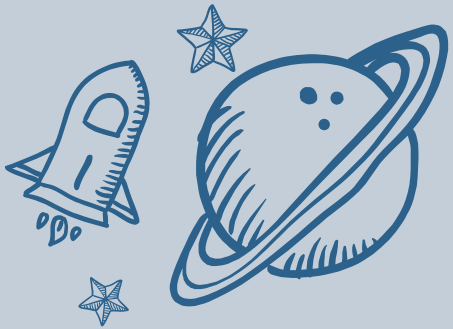
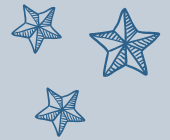
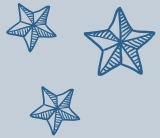
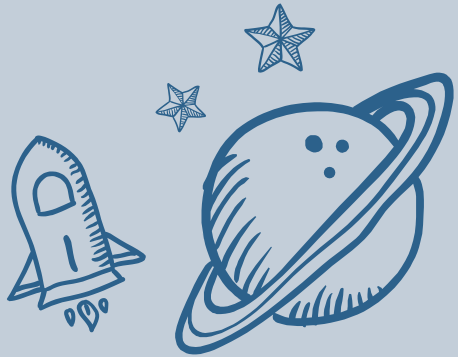




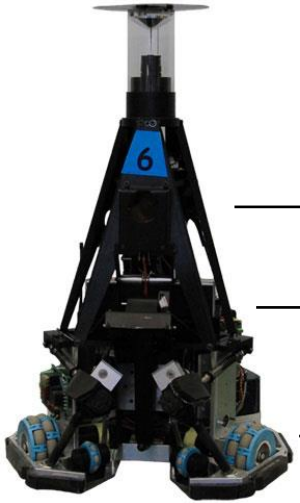
Robot Operating System





Let me introduce myself:

General Robot parts



Program

Electric

Mechanic

Introduction#1

ROS

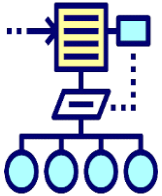
- provides libraries and tools to help software developers create robot applications
- See more: http://wiki.ros.org/

Introduction#2

Why using ROS?

Rapid testing

Has several ways to efficient test of software robots.



Software reuse

good algorithms can be implemented to each new system.

Distributed computation

Robot can runs and treat several programs together

Introduction#3

ROS is *NOT*

Programming language

ROS programs routinely written in C++, Python and other languages.

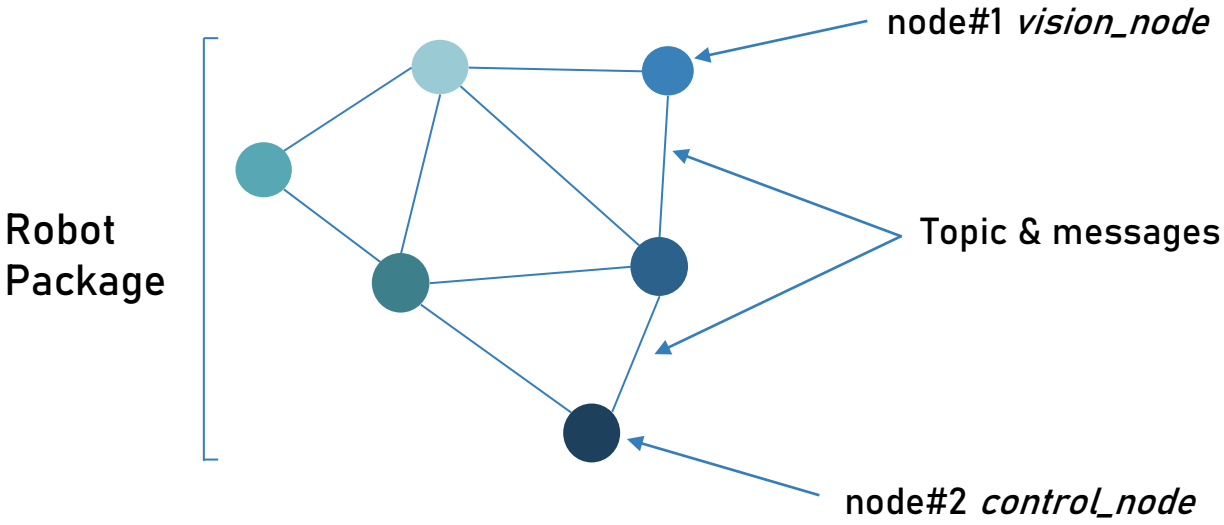
(Only) a library

ROS also include central server, set of command-line, graphical tools, and build system.

