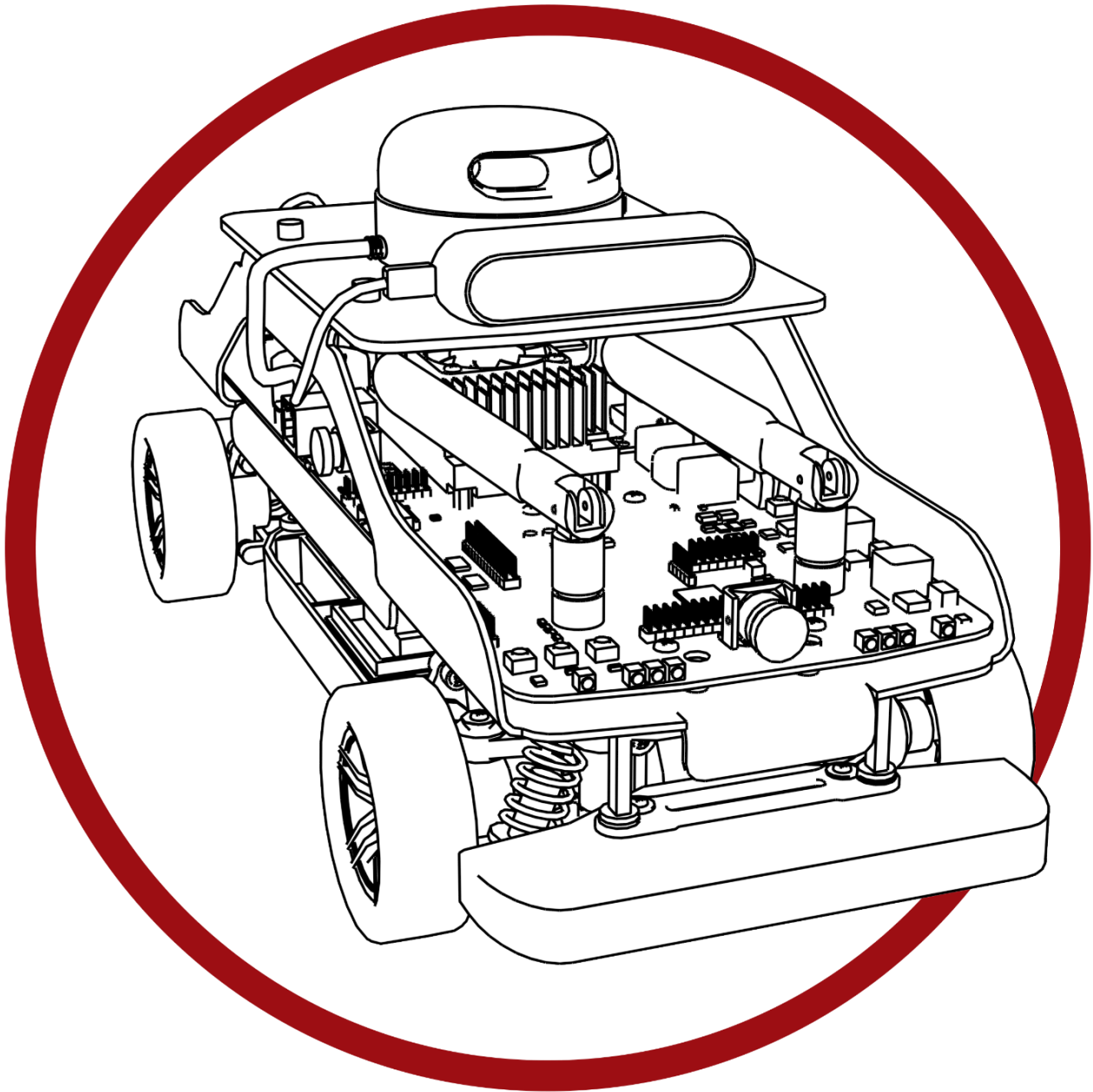QUANSER
INNOVATE · EDUCATE

# Self-Driving Car Research Studio

ROS-to-ROS Communication

V 1.0 (November 2020)

