ECE 4703
Mobile Autonomous Robots

Jenny Zhen Yu
zhenyu@cpp.edu
Department of Electrical and Computer Engineering
California State Polytechnic University, Pomona

Lecture 2: ROS Concepts

Outline

- **Mobile Autonomous Robots Introduction**
  - ROS terminology
  - Message communication
  - Message
  - Name
  - Coordinate transformation (TF)
  - ROS commands
  - ROS tools

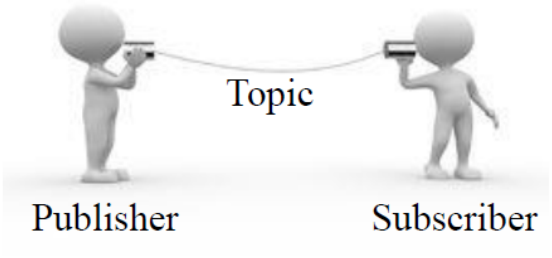
ROS Terms

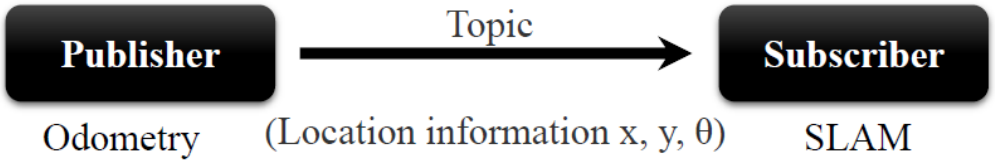
- Node
- The smallest unit of executable processors. It can be regarded as single executable program. In ROS, a system is consist of many nodes. Each node transmits and receives data by message communication.

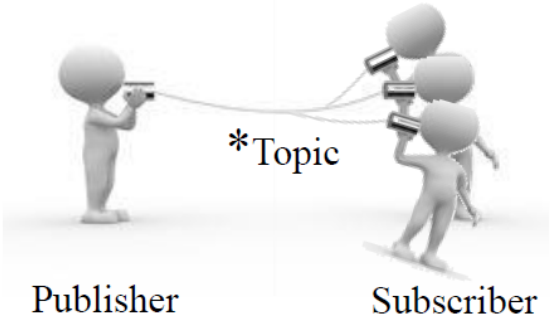
- Package
- One or more nodes, information for node execution, etc. Also, bundles of packages are called as metapackages.

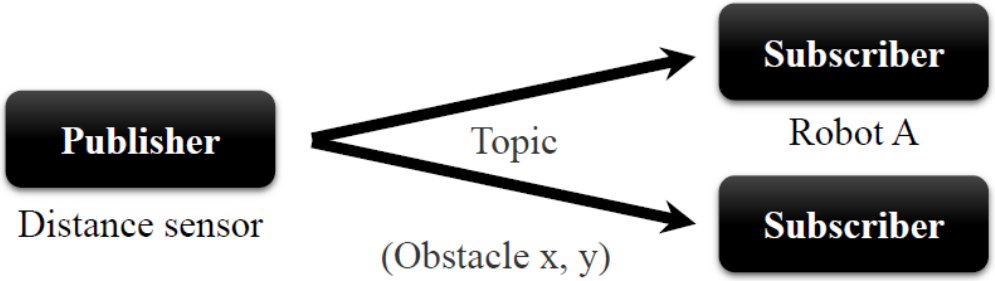
- Message
- Data is transmitted and received through message between nodes. Messages can have various types such as integer, floating point, and boolean. You can also use structures such as a simple data structure and an array of messages that hold messages in the message.

