



UNIVERSITY OF
SOUTH CAROLINA

CSCE 574 ROBOTICS

ROS overview

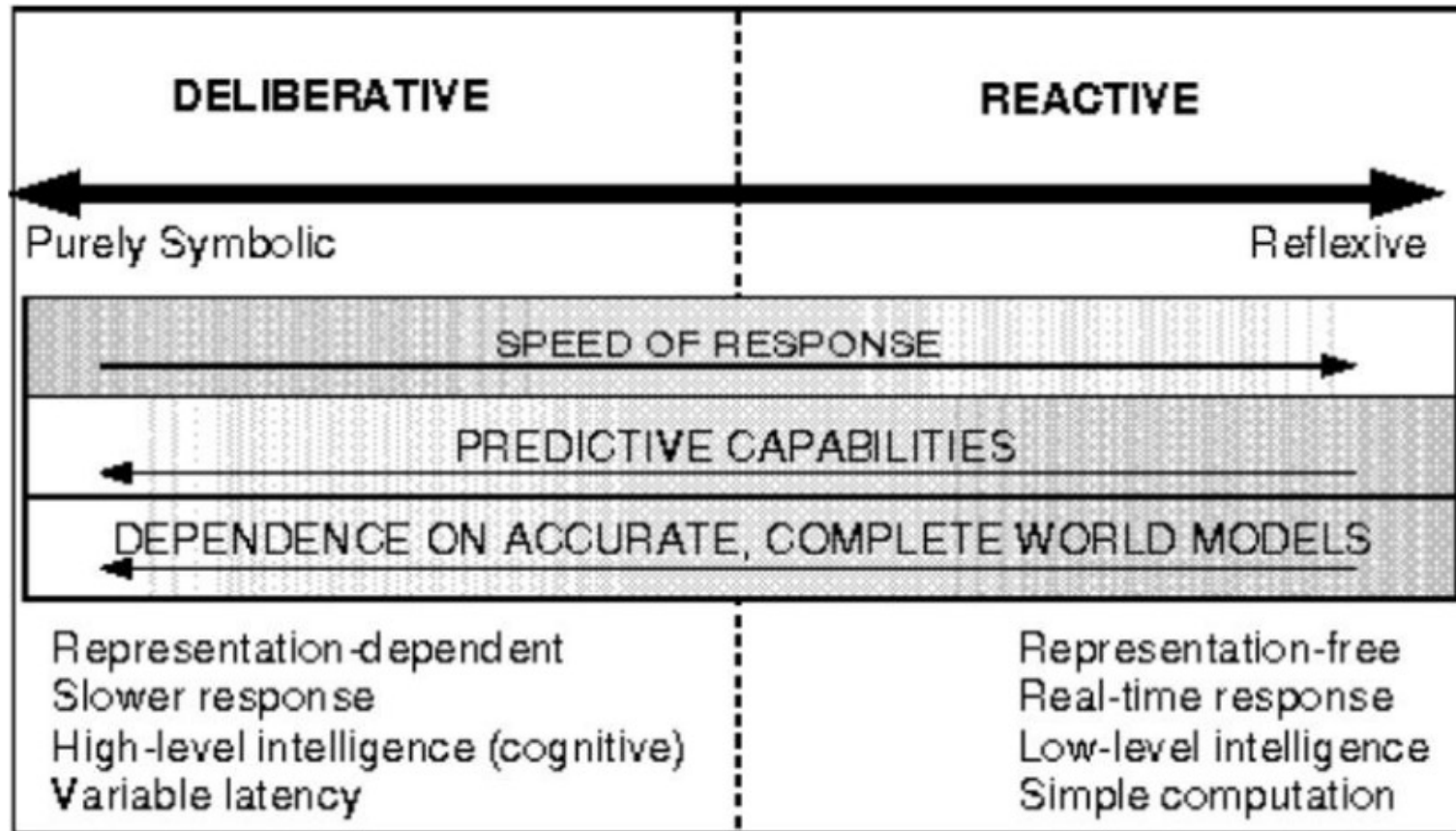
Ioannis REKLEITIS

Computer Science and Engineering

University of South Carolina

yiannisr@cse.sc.edu

Spectrum of control



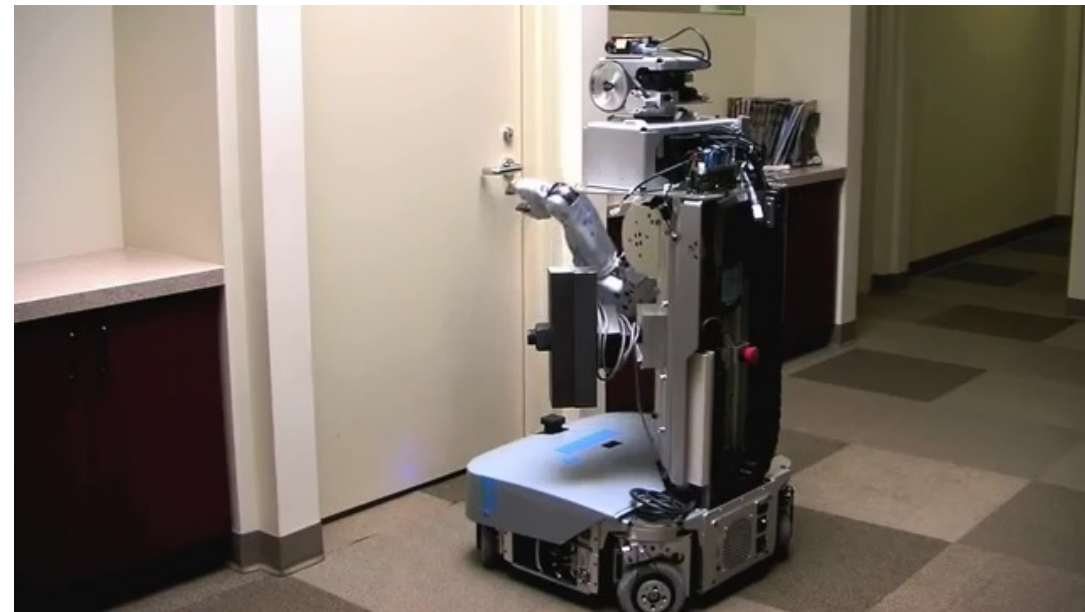
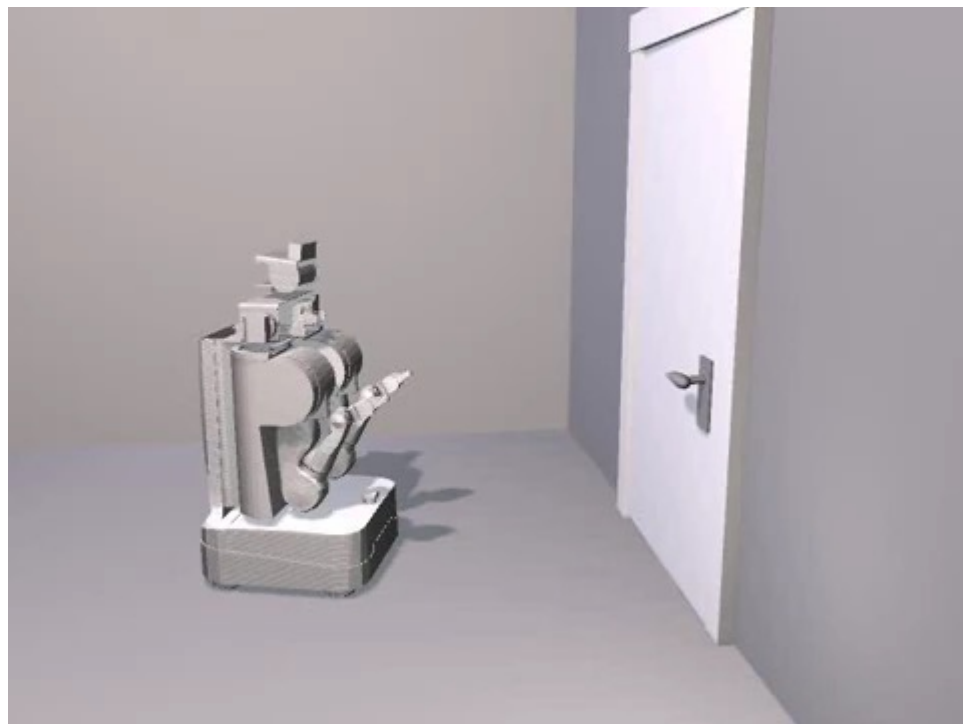
Source: [Arkin, 1998, MIT Press]



