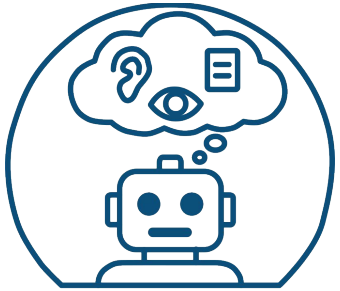
ROS2: Simulator (Gazebo)

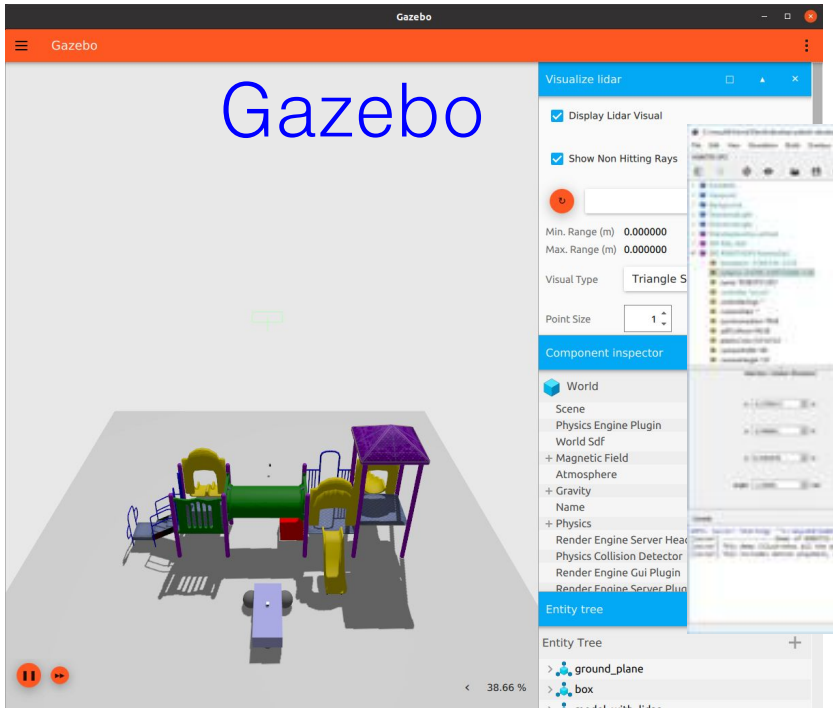
운영체제의 실제

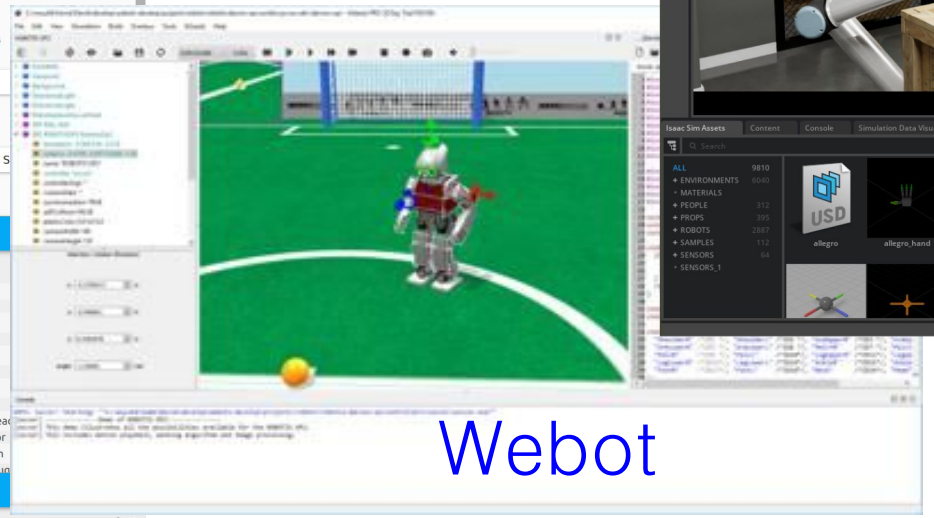
안인규 (Inkyu An)