Topic, Publisher, Subscriber





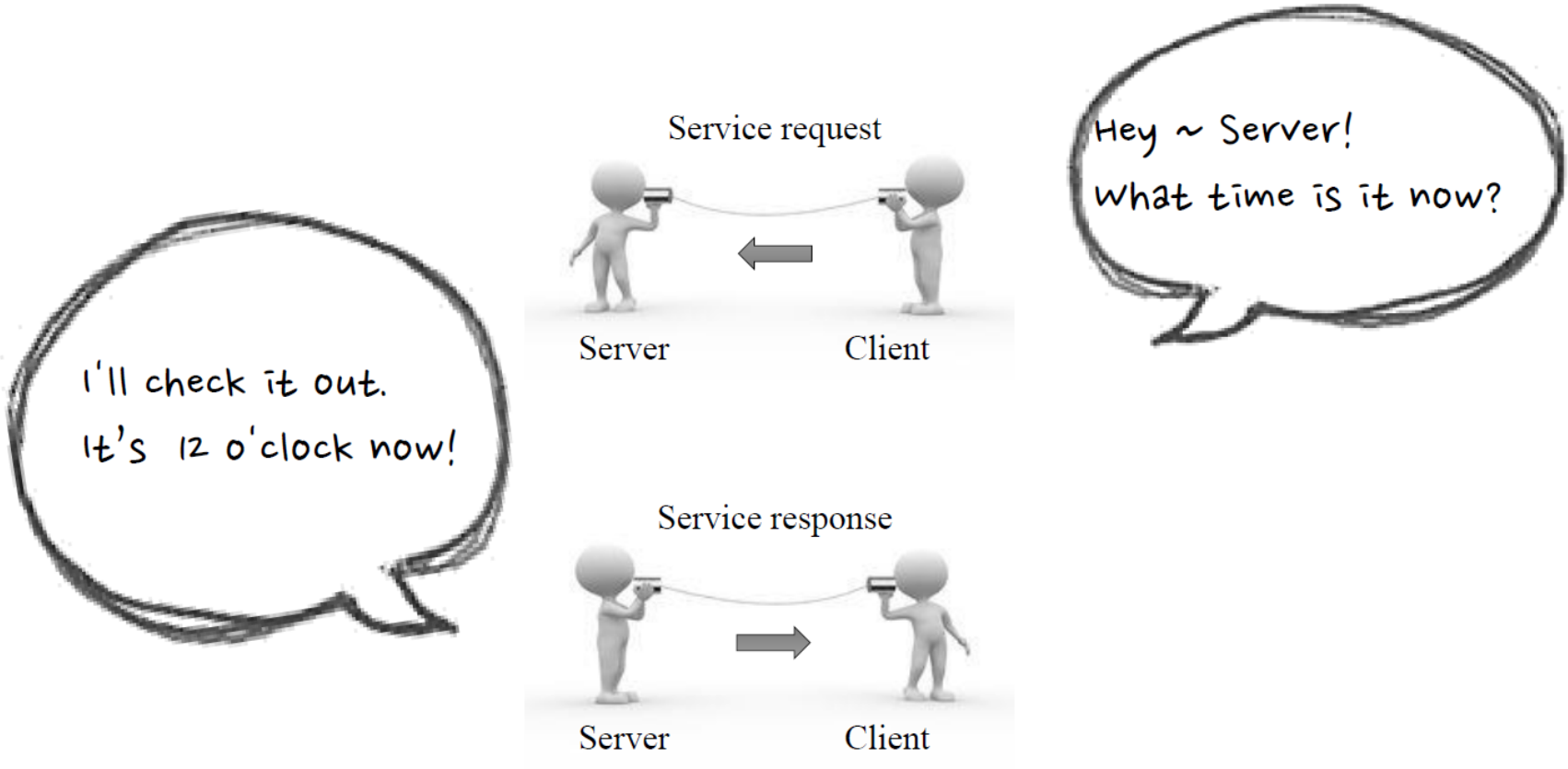




* 1: 1 Publisher and Subscriber communication is also possible for Topic, and 1: N, N: 1, N: N communication is also possible depending on the purpose.

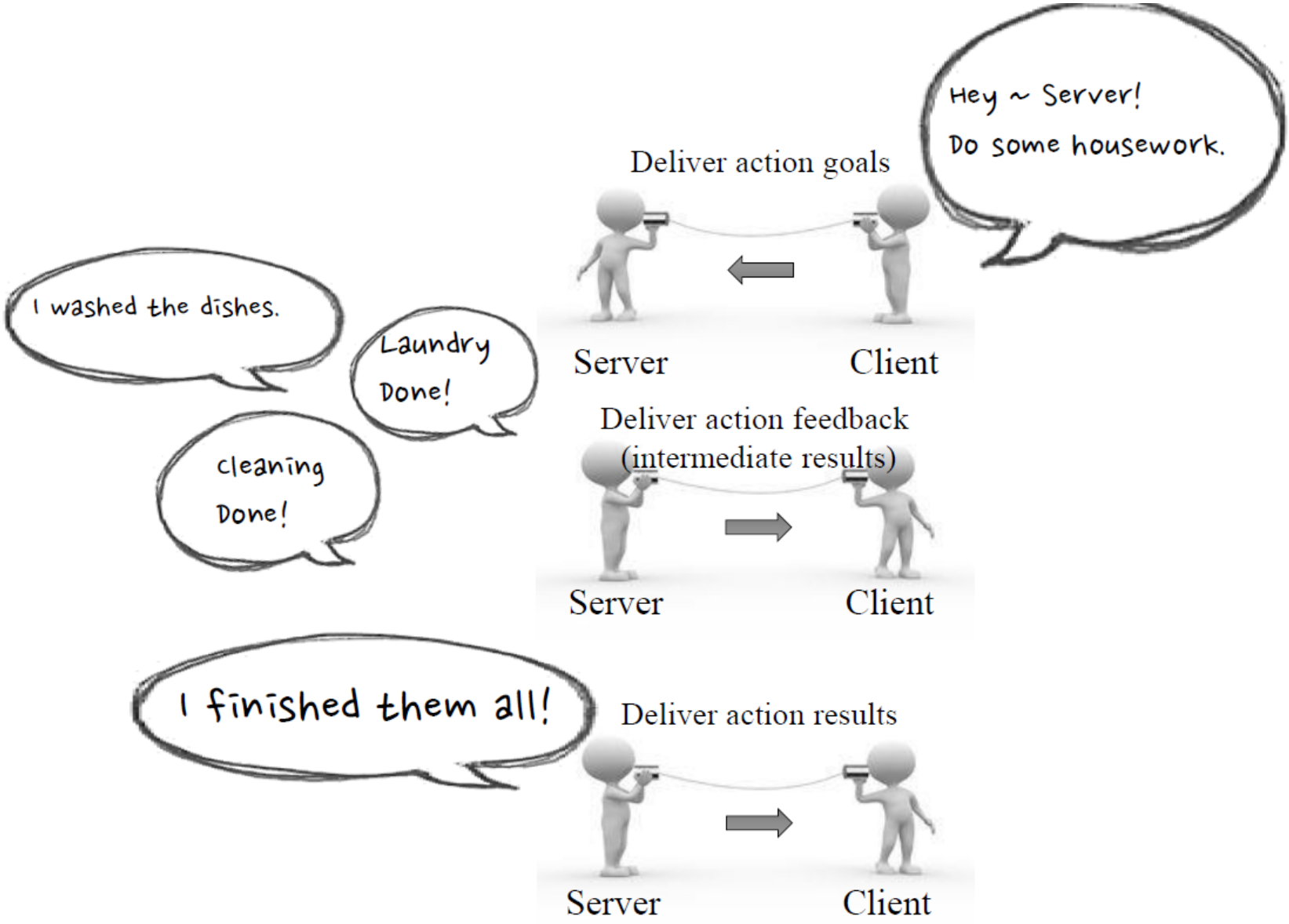
http://www.dreamstime.com/illustration/people-talk-listen-tin-can-phone-communication.html