Middleware



WillowGarage PR2 (2007)



Source: WillowGarage

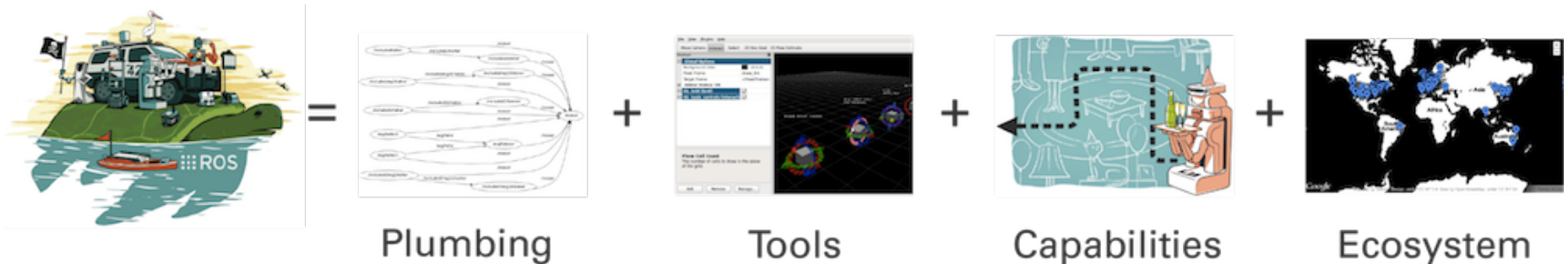


ROS



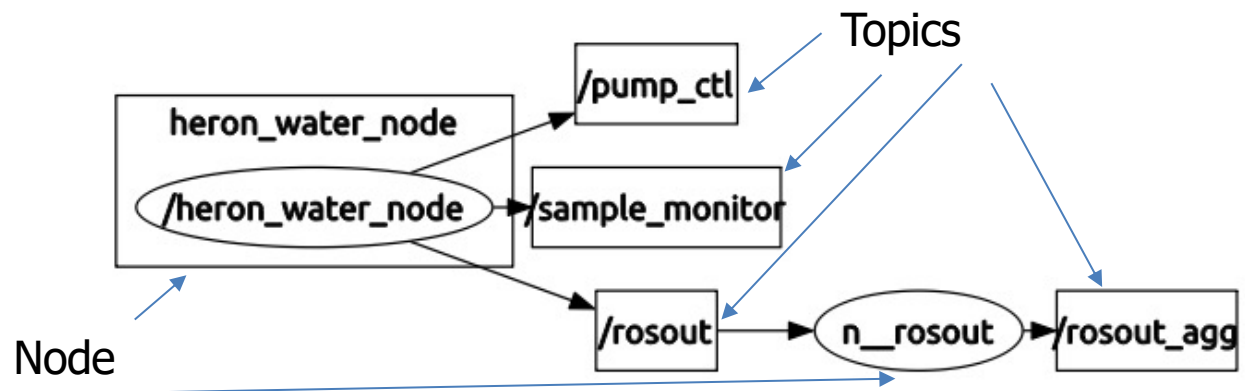
ROS

- The Robot Operating System (ROS) is a flexible framework for writing robot software
 - It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms
- Developed and Maintained by the Open Source Robotics Foundation (OSRF)
- “The primary goal of ROS is to support *code reuse* in robotics research and development.”



ROS

- ROS is based on publish/subscribe message passing approach
- The core elements are:
 - ROS master: process that provides naming and registration to the rest of the nodes
 - Nodes: processes implementing robotic components
 - Topics: named buses over which nodes exchange messages



