳ first security measure

DBW interface allows you to communicate with real cars through CAN-bus protocol

You won't publish Twist topics but the state of the steering wheel, brake pedal, throttle-pedal and gear selected

DBW infrastructure will convert them into CAN-bus messages and publish them into CAN-Bus-like topics which will be read by a node and send them to a CAN-bus device from which car systems will read and publish.

Camera: Read street signs, detect pedestrians and stay in lane

- Front-facing camera
- Between 4 and 6 to cover all visual space

`rosrun rqt_image_view rqt_image_view`
↳ to see live camera data

GPS allows you to know where to go and which path to take once you set it with Satellit Map data

Security elements in a safe autonomous car:

- Obstacle Detection
- System Failure Measures

For security reasons, the car shouldn't be moved directly by /cmd_vel topic, it needs 2 barriers:

- Dead Man Switch: when control communication is lost, autonomous vehicles have to move to a safe state
- Obstacle detection: When an obstacle is detected, the vehicle must stop until the obstacle is removed

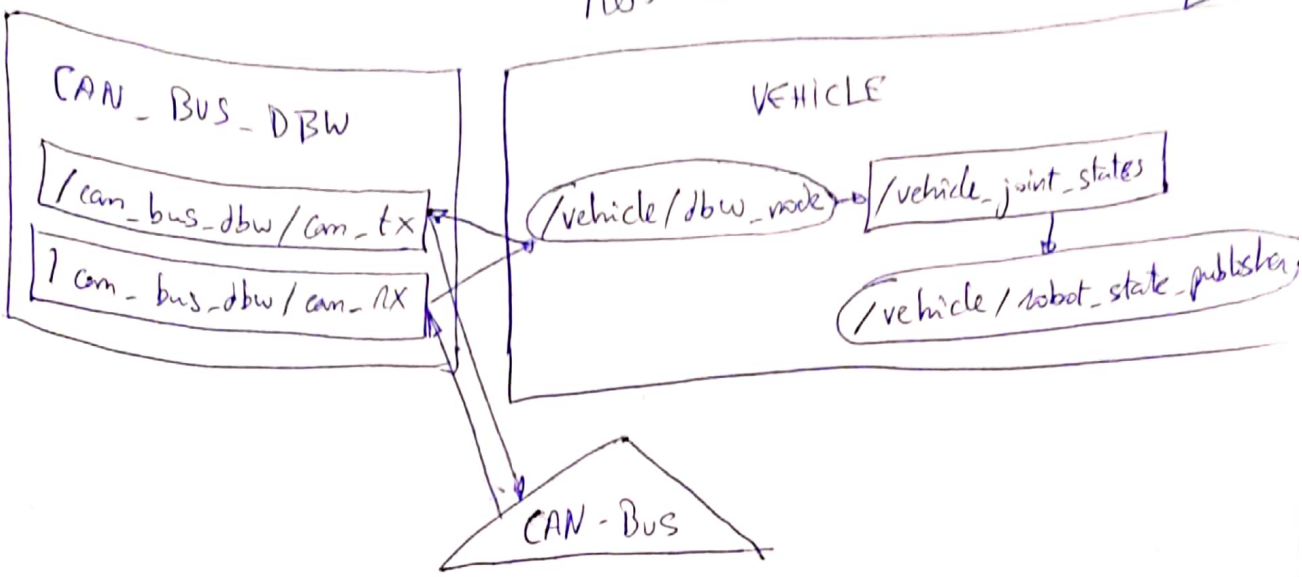
CAN-Bus is widespread standard created by Bosch and it's one of the five protocols used in the on-board diagnostics (OBD)

La abolity ~~to~~ multiplex electrical wiring which comes in handy for avoiding the need ~~for~~ hundreds of different cables to send and receive information from all the electrical devices in the system

- Message
  - Identifier (11 bits long)
  - DLC (data length code) (4 bits long)
  - Data field (0-64 bits long) x (0-8 bytes)

Each sensor will be watching for its own identifier
It will also publish the data this sensor generates with that same identifier
There are two cables:

- tx: CAN-bus output
- rx: " " input



To make the car move, only needed to publish into ~~the bus~~ /can_bus_dbw/can_tx. Anytime you wanted to send a message, publish into this topic. Then a node would just read it and connect to the CAN-Bus driver device through a USB. The same process to read data, but you would receive it through /can_bus_dbw/can_rx