Service, Service server, Service client



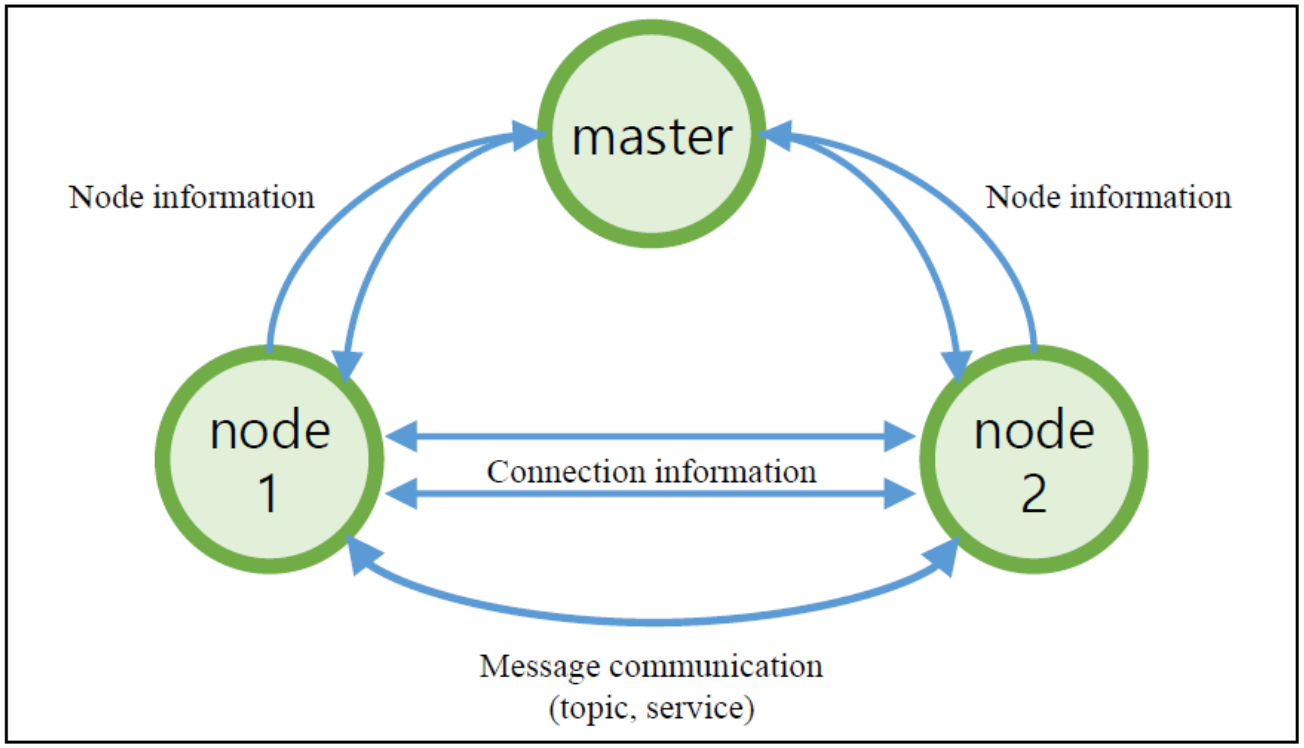
http://www.dreamstime.com/illustration/people-talk-listen-tin-can-phone-communication.html

