# **Section 2:** ROS Workspace and Programming Basics

## AA274 - Principles of Robot Autonomy I

Week 2 – Autumn 2023

# Section Goals

- Start with **mini-lecture**.
- Work through tasks in write-up with groups of 3-4.
- Implement and **test code from HW on Turtlebots.**
- You can leave when a CA signs off on all of your tasks!

- Sections will have recorded attendance (part of your grade), and **each section will be 5% of your total grade**.
- Come to sections **on time**, and with the corresponding portions of your HW **complete**.
- We will not stay after hours, but if you group is struggling to complete the Section we host **Skilling Lab Office Hours** where you can make this up.

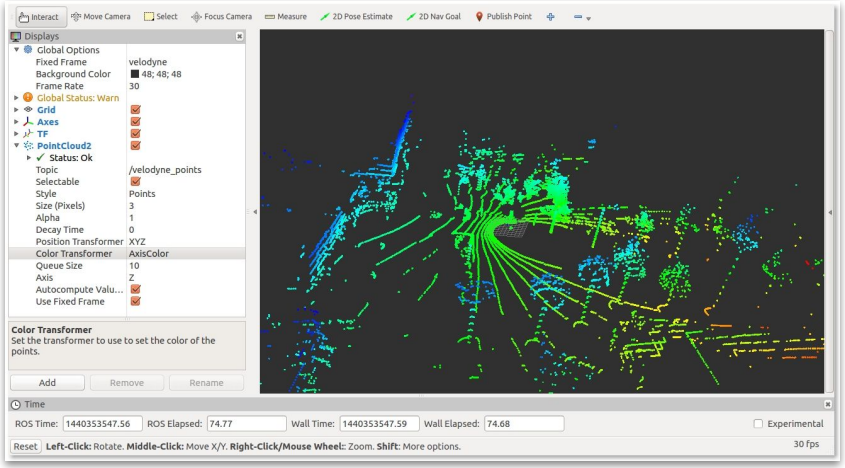
# ROS


Initially released in 2007 by the founder of Zipline!

**16+ years of open-source.**

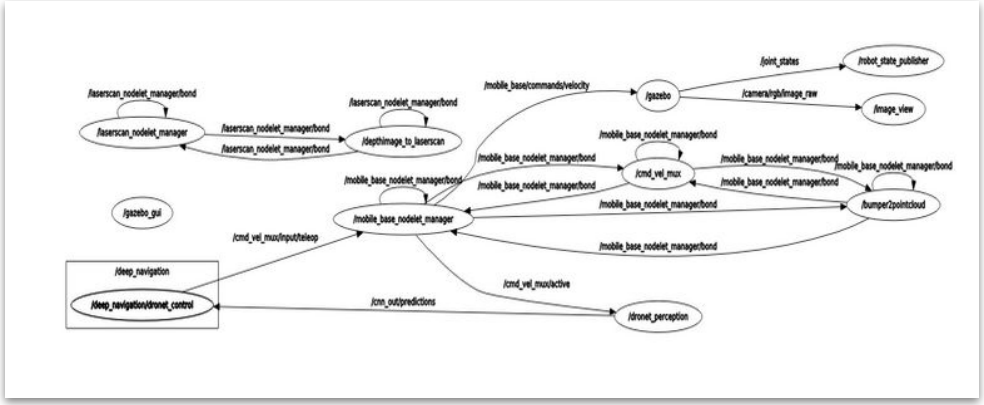
# What is ROS?

- **Event programming, networking**, and **build** tools.
- **Drivers** for sensors and motors.
- **Visualization** and logging tools.
- An interface for all sorts of **external libraries** (SLAM, planning, perception).



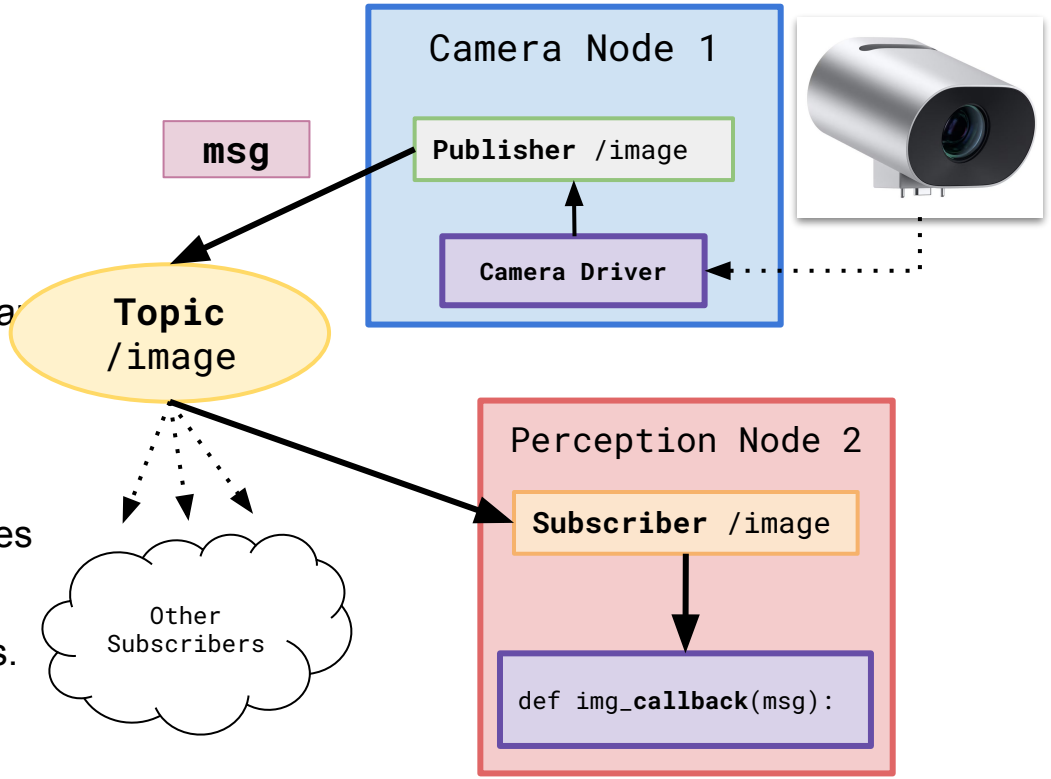
# Why use event programming?