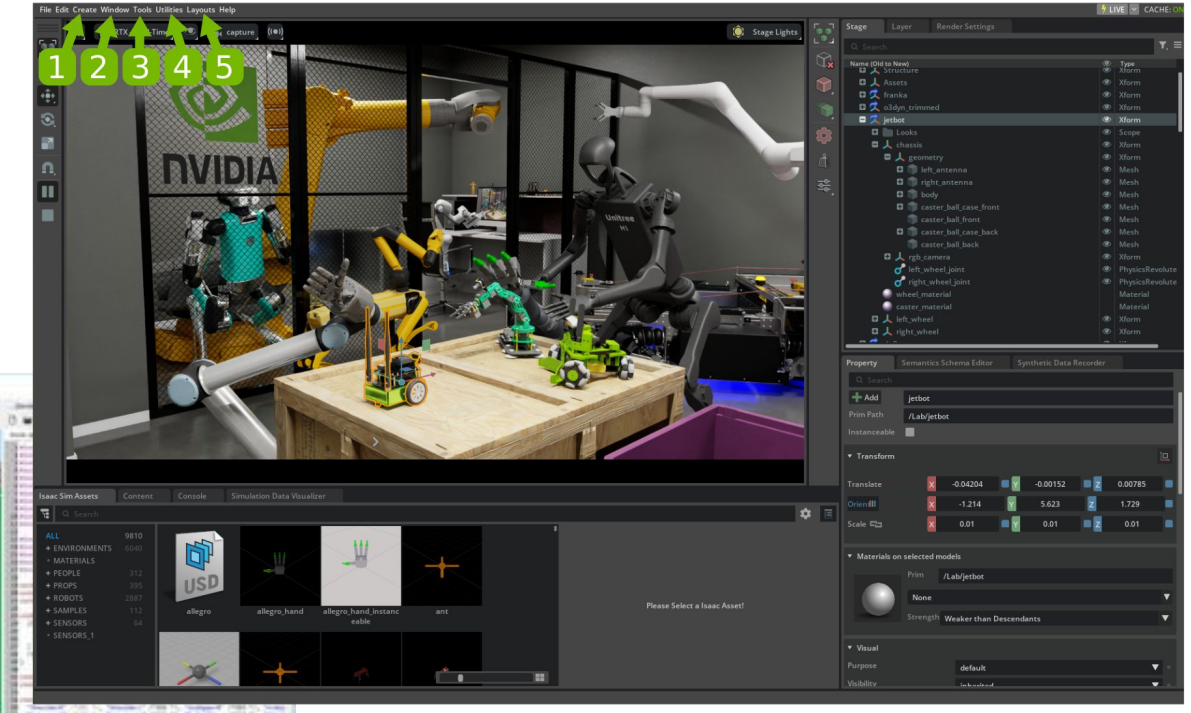


Simulator

- Several advanced robot simulators can be used with ROS2
  - Gazebo
  - Webot
  - Isaac Sim