Action, Action server, Action client



Message Communication

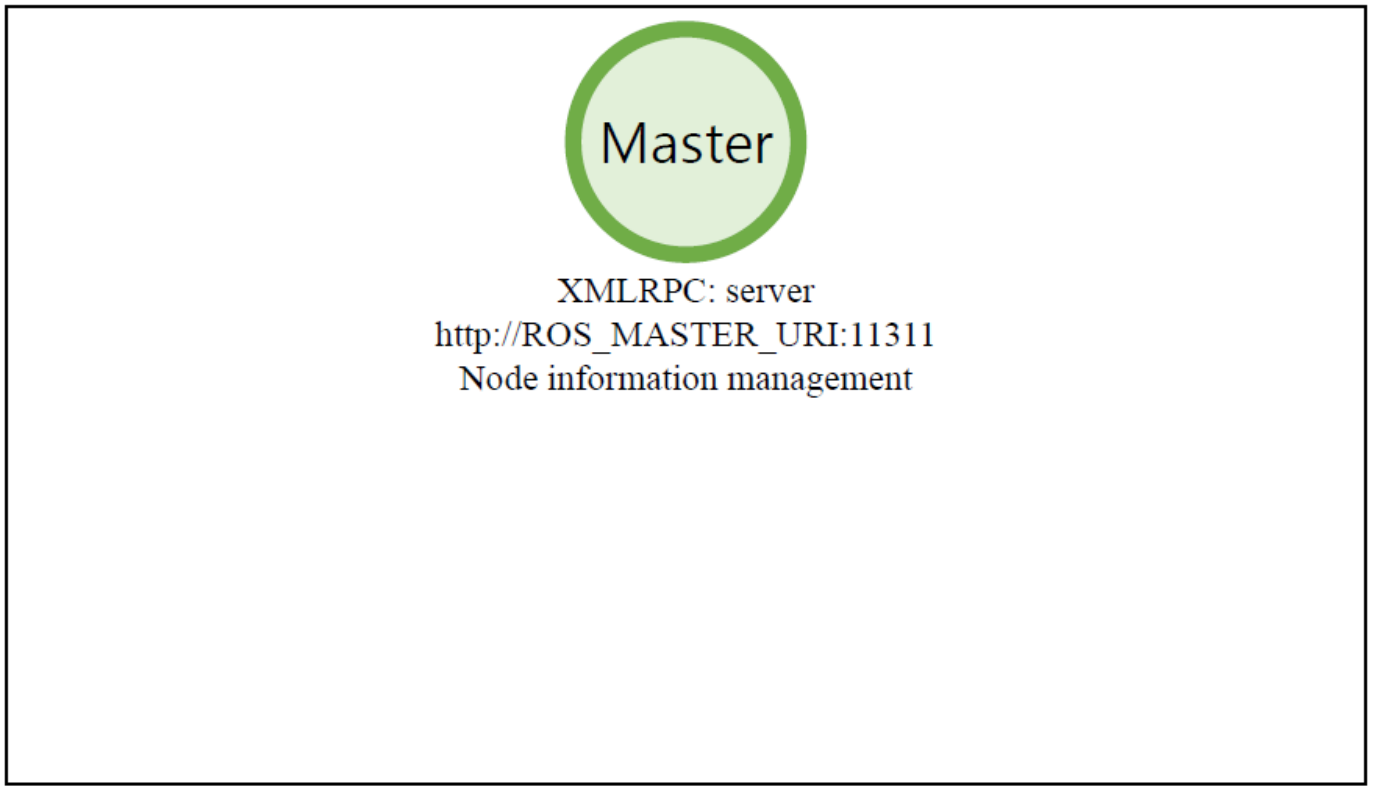
- The most fundamental technical point of ROS: **message communication among nodes!**