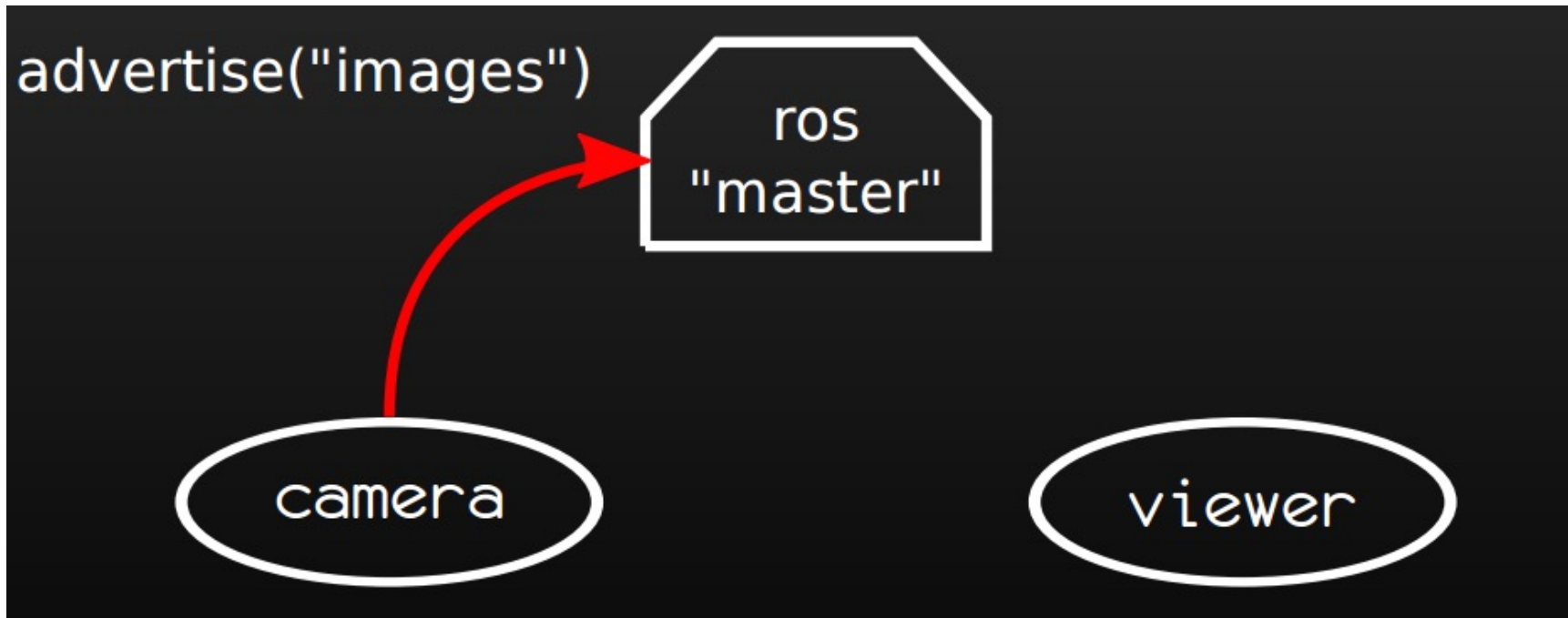
ROS overview

- Process for making two nodes interact with each other



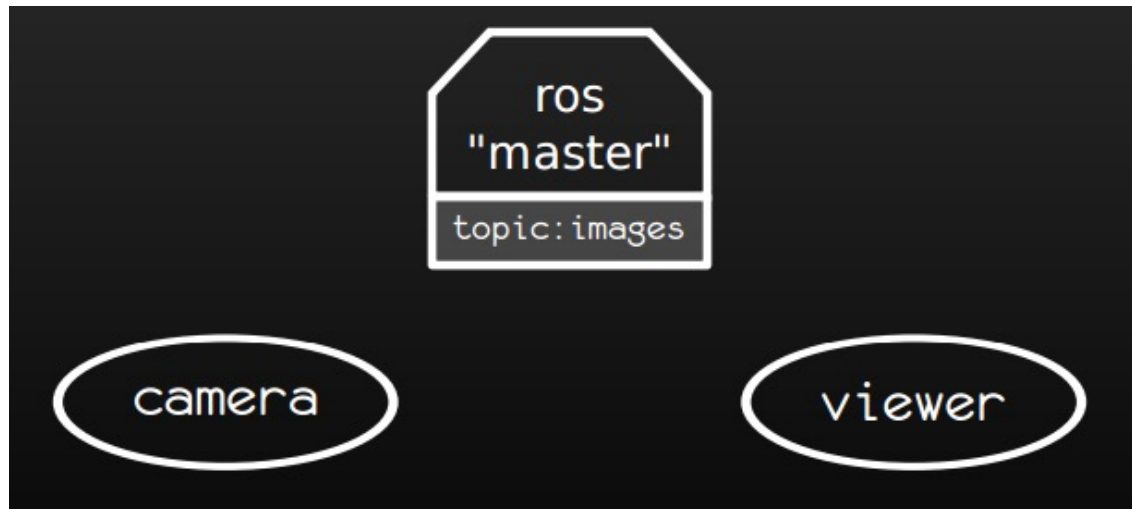
ROS overview

- Process for making two nodes interact with each other



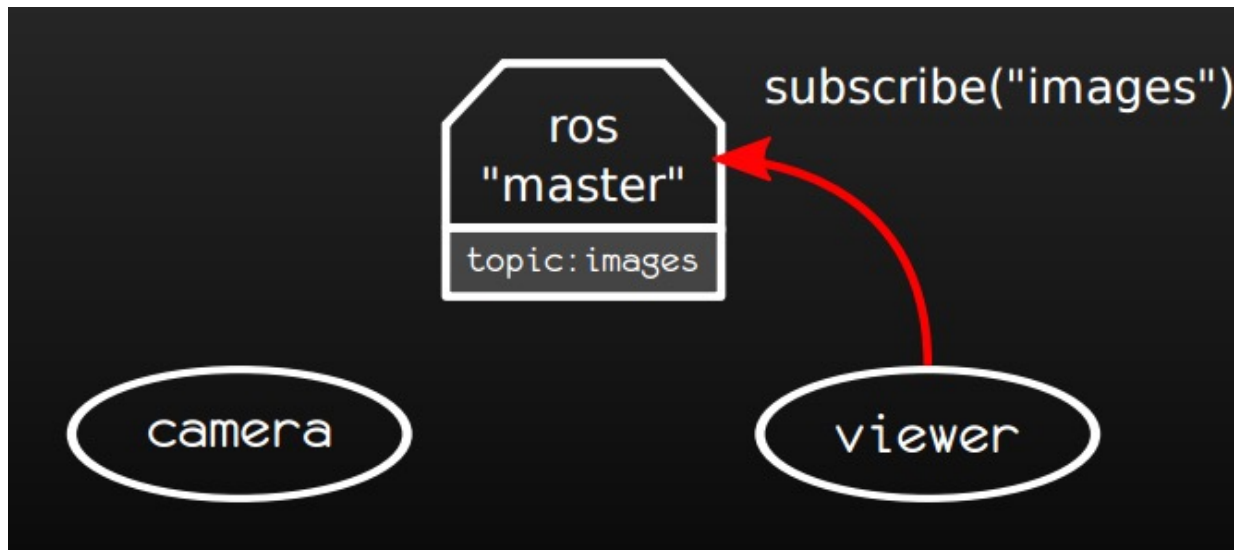
ROS overview

- Process for making two nodes interact with each other



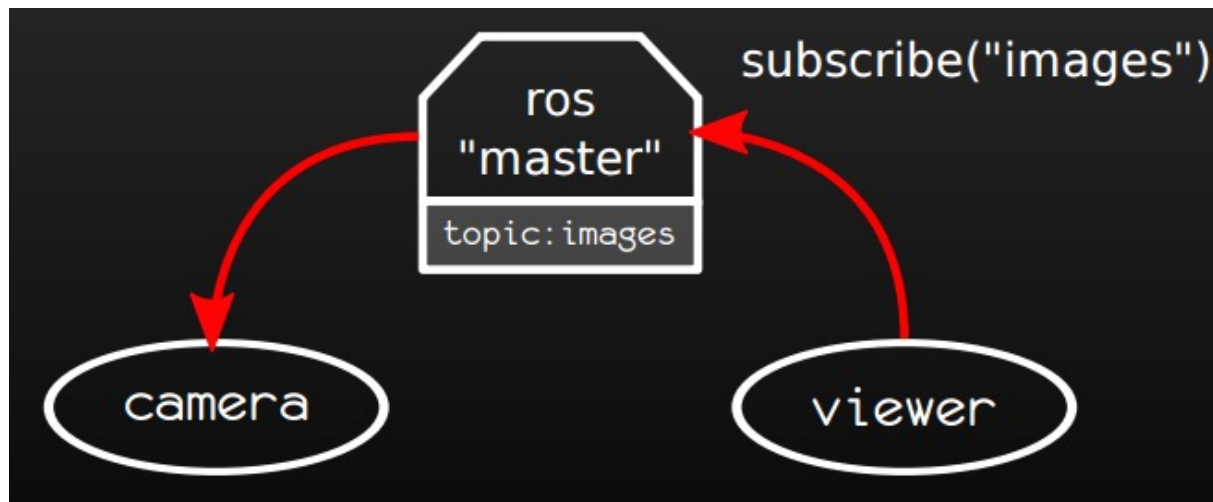