- **Even simple robots are complicated!**
- Lots of sensors and actuators means that we need **scalable tools** to handle everything.
- **For loops won't cut it.**



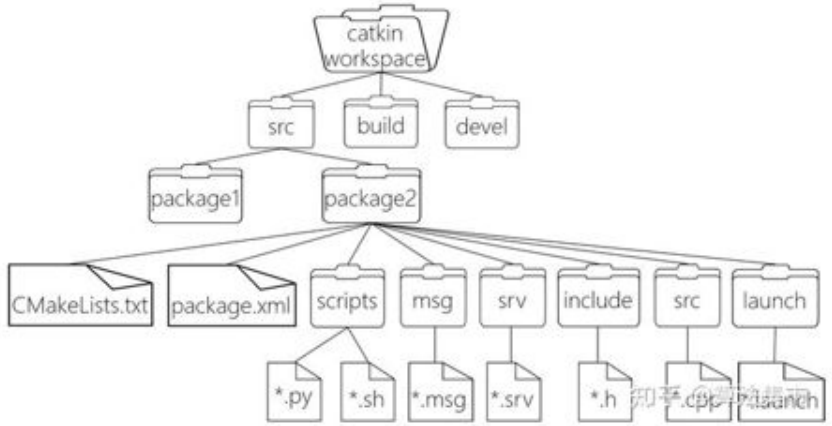

# ROS Event Programming Basics

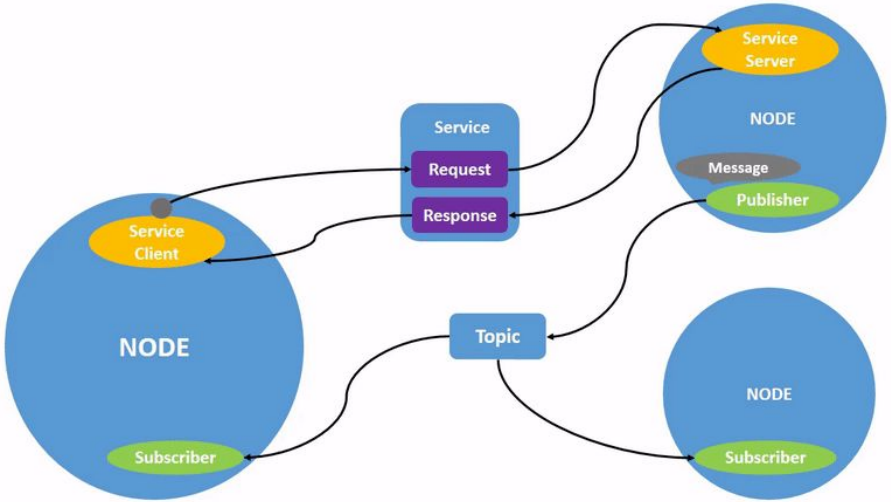
- Nodes
  - Manage **publishers/subscribers, callbacks**, and external code.
- Topics and Messages
  - Topics are “hubs” where messages a
    “posted” by publishers.
  - **Messages are formatted data!**
- Publishers
  - Send messages **to** topics **from** nodes
- Subscribers
  - Pull messages **from** topics **to** nodes.



# ROS Communication

- Message Types
  - Data structure that holds some information about the robot
- Publication
  - Broadcast message to the ROS network
- Subscription
  - Listens to some broadcasted channel



# ROS Development Environment

- **Workspaces**
  - Organizes packages.
- **Packages**
  - Organizes code for nodes.
- **CMakeLists.txt**
  - Tells *colcon* how to build.
- **package.xml**
  - Tells *rosdep* what to install.

```
~/tb_ws/    # My group's workspace root directory
       build/              # Compiled libraries
       install/setup.bash  # Source workspace packages
       src/                # Top-level package directory
          external_pkg1/   # Some external package that I am using
          external_pkg2/   # Another external package
          group4_repo/     # My group's autonomy repository
                     navigation_pkg/    # My group's Navigation ROS Package
                     perception_pkg/    # My group's Perception ROS Package
                                  CMakeLists.txt  # colcon build instructions
                                  package.xml     # ROS package details
                                  src/            # C++ Code lives here!
                                  my_pytorch_py_lib/   # My custom python module
                                  scripts/             # ROS Python Executables
                                         camera_node.py     # Camera node code
                                         detection_node.py  # Detection node code
```

# Live ROS Demo.