Message Communication

1. Run Master: XMLRPC(XML-Remote Procedure Call)

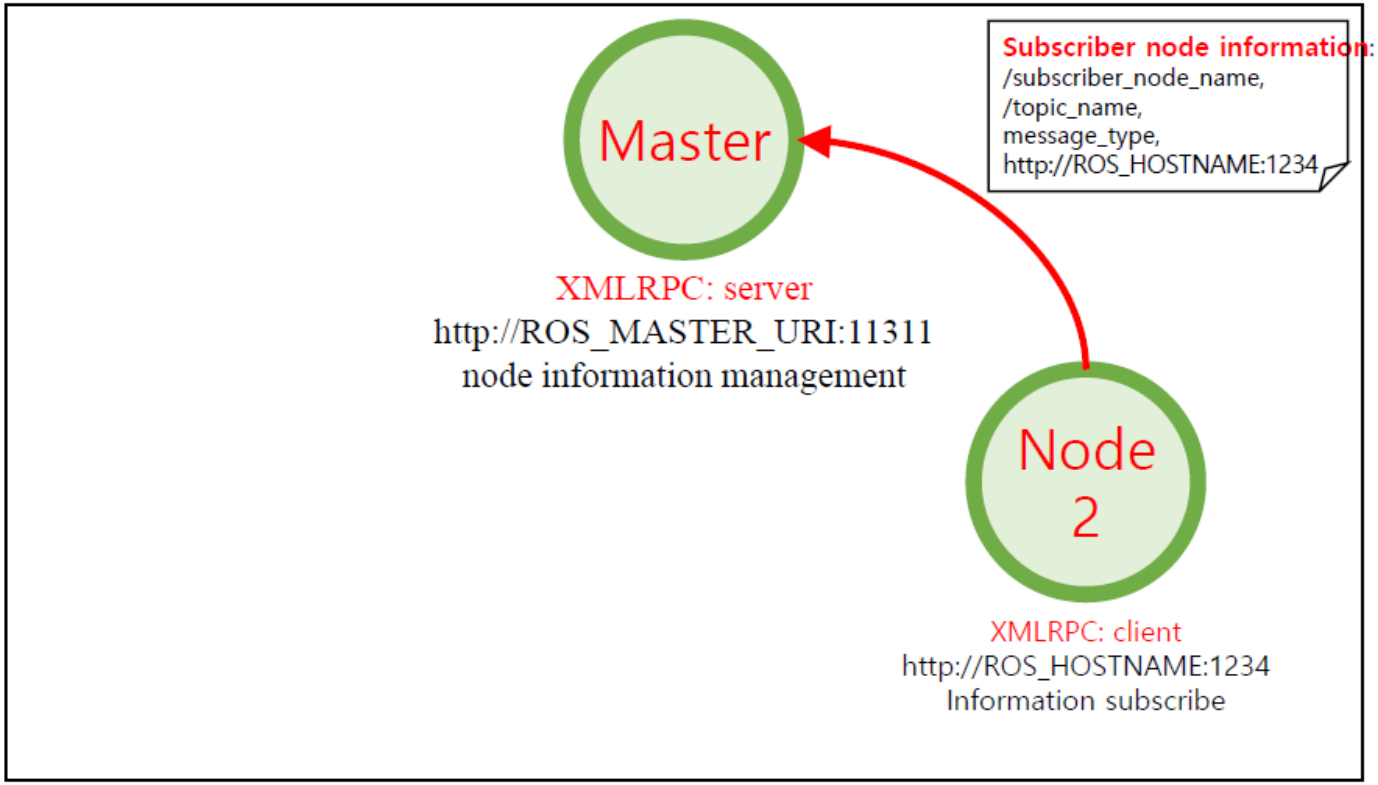
- $ roscore



Message Communication

2. Run Subscriber node

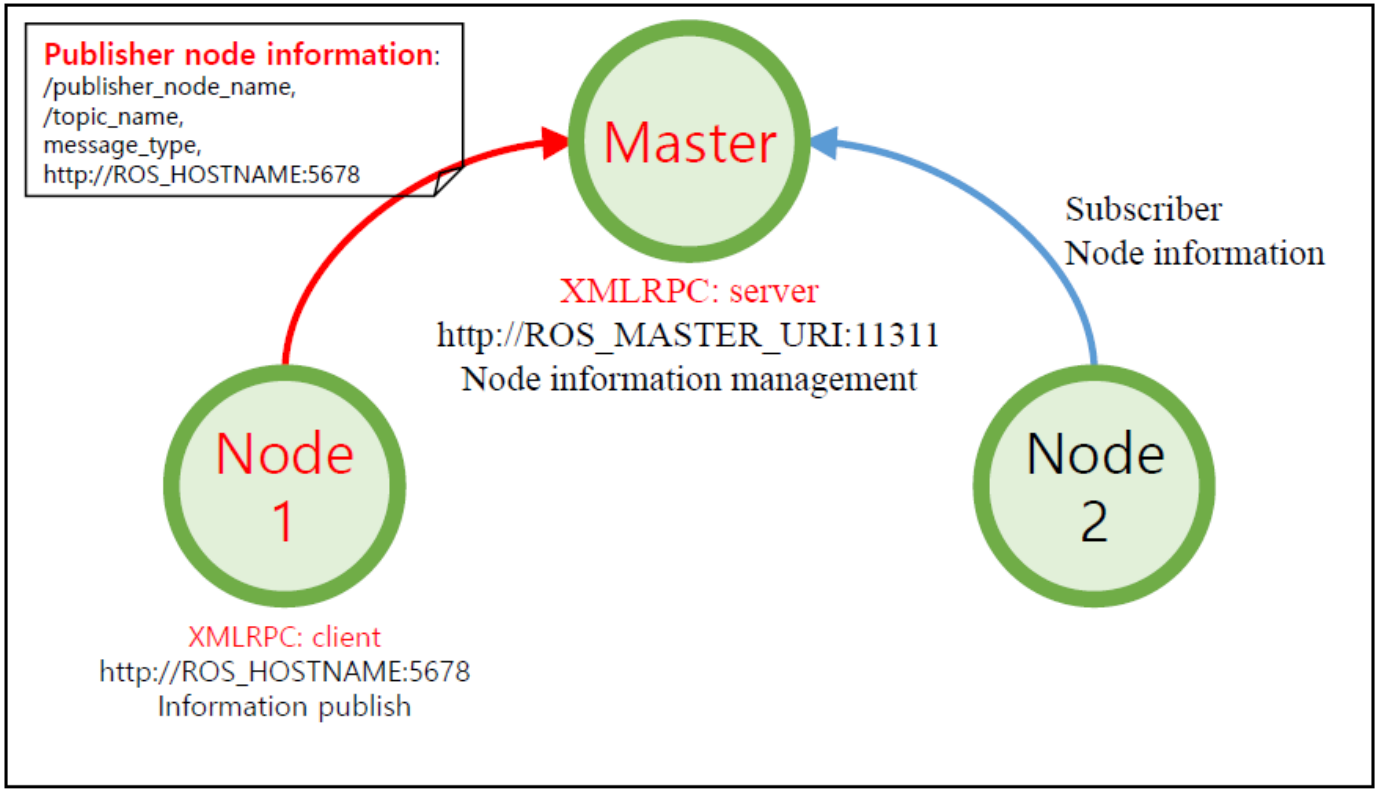
- $rosrun packagename nodename



Message Communication