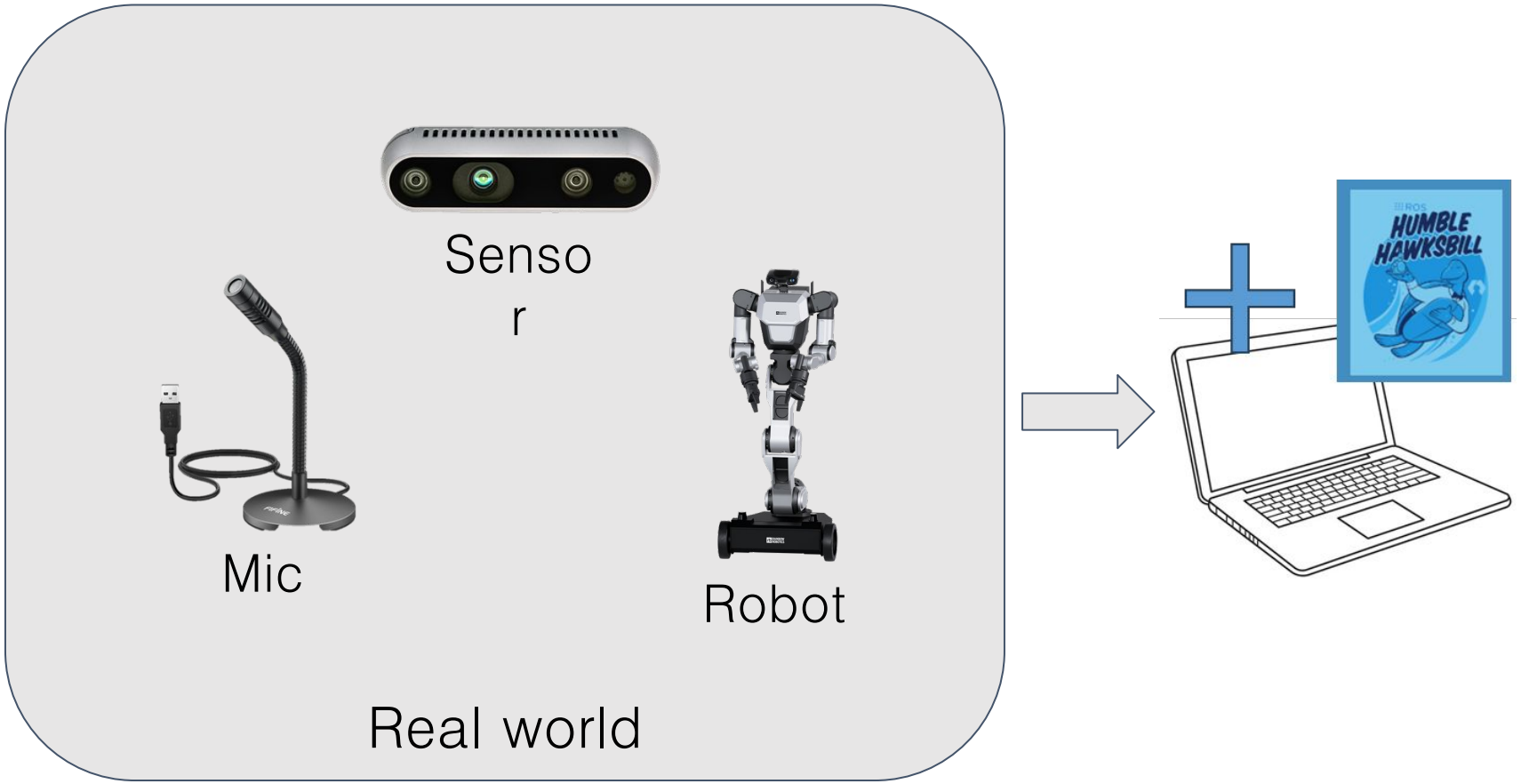






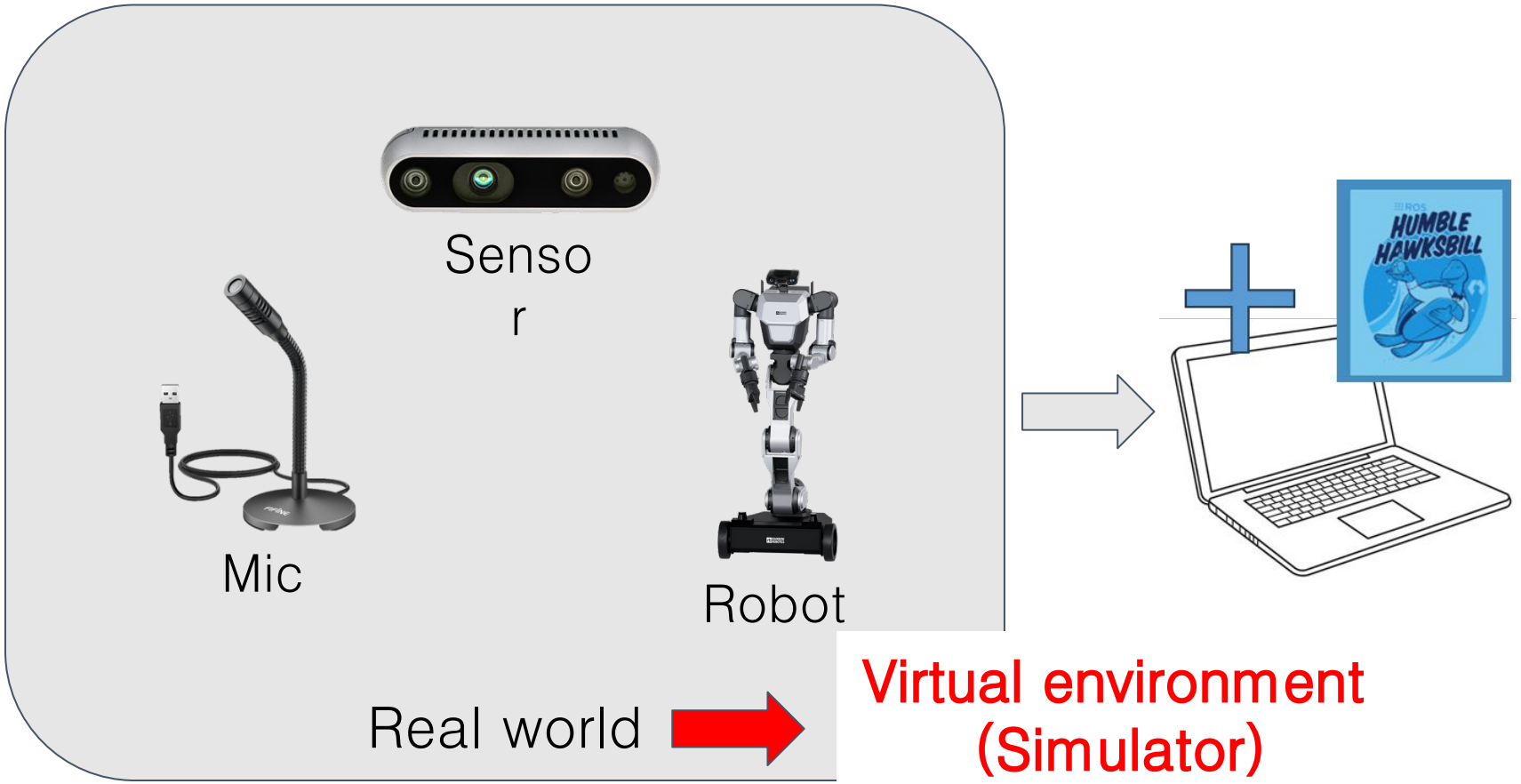
Simulator

- What can we do with simulation?
  - We can operate the robot in a virtual environment



