Introduction to ROS

Admin

- Questions about the project?
- For quiz
- https://www.youtube.com/watch?v=JyWQ-71QAtI
-
- Refs:
  - Some of the material here (and many of the images) was distilled from
  - https://www.clearpathrobotics.com/assets/guides/melodic/ros/index.html

Controlling computing machines

- Computing machines are big silicon calculators that
  - Do math
  - Send electrical impulses to peripherals
- Soooooooooo
  - How do we control them?

Controlling computing machines

- Computing machines are big silicon calculators that
  - Do math
  - Send electrical impulses to peripherals
- Soooooooooo
  - How do we control them?
  - Generally have an operating system running in kernel/protected mode
    - Then write programs running in user mode.
  - OR – write final program to run on hardware as embedded systems.

And robots?

- And robots? They are just computing machines
  - With lots more peripherals than usual
- So in the old days?

And robots?

- And robots? They are just computing machines
  - With lots more peripherals than usual
- So in the old days?
  - Often embedded system
    - (don’t tell Dr. Black)
    - Shakey: Lisp machine with drivers
    - Rod Brook’s robot ‘insects’
  - Or standard OS with custom drivers for peripherals
    - Eg: GRACE from the 2000s
    - Linux with extra drivers

The upshot

- The upshot of this
  - Every time there is a new robot
  - Have to build significant low level software
    - Again and again
  - Lots of low level work gets done
  - But higher level work comes to a screeching halt as we get better robots
    - And then often gets rewritten when we have to do the whole thing again on top of a new platform.

Robot Frameworks

- So in the late 2000s/early 2010s groups started to write robot frameworks
  - Framework would interface with the hardware
  - And publish a standard set of functions for interacting with that hardware.
  - So framework supports new robot
    - And higher level code moves to new robot
  - Of course hardware matters
    - Cameras etc
    -

Robot Frameworks

- Some popular* robot frameworks:
  - GoBot (for go programming) https://gobot.io/
  - ROS (Robot Operating System) http://wiki.ros.org/ROS
  - YARP (Yet and Robot Platform) https://yarp.it/latest/index.html
  - Etc
  -

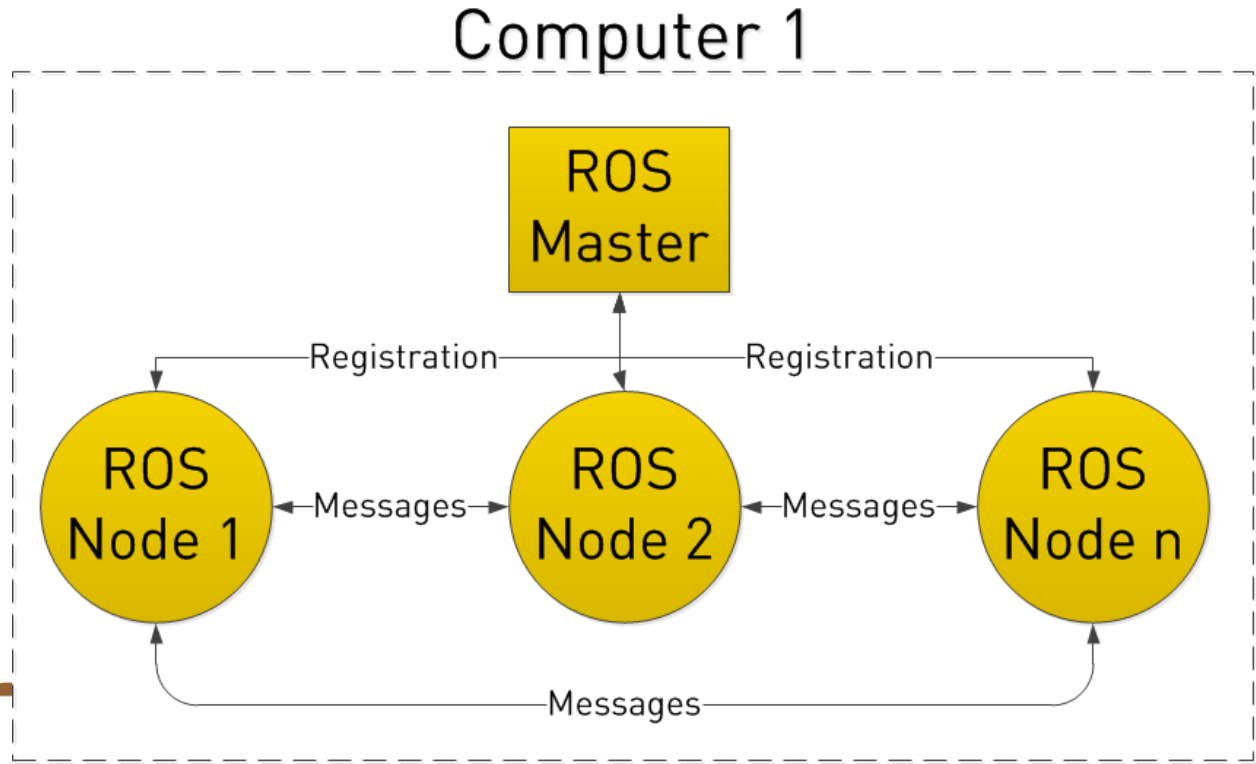
* of course ‘popular’ in robot frameworks is much smaller uptake than popular in web frameworks