3. Run Publisher node

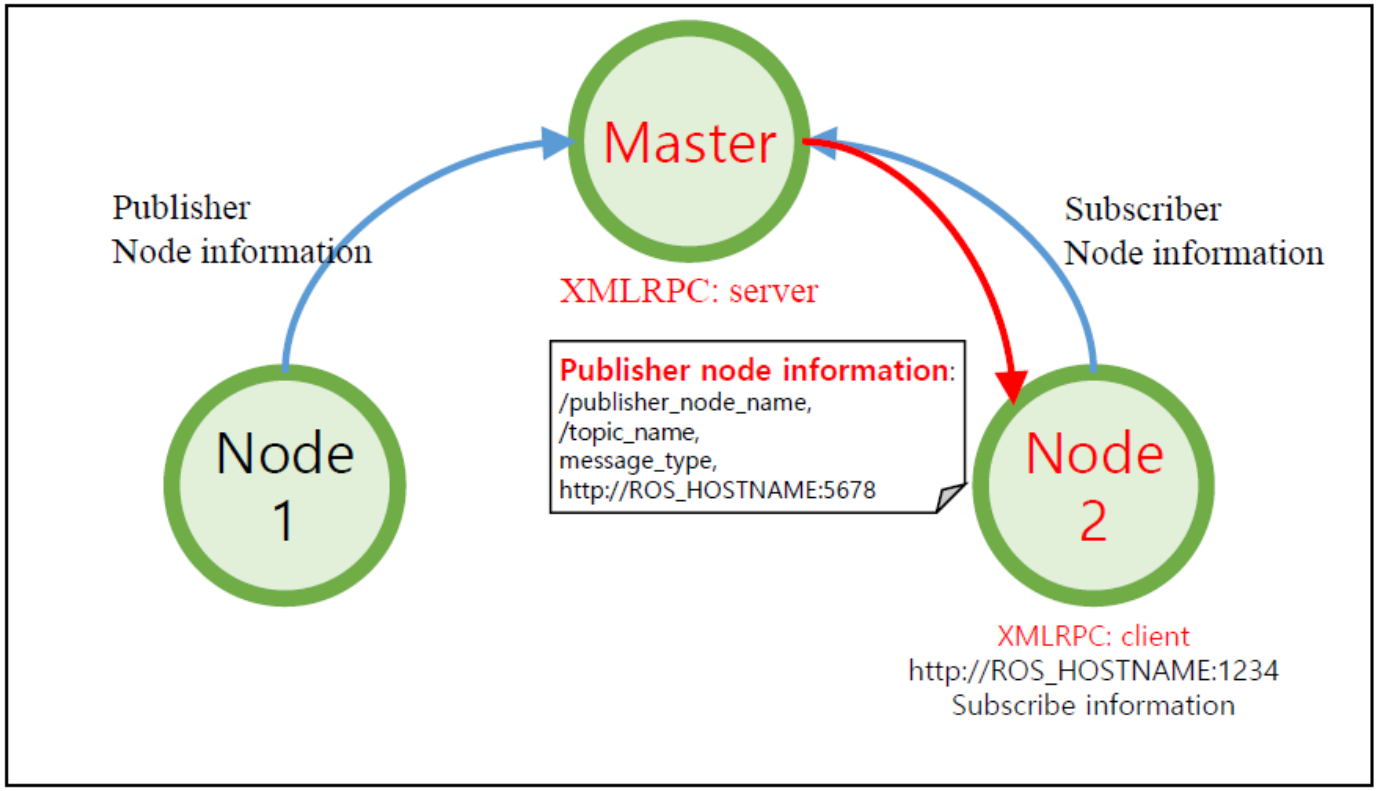
- $rosrun packagename nodename



Message Communication

4. Publisher Information

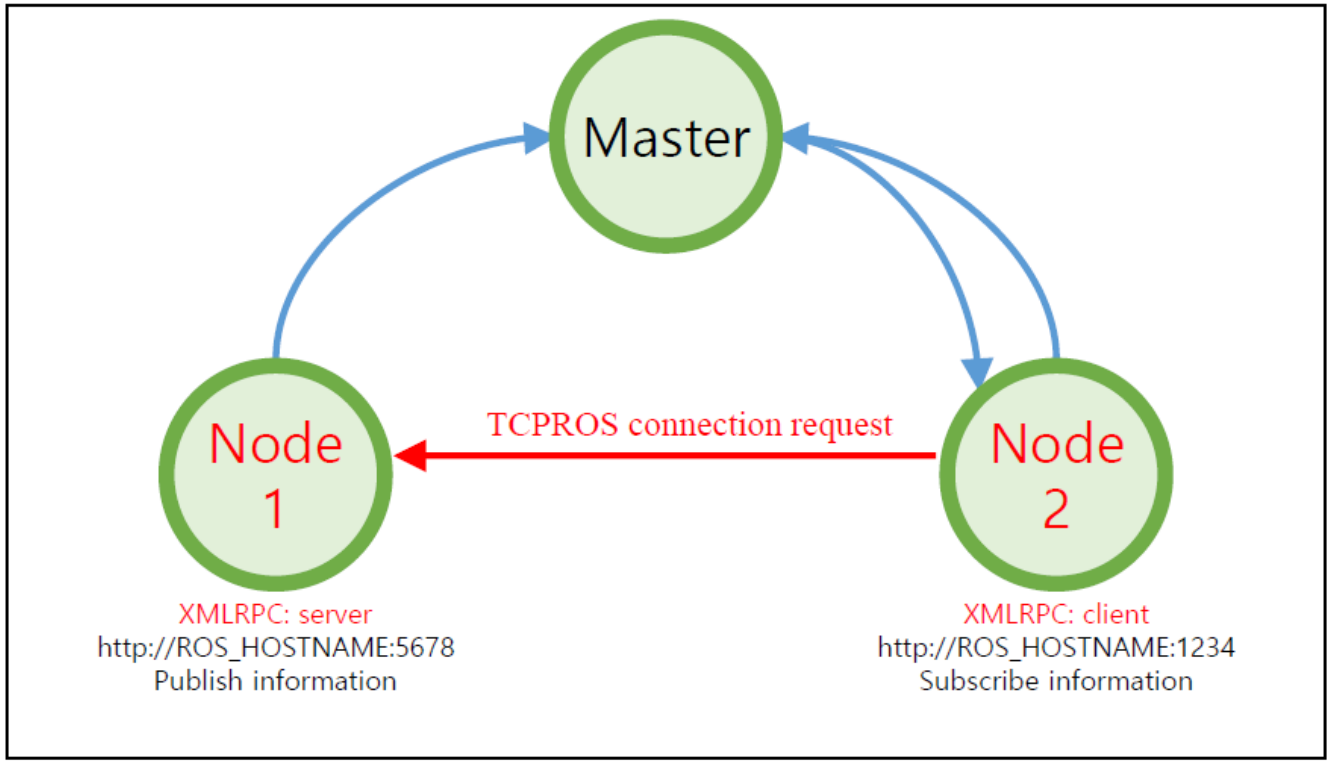
- The master informs the subscriber node of the new publisher information.