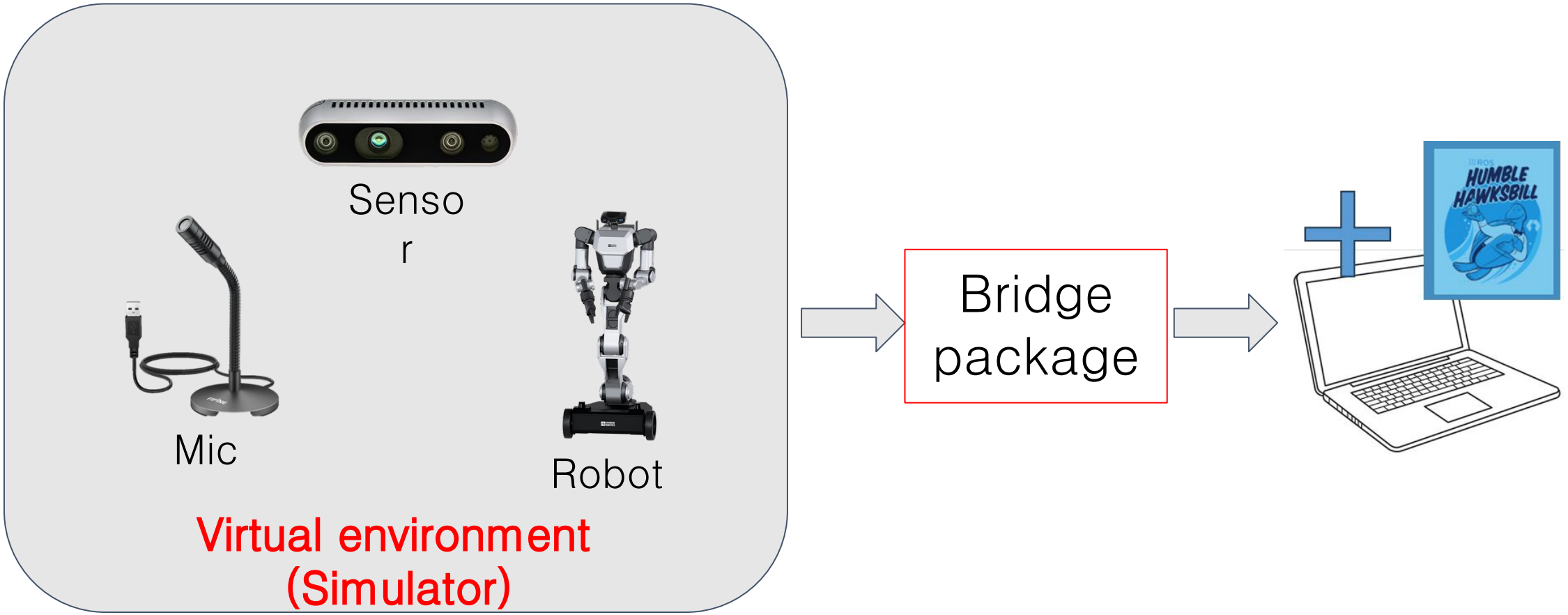
Simulator

- What can we do with simulation?
  - We can operate the robot in a virtual environment



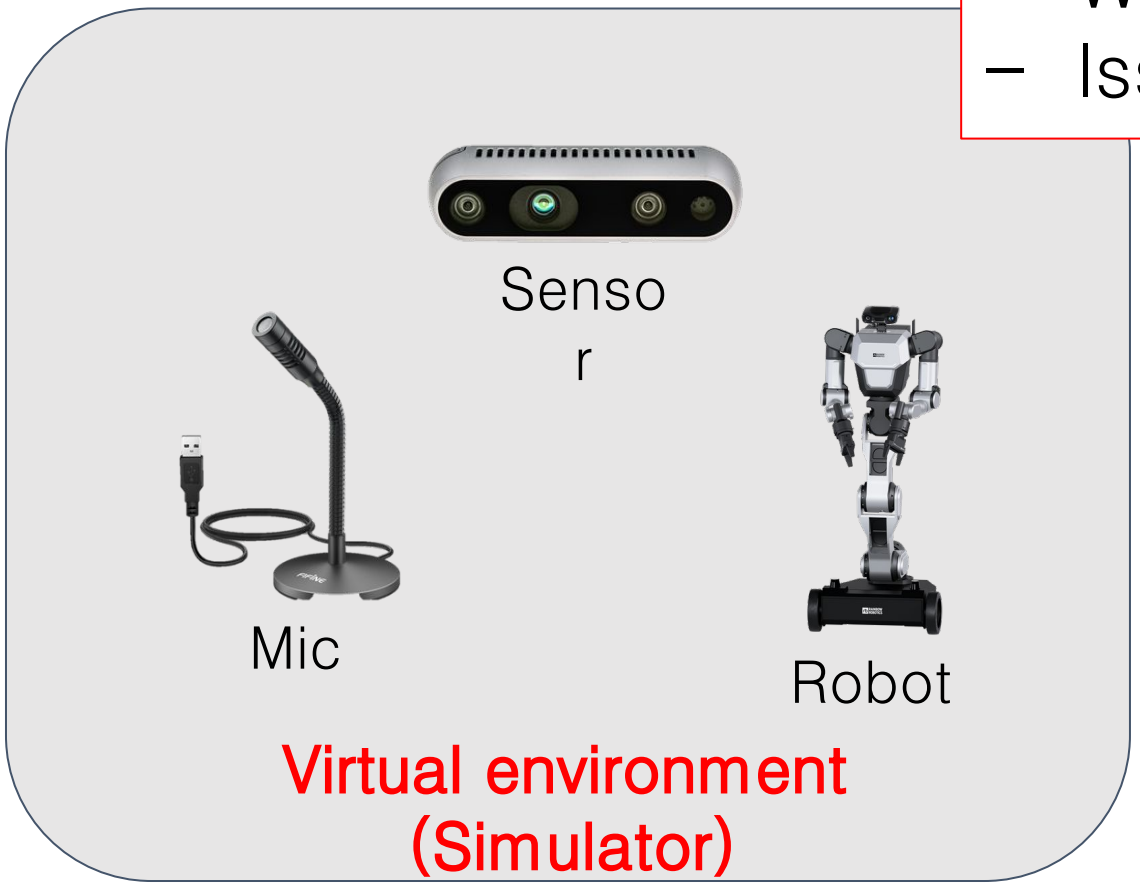
Simulator

- What can we do with simulation?
  - We can operate the robot in a virtual environment

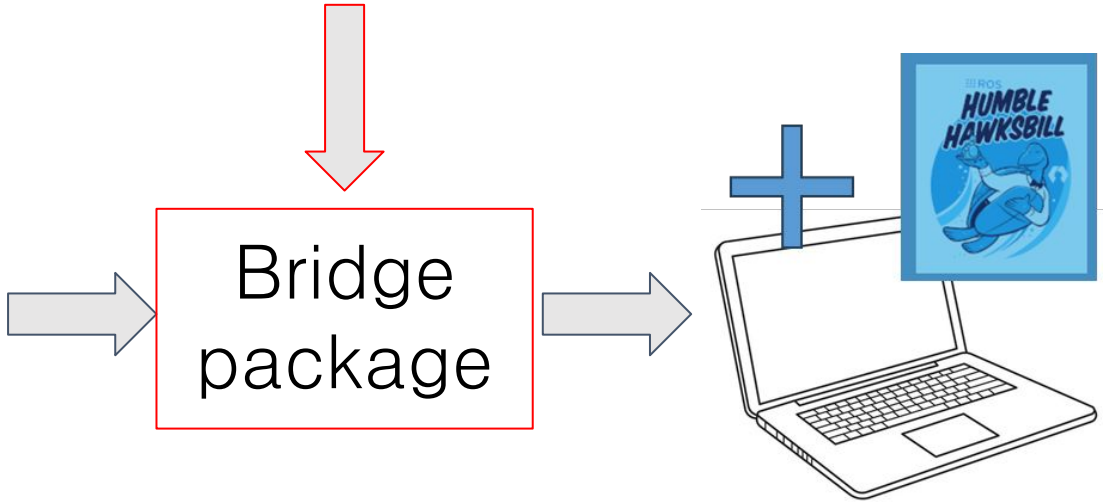


Simulator

- What can we do with simulation?
  - We can operate the robot in



- Gazebo: ros2_gz_bridge
- Webots: webots_ros2_core
- Issac Sim: isaac_ros_bridge