ROS overview

- Process for making two nodes interact with each other



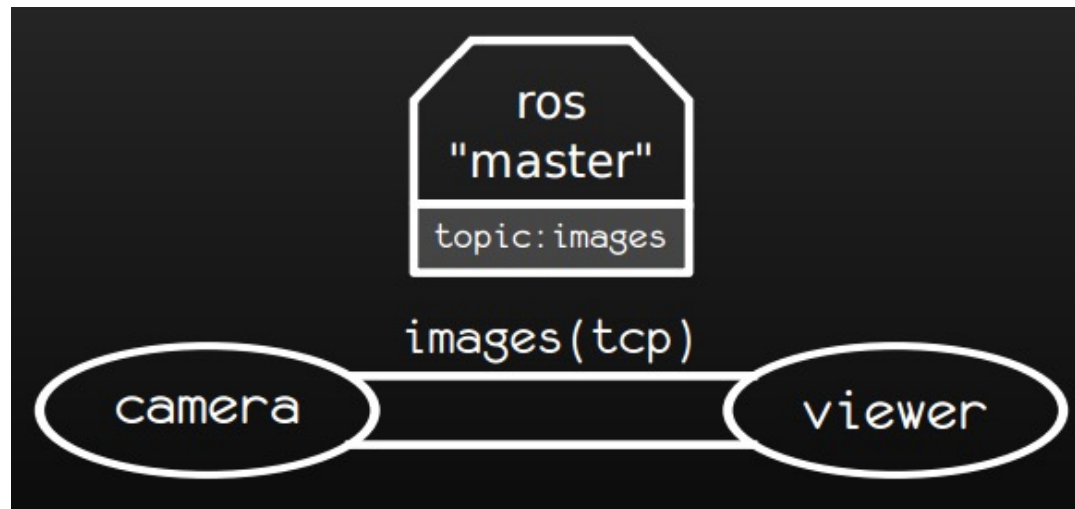
ROS overview

- Process for making two nodes interact with each other



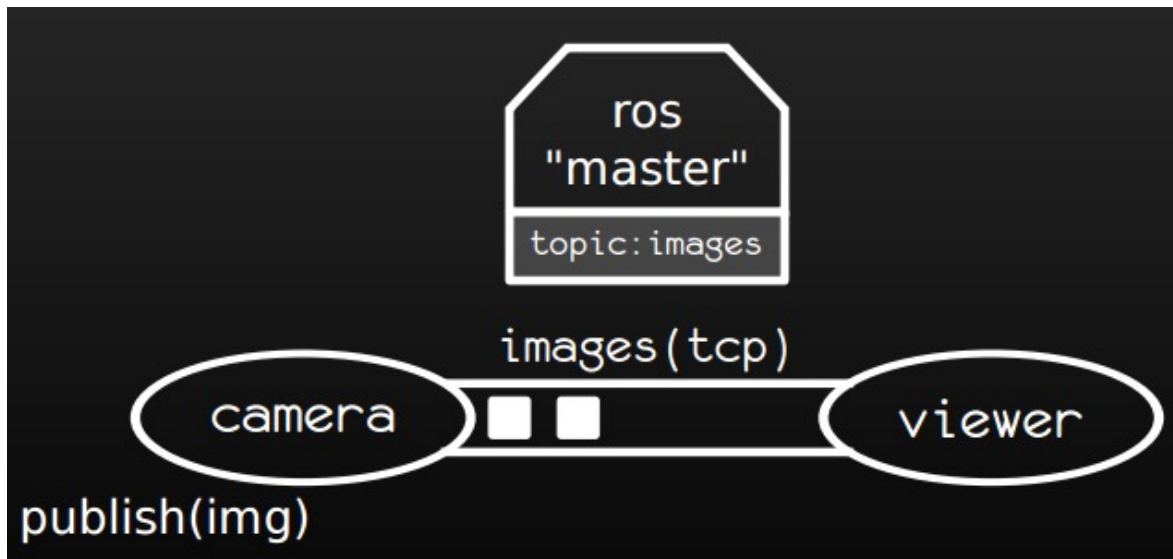
ROS overview

- Process for making two nodes interact with each other



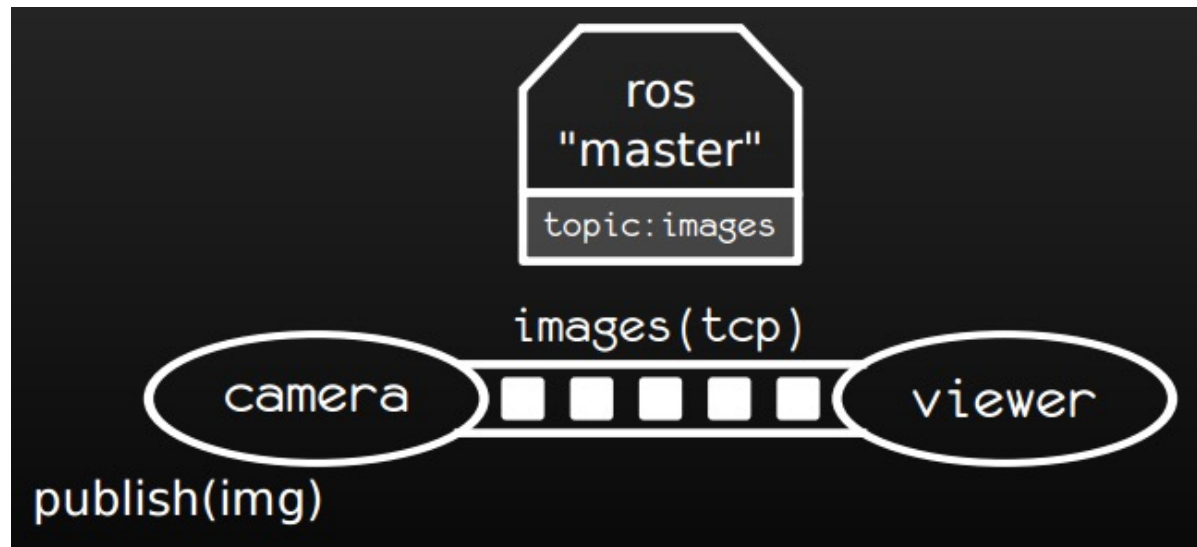
ROS overview