Message Communication

5. Request access to the publisher node

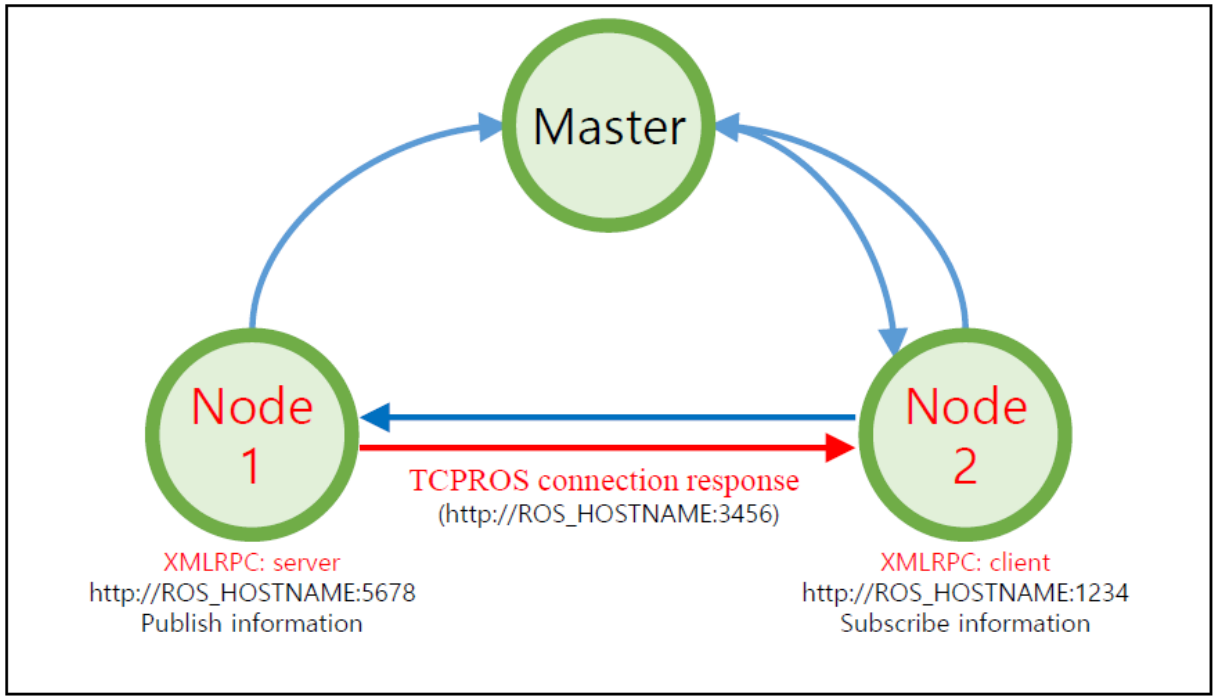
- Request TCPROS connection using the publisher information from the master



Message Communication

6. Connection response to subscriber node

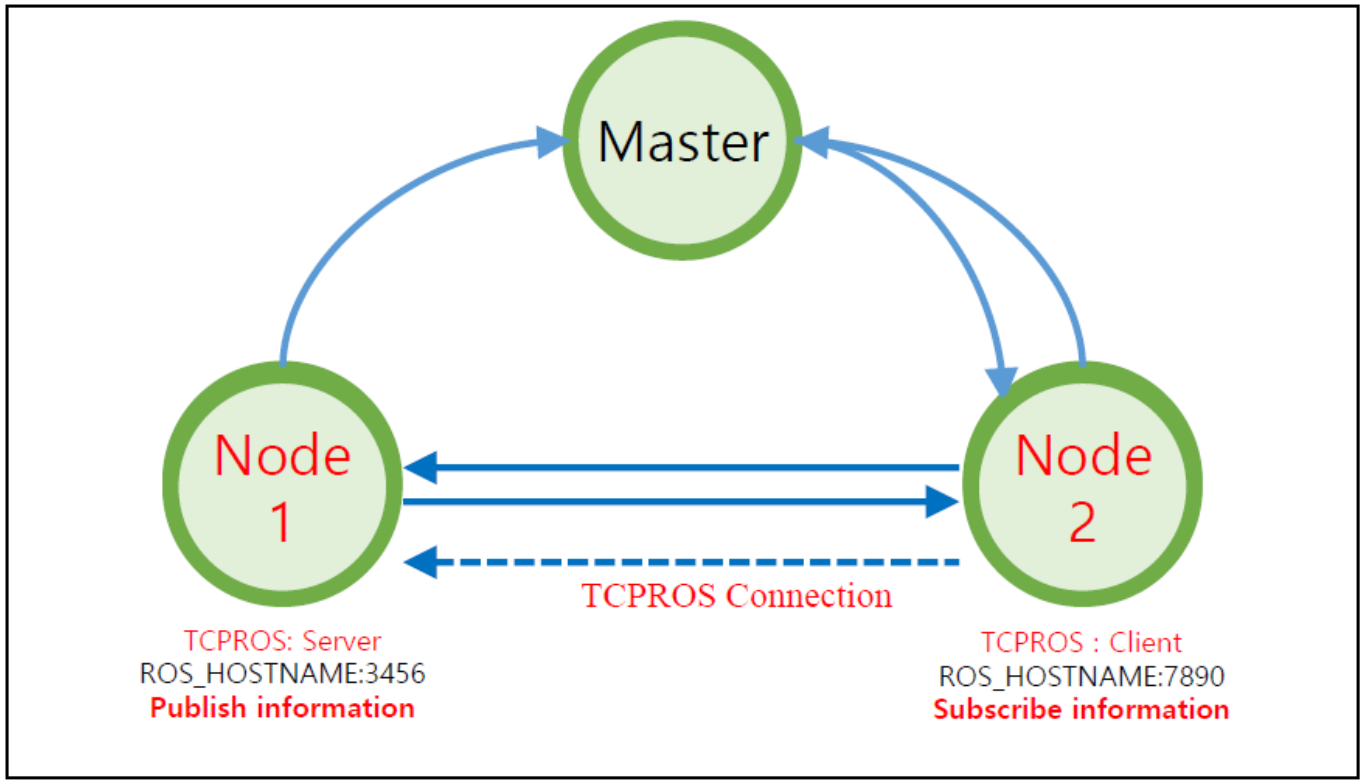
- Return TCP URI address and port number corresponding to the connection response