Simulator – Gazebo

- Install Gazebo (Fortress):
  - Install some necessary tools:

```
$ sudo apt-get update
$ sudo apt-get install lsb-release gnupg
$ sudo apt install ros-dev-tools
$ sudo apt-get install wget
```

  - Install Ignition Fortress:

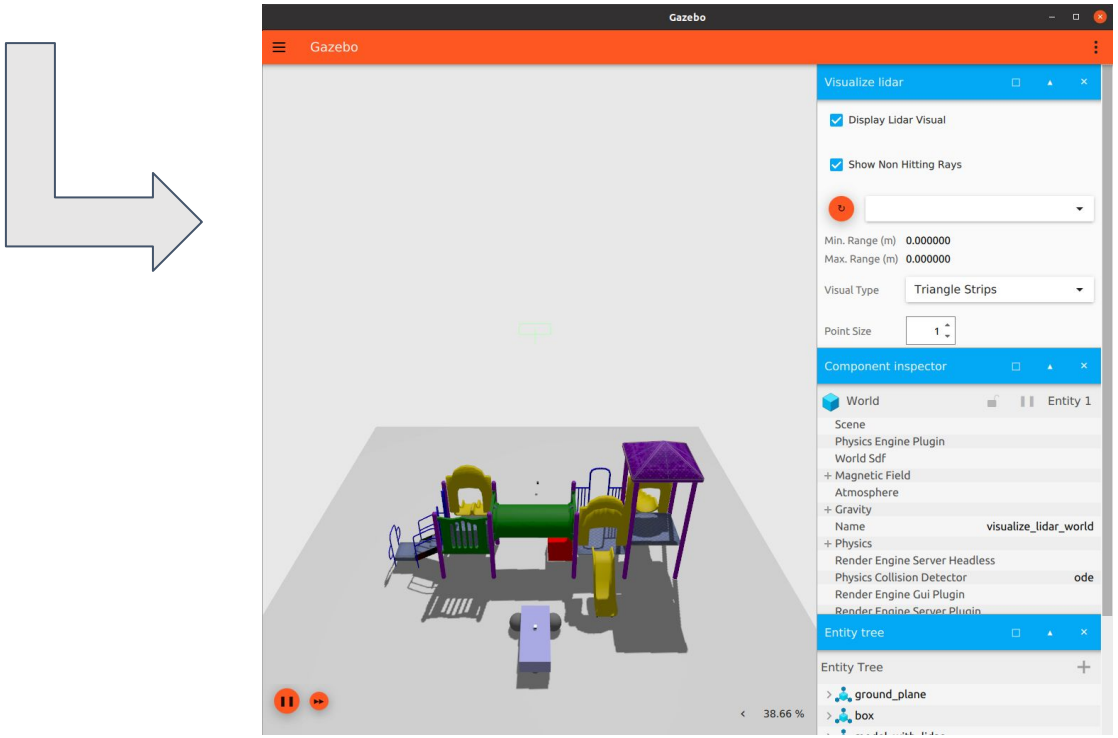
```
$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable
`lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'
$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
$ sudo apt-get update && sudo apt-get install ignition-fortress
```

Please ref: “https://gazebosim.org/docs/fortress/install_ubuntu/” and “https://turtlebot.github.io/turtlebot4-user-manual/software/turtlebot4_simulator.html”

Simulator – Gazebo

- **Launch the simulation:**
  - In this demo, we are going to simulate a simple diff drive robot in Gazebo

```
$ ign gazebo -v 4 -r visualize_lidar.sdf
```



Simulator – Gazebo

- **Launch the simulation:**
  - When the simulation is running, we can check the topics provided by Gazebo:

```
$ ign topic -l
```

```
$ ign topic -l
/clock
/gazebo/resource_paths
/gui/camera/pose
/gui/record_video/stats
/model/vehicle_blue/odometry
/model/vehicle_blue/tf
/stats
/world/visualize_lidar_world/clock
/world/visualize_lidar_world/dynamic_pose/info
/world/visualize_lidar_world/pose/info
/world/visualize_lidar_world/scene/deletion
/world/visualize_lidar_world/scene/info
/world/visualize_lidar_world/state
/world/visualize_lidar_world/stats
```

Simulator – Gazebo

- **Launch the simulation:**
  - When the simulation is running, we can check the topics provided by Gazebo:

```
$ ign topic -l
```

```
$ ign topic -l
/clock
/gazebo/resource_paths
/gui/camera/pose
/gui/record_video/stats
/model/vehicle_blue/odometry
/model/vehicle_blue/tf
/stats
/world/visualize_lidar_world/clock
/world/visualize_lidar_world/dynamic_pose/info
/world/visualize_lidar_world/pose/info
/world/visualize_lidar_world/scene/deletion
/world/visualize_lidar_world/scene/info
/world/visualize_lidar_world/state
```

  - Since we have not launched an ROS 2 nodes (bridge), the output from "ros2 topic list" should be free of any robot topics:

```
$ ros2 topic list
```

```
$ ros2 topic list
/parameter_events
/rosout
```