ROS-to-ROS Communication

The QCar was built around ROS2, but with the vast libraries built around ROS1, sometimes you need to mix ROS1 and ROS2 nodes. ROS-to-ROS communication can be done in multiple ways. In this documentation, we will explain how to setup a QCar-to-Ubuntu PC communication using a ROS2 package called **ros1_bridge**.
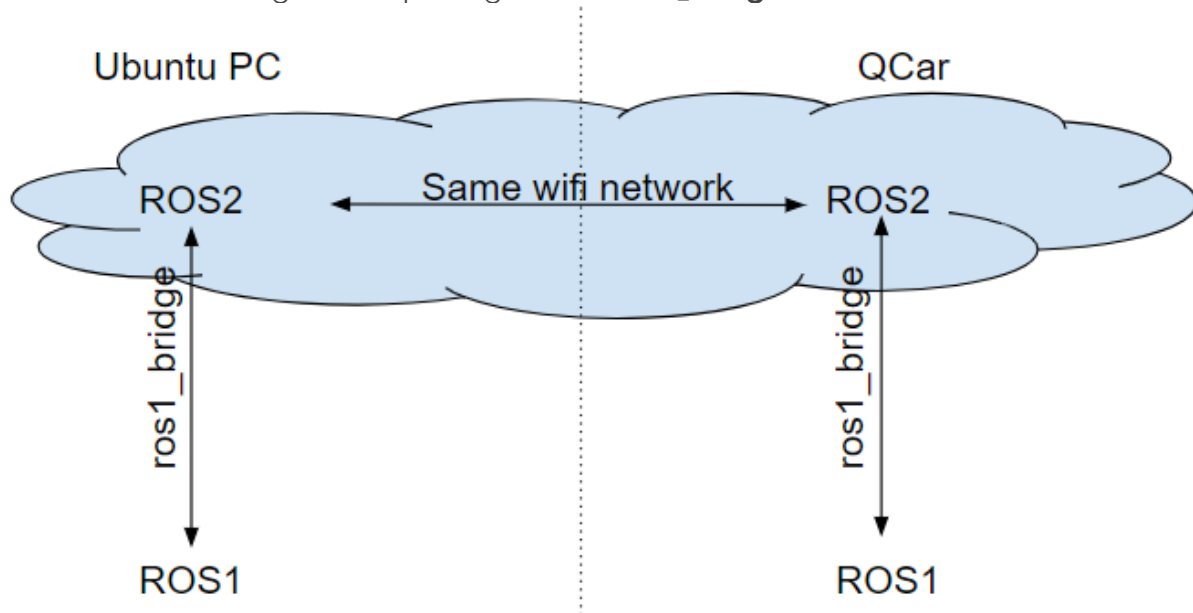
All the topics that are published by a ROS2 device can be subscribed by other ROS2 devices as long as they are in the same network. **If users are only using ROS2** as their main development platform, and they are running ROS2 on both QCar and their Ubuntu PC, then the ROS-to-ROS communication does **not need extra setup**.

If users will also use ROS1 on either/both platforms, then they need to download a ROS2 package called **ros1_bridge.** This package will bring all the topics from ROS1 environment to ROS2 or vice-versa.

A typical use case is a ROS1 node on the QCar capturing a CSI image which needs to be used by a ROS 1 node on a remote machine. Using the **ros1_bridge** on the **QCar,** ROS2 on the **QCar** will bring that topic from ROS1 and share it with the other ROS2 device **Ubuntu PC** on the same network. The **ros1_bridge** on **Ubuntu PC** will detect that topic in the *ros2 topic list* and bring it to ROS1 on the **Ubuntu PC** where it can process the image to reading a sign, detect lanes, etc. Then the resulting command will use the same tunnel to return to the ROS1 environment on QCar**.**

When measuring the performance, WiFi latency should be considered. See user guide **II – Connectivity** for discussion on WiFi configurations.

Please follow this link https://github.com/ros2/ros1_bridge/blob/master/README.md to setup **ros1_bridge** on each device.