- Process for making two nodes interact with each other



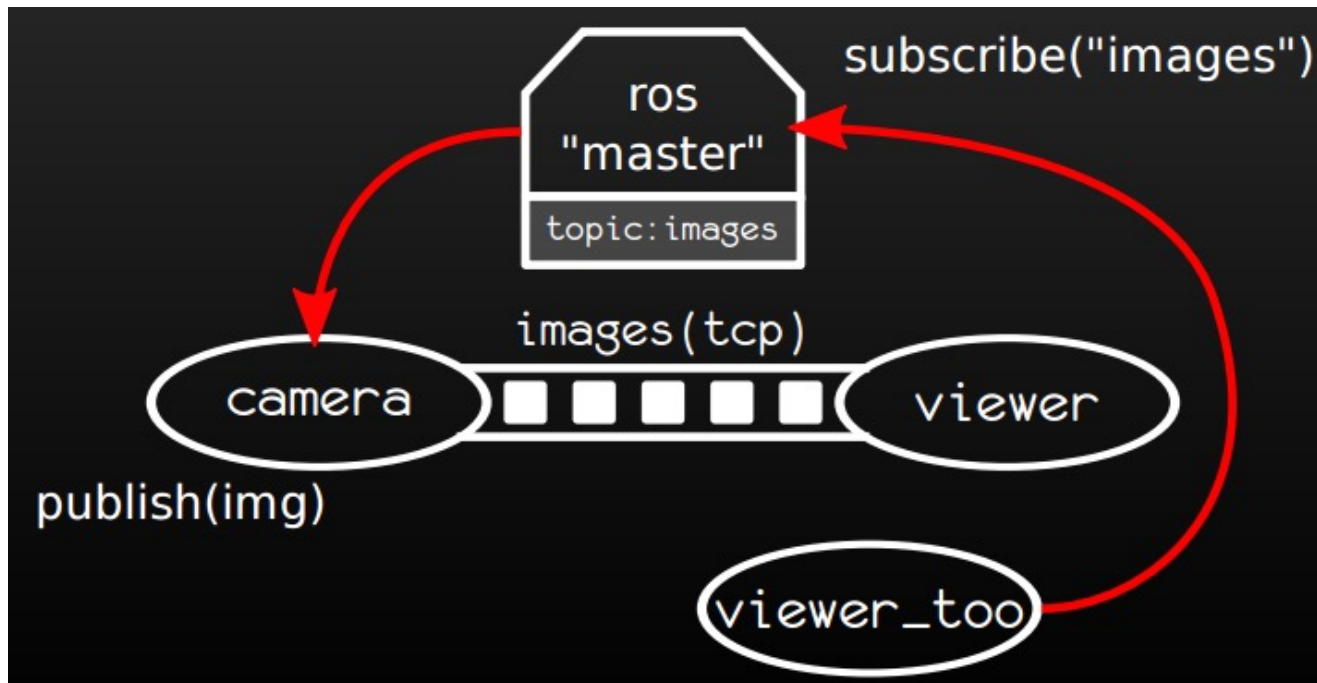
ROS overview

- Process for making two nodes interact with each other



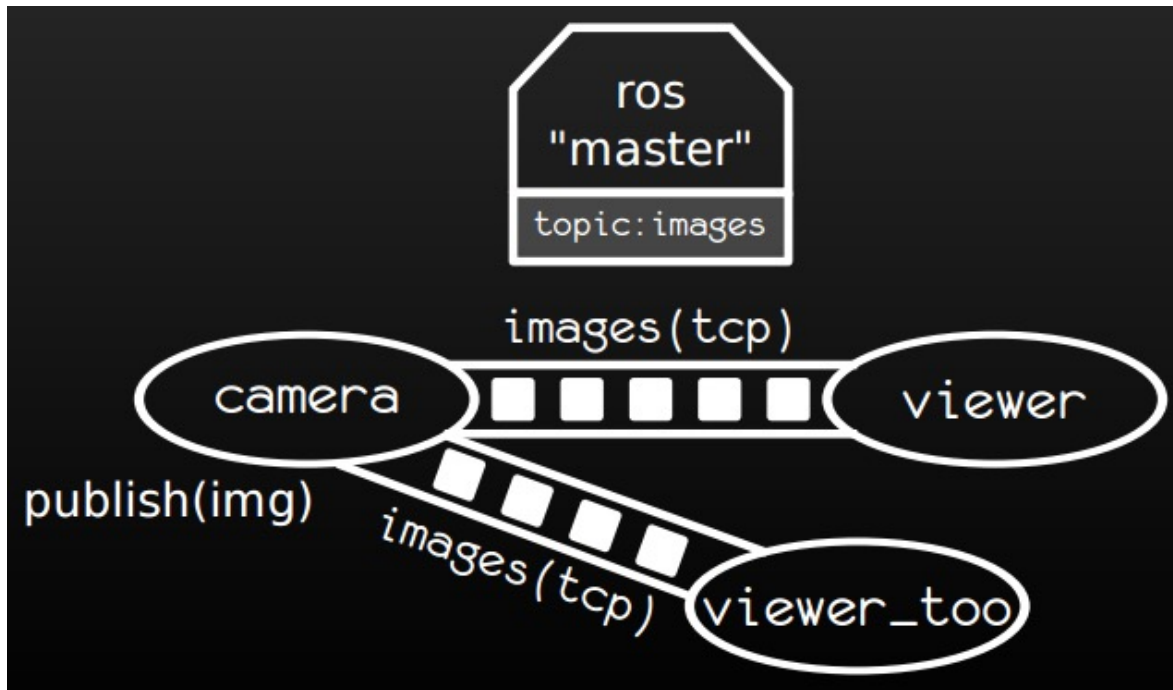
ROS overview

- Process for making two nodes interact with each other



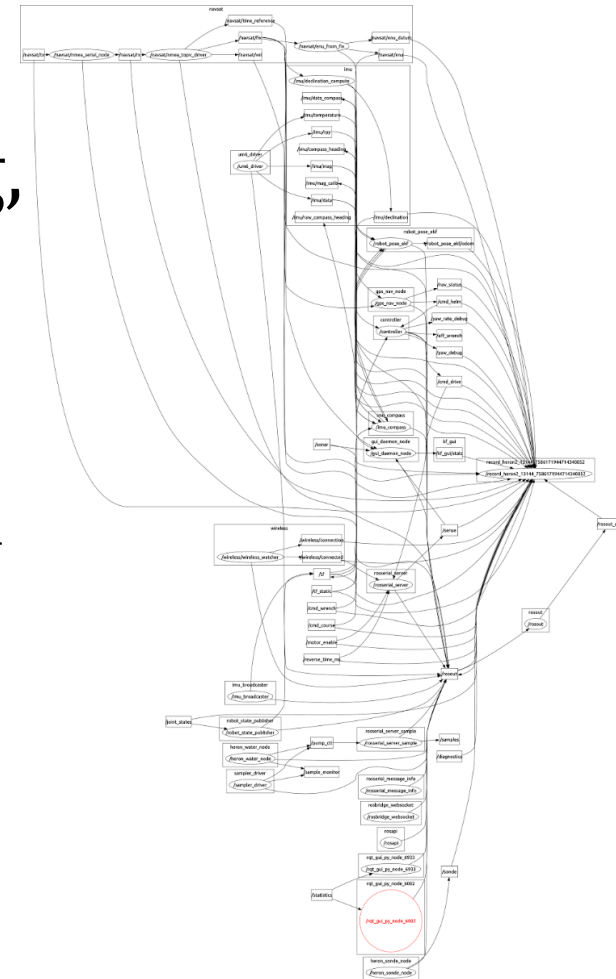
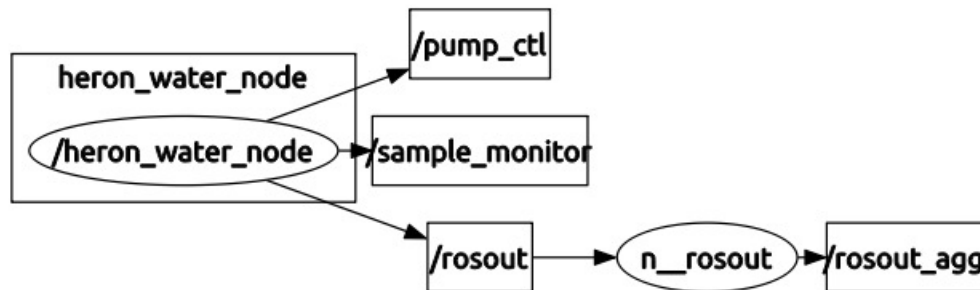
ROS overview