Simulator – Gazebo

- **Configuring ROS2:**
  - To be enable to communicate our simulation with ROS2, we need to use a package called “ros_gz_bridge”

```
$ sudo apt-get install ros-humble-ros-ign-bridge
```

  - We are going to create a bridge for the topic “/model/vehicle_blue/cmd_vel”

```
$ source /opt/ros/humble/setup.bash
$  ros2 run ros_gz_bridge parameter_bridge
/model/vehicle_blue/cmd_vel@geometry_msgs/msg/Twist]ignition.msgs.Twist
```

  - Send a command to the topic using “ros2 topic pub”

```
$ ros2 topic pub /model/vehicle_blue/cmd_vel geometry_msgs/Twist "linear: { x: 0.1
}"
```

For more details, please ref: “https://github.com/gazebosim/ros_gz/tree/ros2/ros_gz_bridge”

Simulator – Gazebo

- **Visualizing lidar data in ROS2:**
  - The diff drive robot has a lidar. To send the data generated by Gazebo to ROS 2, we need to launch another bridge:

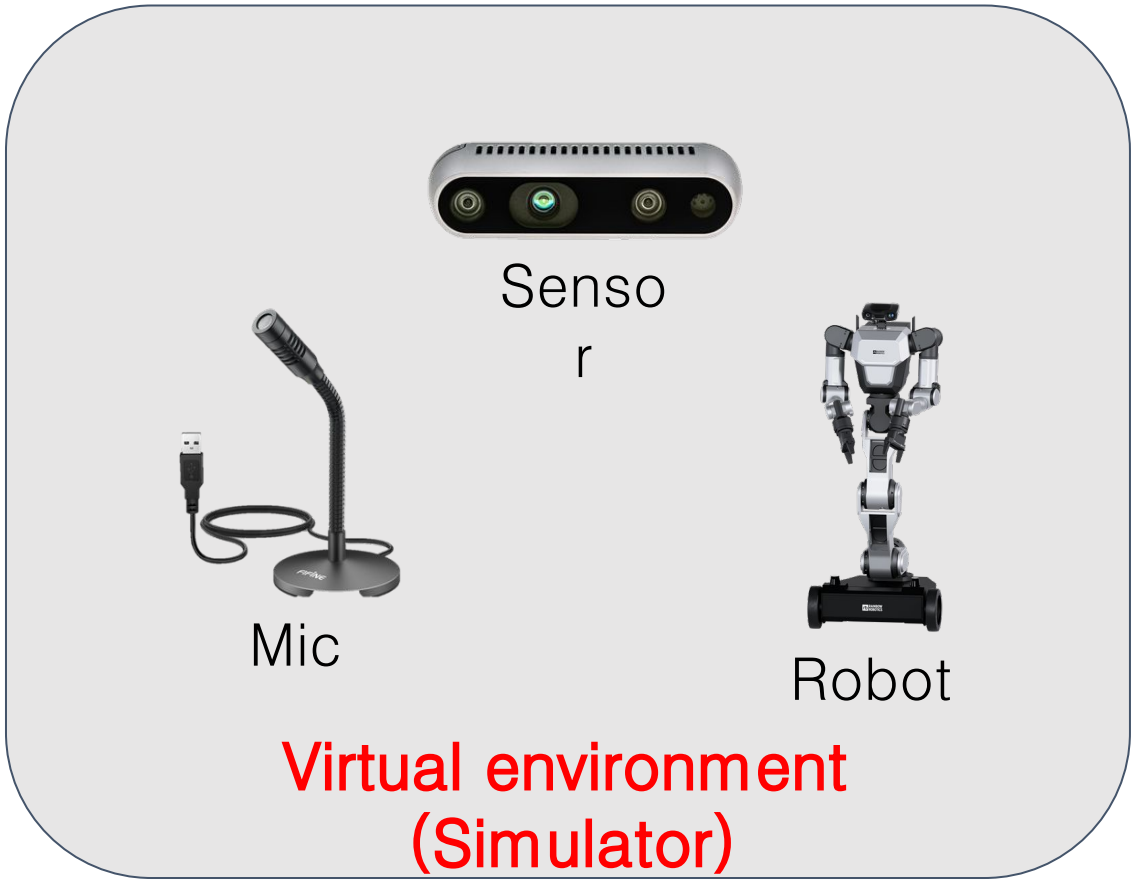
```
$ ros2 run ros_gz_bridge parameter_bridge
/lidar2@sensor_msgs/msg/LaserScan[ignition.msgs.LaserScan --ros-args -r
/lidar2:=/laser_scan
```

For more details, please ref: “https://github.com/gazebosim/ros_gz/tree/ros2/ros_gz_bridge”

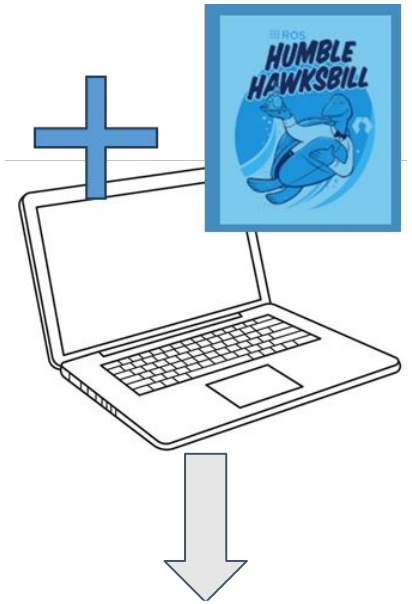
Simulator

- Gazebo: ros2_gz_bridge
- Webots: webots_ros2_core
- Issac Sim: isaac_ros_bridge