Robot Frameworks III

- In the past couple of robotics classes
  - Used Gobot
  - For gopigo and drones
  - Same basic framework for both
  - But need to learn go programming
- Turtlebot4's use ROS
  - So lets take a look at ROS

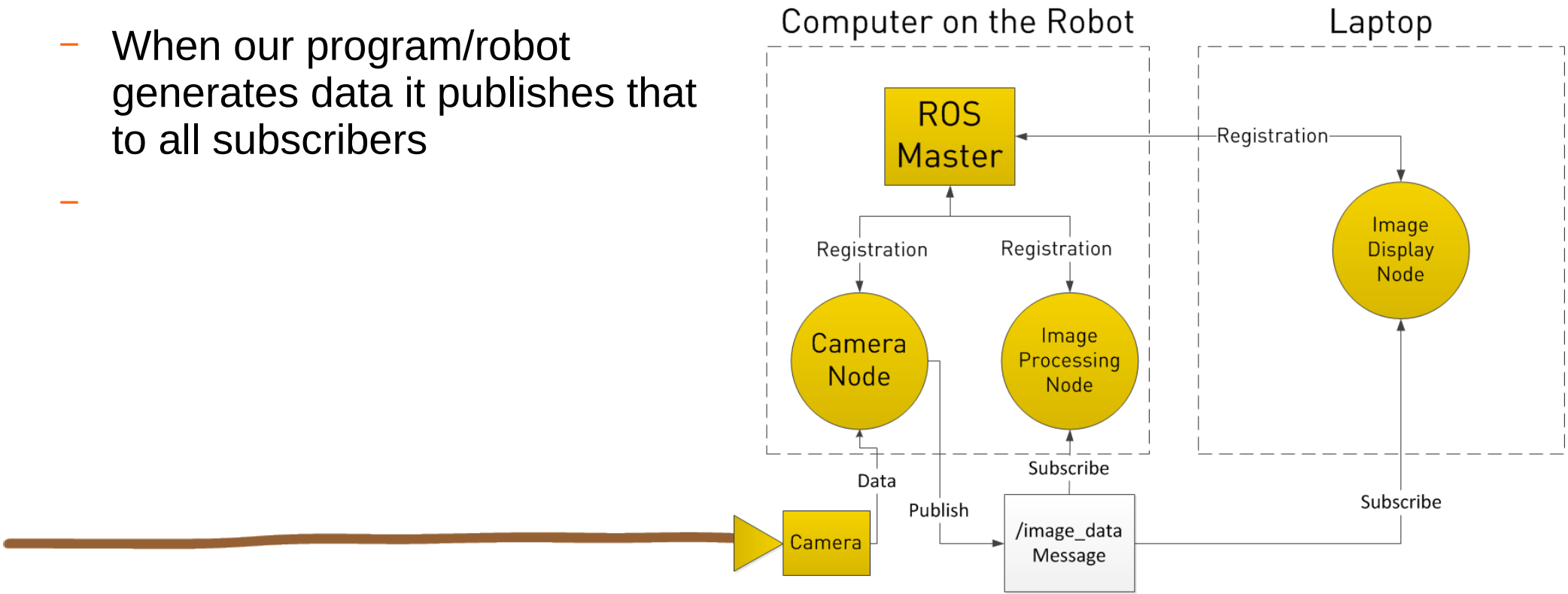
ROS Architecture

- ROS at its heart follows a publisher/subscriber model
  - Each ROS service is a node
  - In our case the 'ROS Master' is the discovery server living on the robot



ROS topic subscription

- To let the system know we want events when they occur we have our node subscribe
  - When our program/robot generates data it publishes that to all subscribers
  -



In Turtlebot4s

- Laptop
  - Running teleop twist
    - Creates a publisher for <your robot>/cmd_vel topic
  - Running rviz
    - Creates a subscriber for
      - <robot>/oakd/rbg/preview/image_raw
      - <robot>/scan
- Turtlebot4
  - Subscribes to <your robot>/cmd_vel
  - Publishes <robot>/oakd/rbg/preview/image_raw
  - Publishes <robot>/scan

Robots and people

- Wearable robotics
  - What are the ethical issues in wearable robotics
  - https://www.youtube.com/watch?v=6T9CYkJ_hwg
  - https://www.youtube.com/watch?v=VEvV1jnqacs
  - https://www.youtube.com/watch?v=0vPVNHxjER0
  - Prosthetic arms
  - https://www.youtube.com/watch?v=I8hSv244oXs