- Process for making two nodes interact with each other



ROS overview

- Typically, a node represents one task (driver, localization, mapping, path planning, ...)
- Nodes run in parallel
- To debug problems, use rqt_graph



ROS overview

- The main mean of communication in ROS are topics and messages
- However, there are other ways for nodes to communicate with each other
 - Services: similar to Remote Procedure Calls
 - Actionlib: preemptable tasks



ROS overview

- How to decide what to use:
 - Topics: especially for stream of data
 - Services: execution of fast tasks
 - Actions: execution of tasks that need to be tracked and should be preempted in some cases



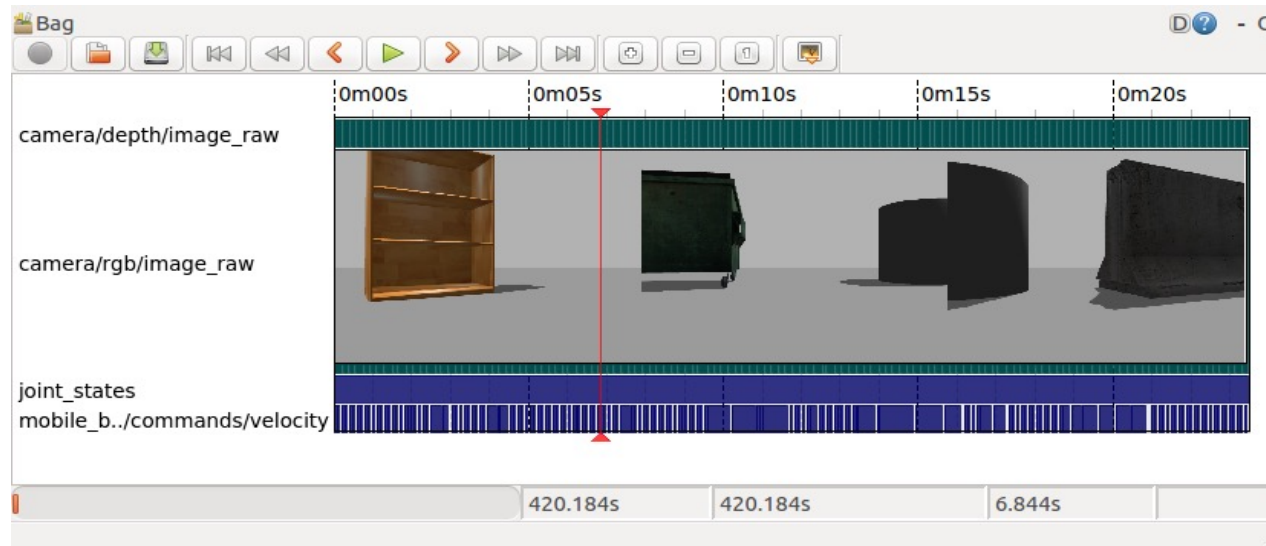
ROS overview

- Parameters can be easily set
 - Statically: `rosparam`
 - Dynamically: `rqt_reconfigure`
- Be careful in which namespace the parameters are defined: global or private



ROS overview

- Logging data streams can be achieved by using rosbag
- Remember the ROS parameter `sim_time` especially to run algorithms on bag files

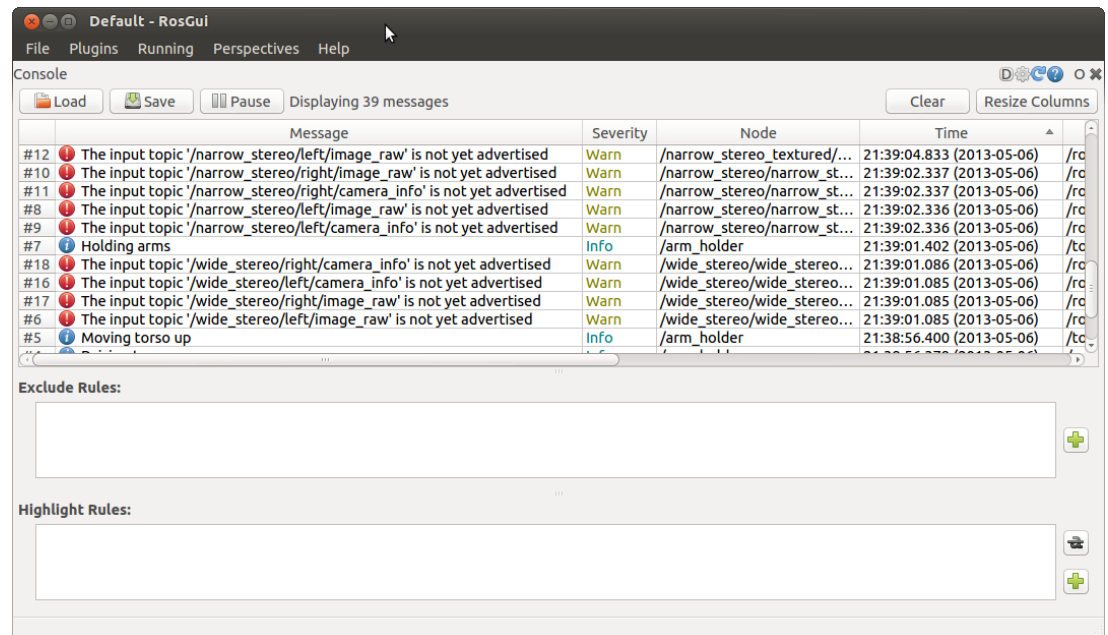


Source: ros.org



ROS overview

- Logging messages are published in rostopic topic
 - Different log levels should be used according to the severity of the message
- rqt_console can be used to visualize them