- **What can we do with simula**
  - We can operate the robot in a virtual environment



Bridge package

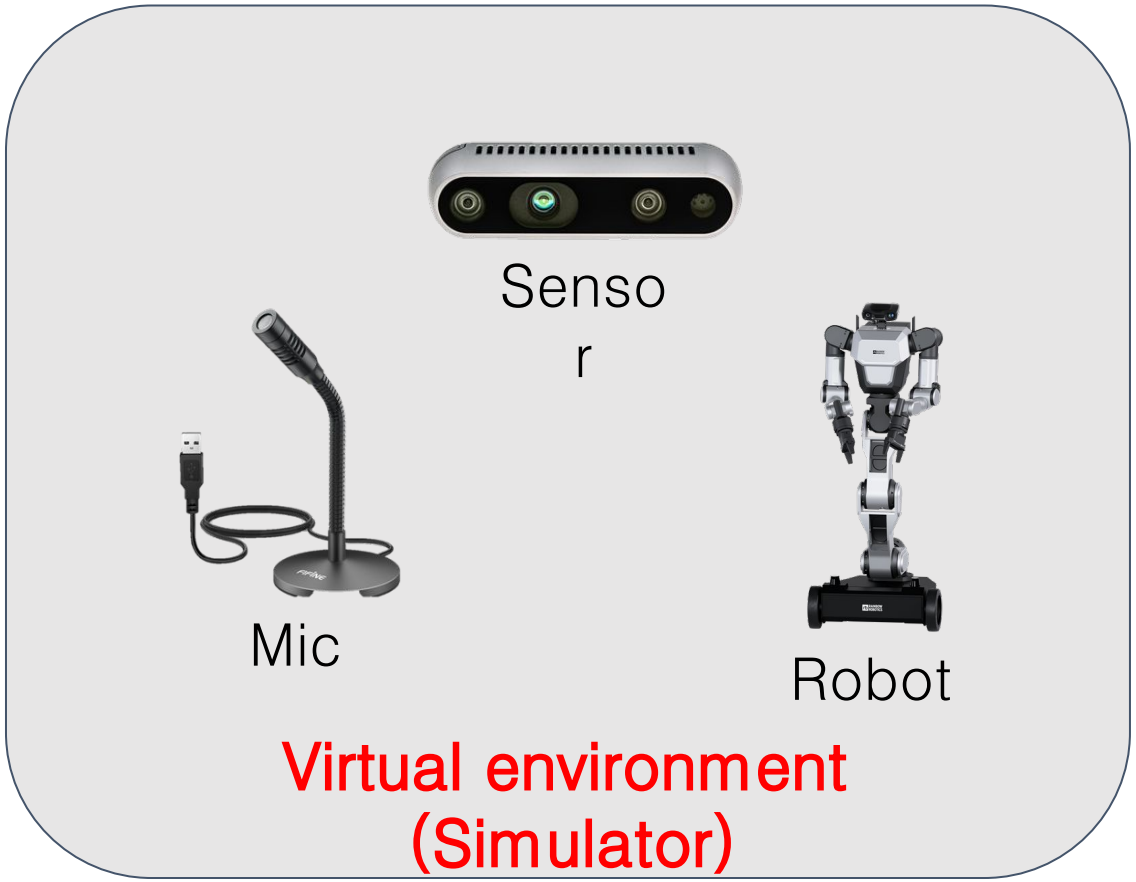


How to visualize the received topics?

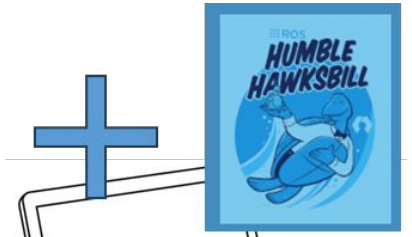
Simulator

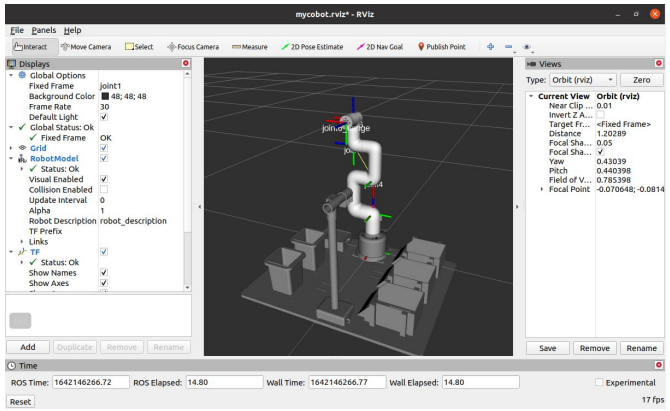
- Gazebo: ros2_gz_bridge
- Webots: webots_ros2_core
- Issac Sim: isaac_ros_bridge

- **What can we do with simula**
  - We can operate the robot in a virtual environment