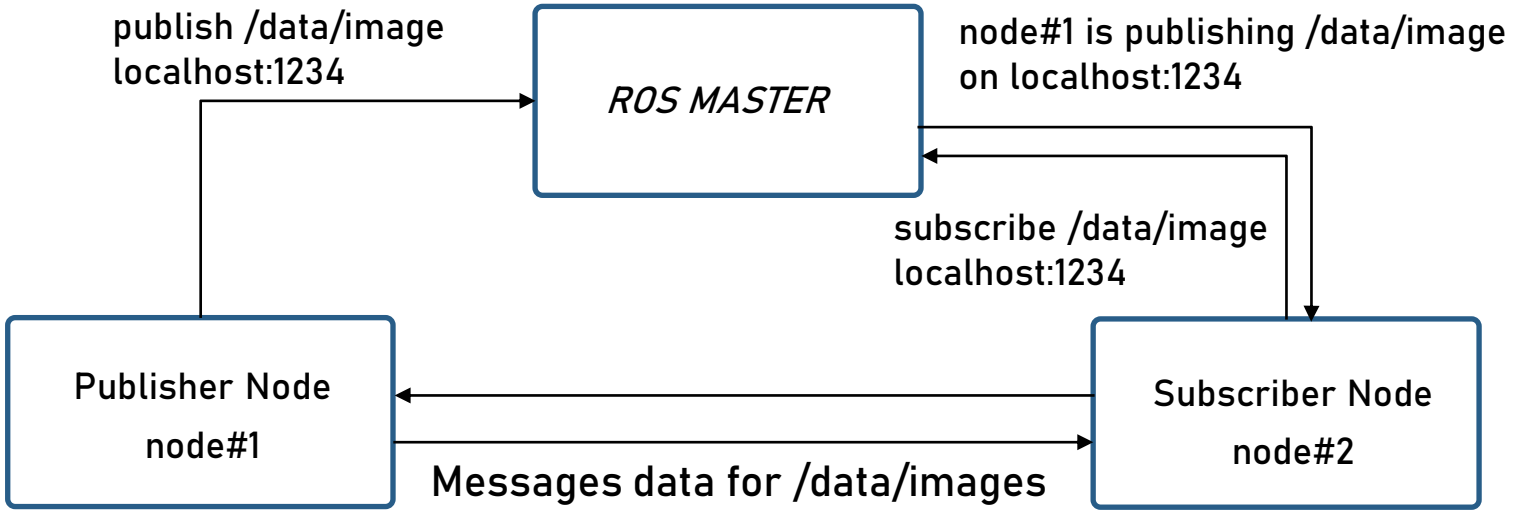
IDE Programming

ROS does not require any IDE. It can be used with most popular IDE.

Communication Structure



Communication Structure



ROS Package

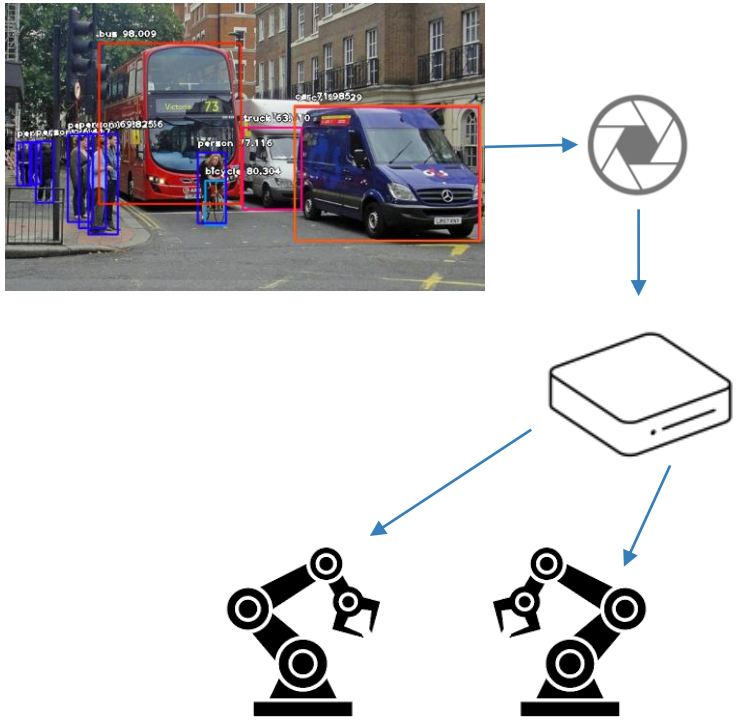
Robot Package

coherent collection of files, generally including both executables and supporting files, that serves a specific purpose.

The goal of these packages it to provide this useful functionality in an easy-to-consume manner so that software can be easily reused

See more: http://wiki.ros.org/Packages

ROS Node



A node is a process that performs computation. Combined together into a graph and communicate with one another using streaming topics.

A robot control system will usually comprise many nodes. For example, one node control robot's wheel motor, one node perform localization, one node provide graphical view of system, and so on

See more: http://wiki.ros.org/Nodes

Master

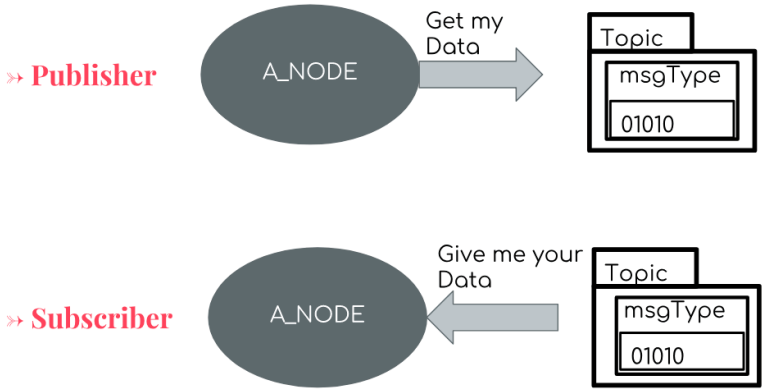
Provide naming and registration service to the rest of all nodes.

The role of the Master is to enable individual ROS nodes to locate one another. Once these nodes have located each other they communicate with each other peer-to-peer.

Using command *roscore* to activate master.

See more: http://wiki.ros.org/Master

Message and Topics



Nodes communicate to each other by send a *messages*, which organized into named *topics*.

When publisher node sent messages in specific topics, subscriber node would subscribe in *same* topics. The message themselves are sent directly from publisher to subscriber.

See more: A Gentle Introduction To ROS

THANK YOU