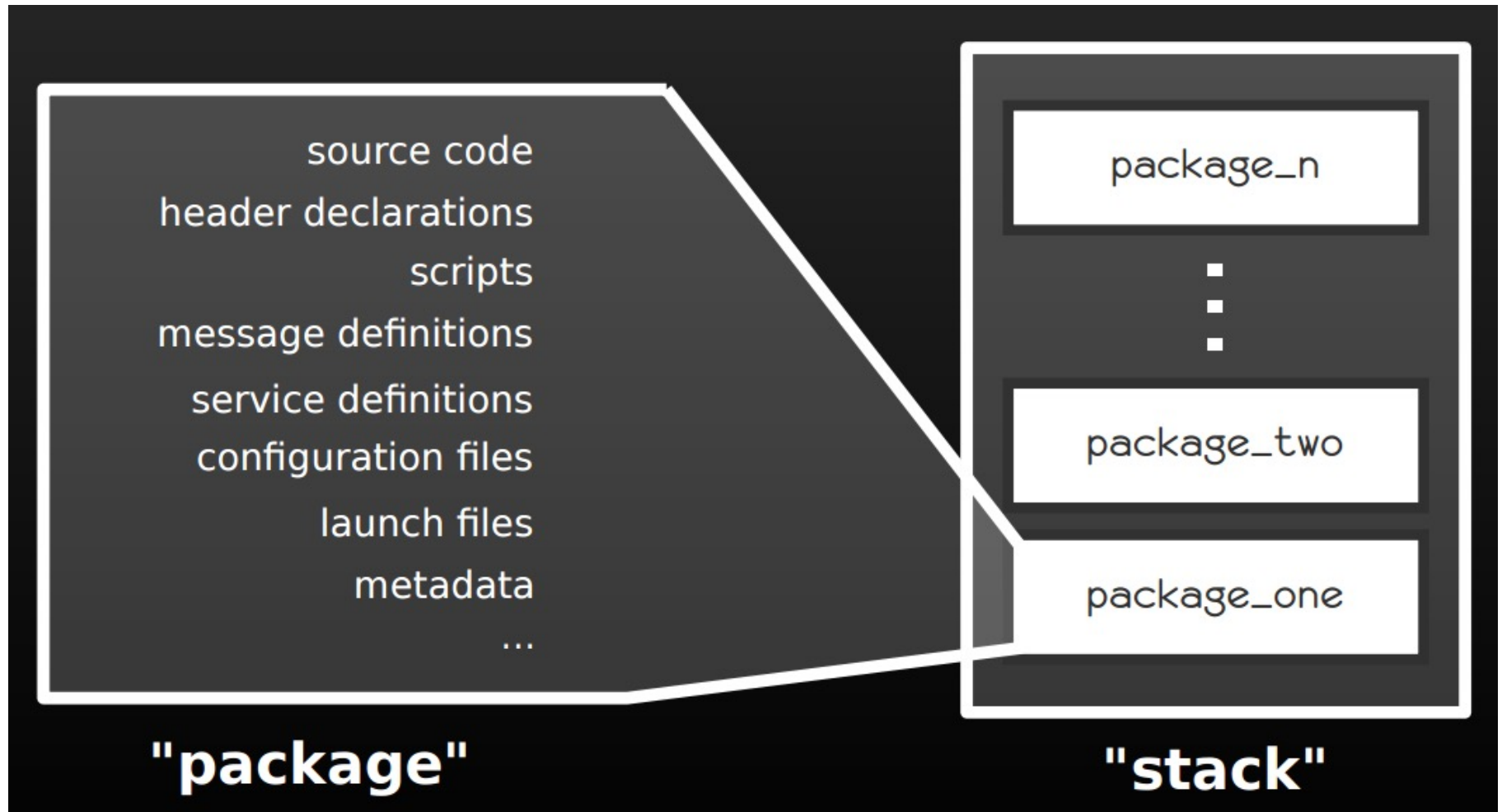


ROS tools

- **Navigate:** roscd, rosls
- **Setup:** catkin_init_workspace, catkin_create_pkg
- **Configure:** package.xml, CmakeLists.txt
- **Build:** catkin_make
- **Execute:** roscore, rosrn, roslaunch, rosparam, rqt_reconfigure
- **Inspect:** rosnod, rostopic, rosservice
- **Debug:** rqt_graph, rostest, rqt_plot
- **Log & Analyze:** rosbag, rqt_bag, rqt_console



ROS packages

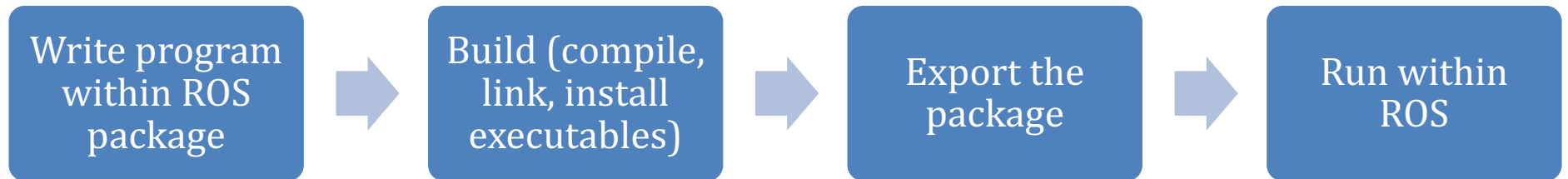


ROS distribution



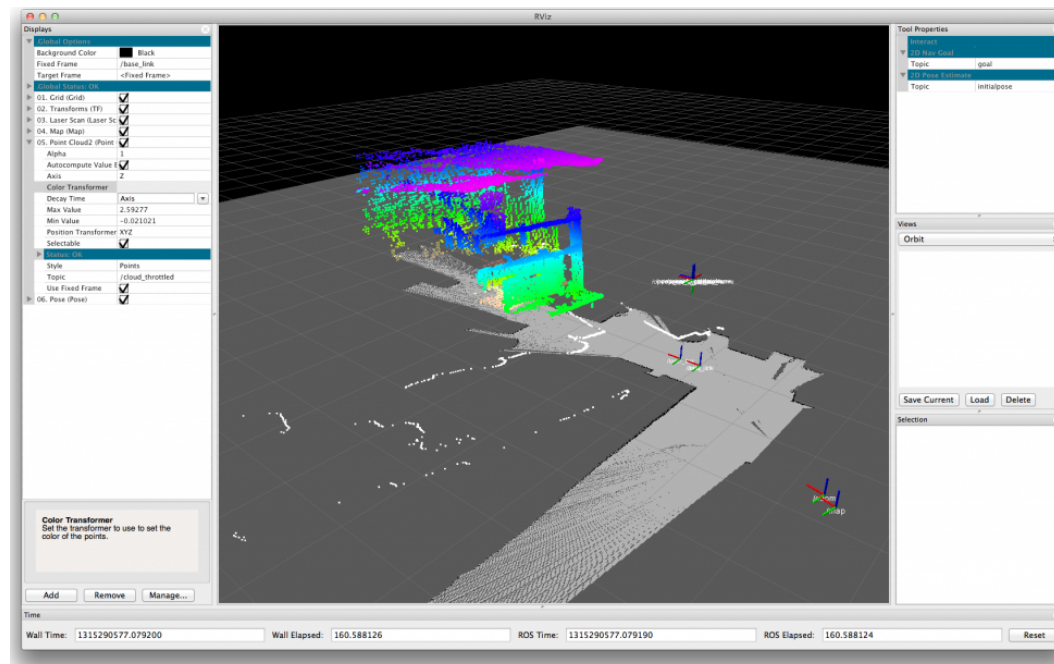
- Note that in Ubuntu, there are many packages ready just to be installed with `sudo apt-get install`