Bridge package



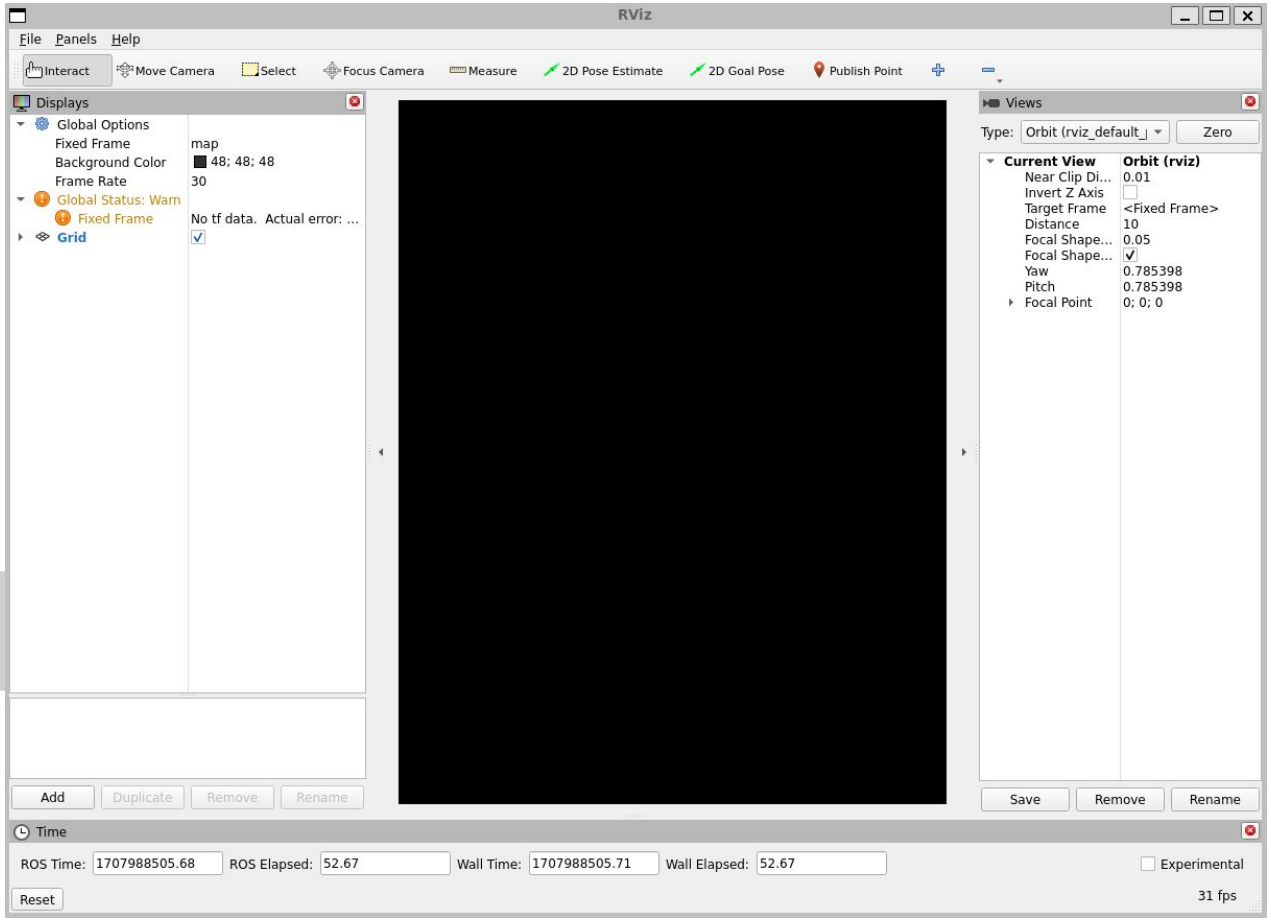


RViz

RViz2

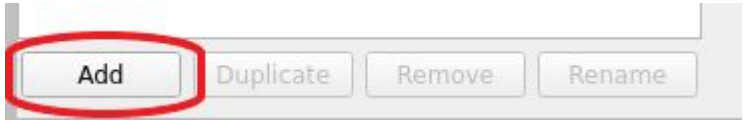
- 3D visualization tool for ROS2
- Subscribes to topics and visualizes the message contents
- Different camera views (orthographic, top-down, etc.)
- Interactive tools to publish use information
- Save and load setup as RViz configuration
- Extensible with plugins

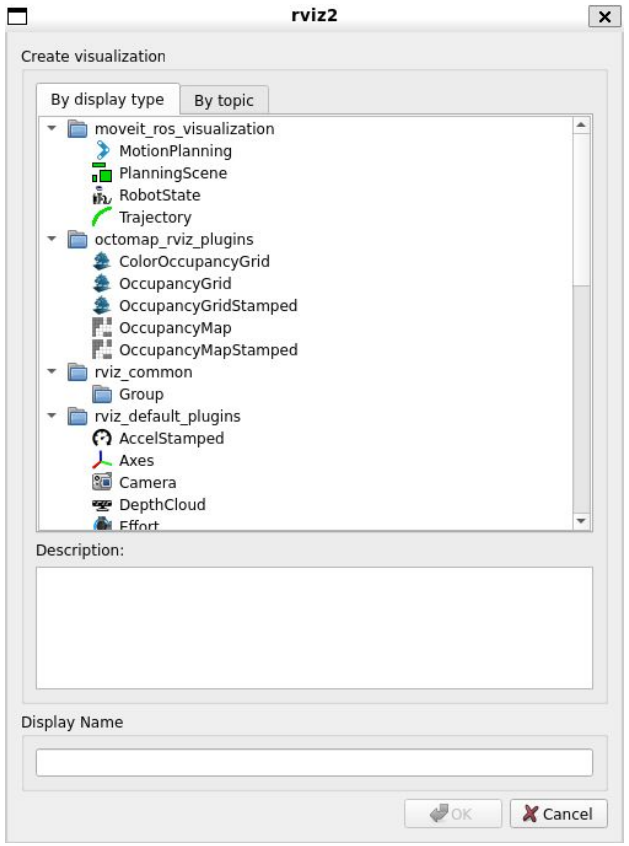
```
$ ros2 run rviz2 rviz2
```



RViz2

- Add a new display





We have to configure the fixed frame ("vehicle_blue/lidar_link/gpu_lidar") → Why?