From source code to executable



ROS tools

- rviz can be used to visualize data



Source: iheartrobotics.com



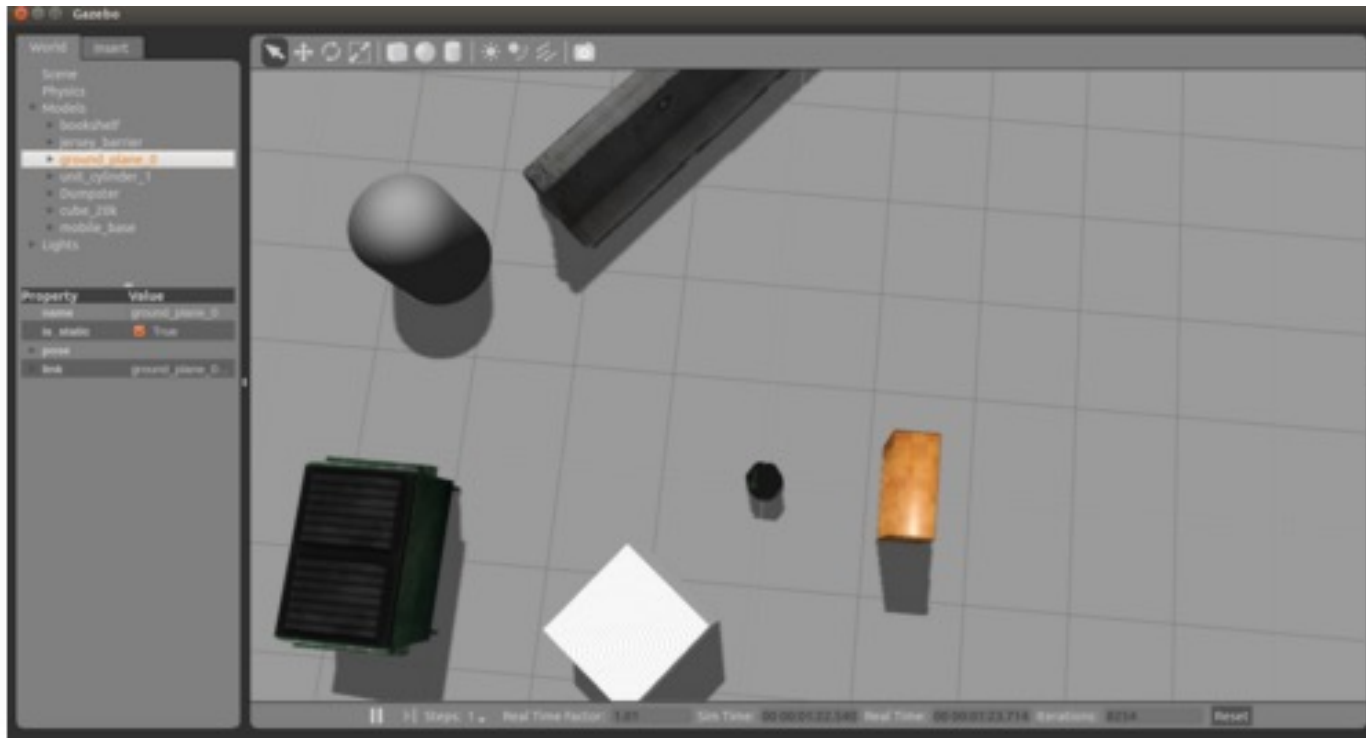
ROS tools

- Stage, 2D simulator



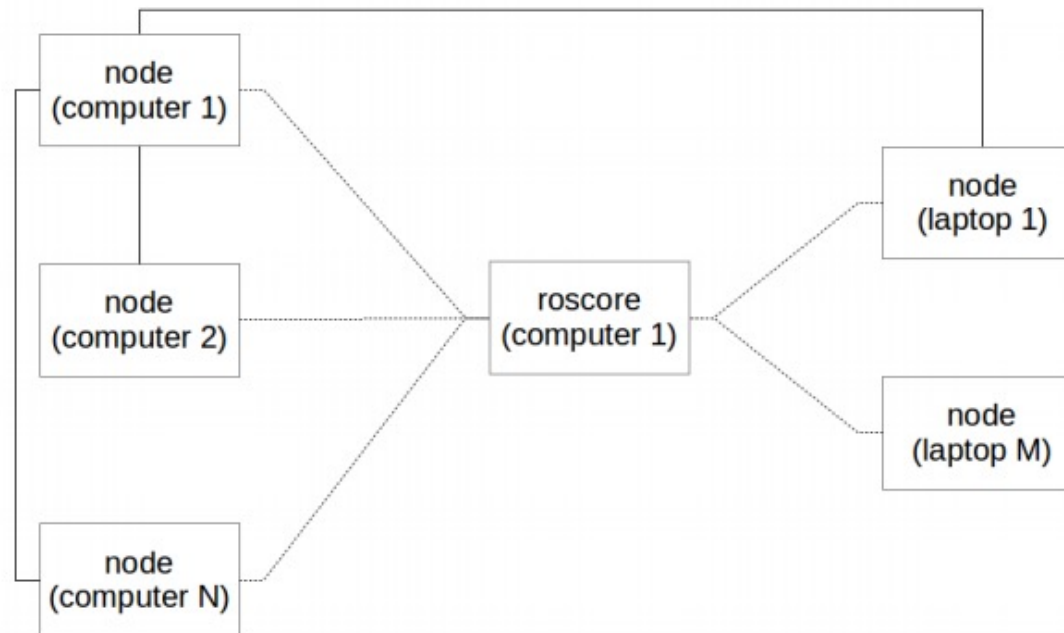
ROS tools

- Gazebo, 3D simulator



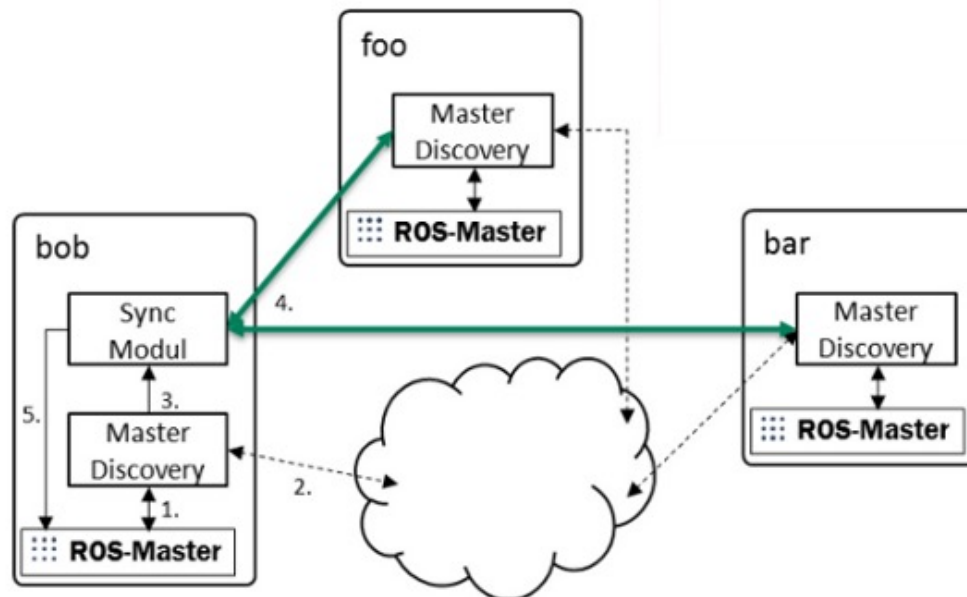
ROS overview

- In a multirobot settings, a possibility is to share the ROS master over all of the computers



ROS overview

- However, to have the system more robust, *multi-master-fkie* can be used to allow robots to see other ROS masters



Summary

- Problems to solve in robotics
 - Localization, Mapping, Planning
- Robot software architectures with some history
 - Deliberative
 - Reactive
 - Hybrid
- Middleware
 - ROS
 - Core elements
 - Packages
 - Tools



Questions?