Message Communication

7. TCP Connection

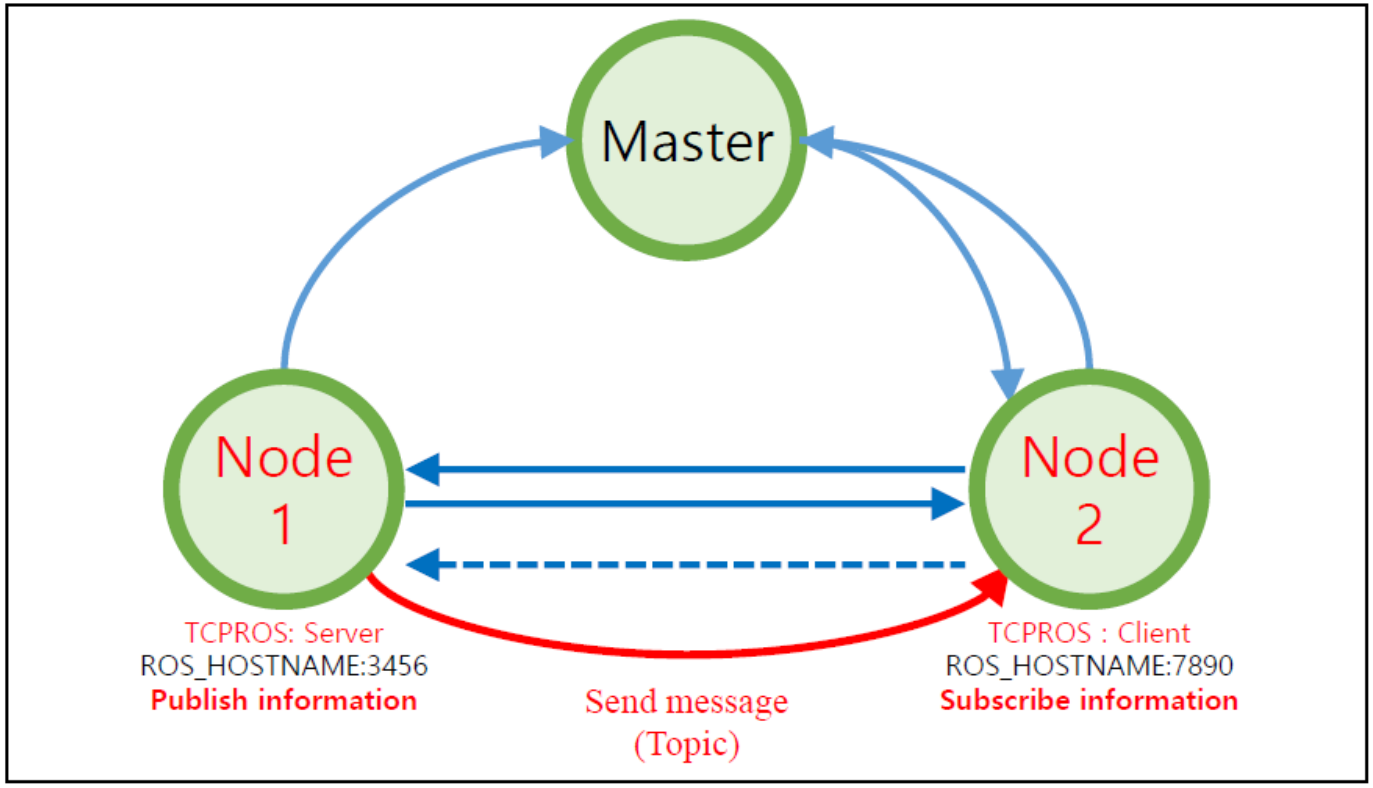
- Establish connection with the publisher node using TCPROS.



Message Communication

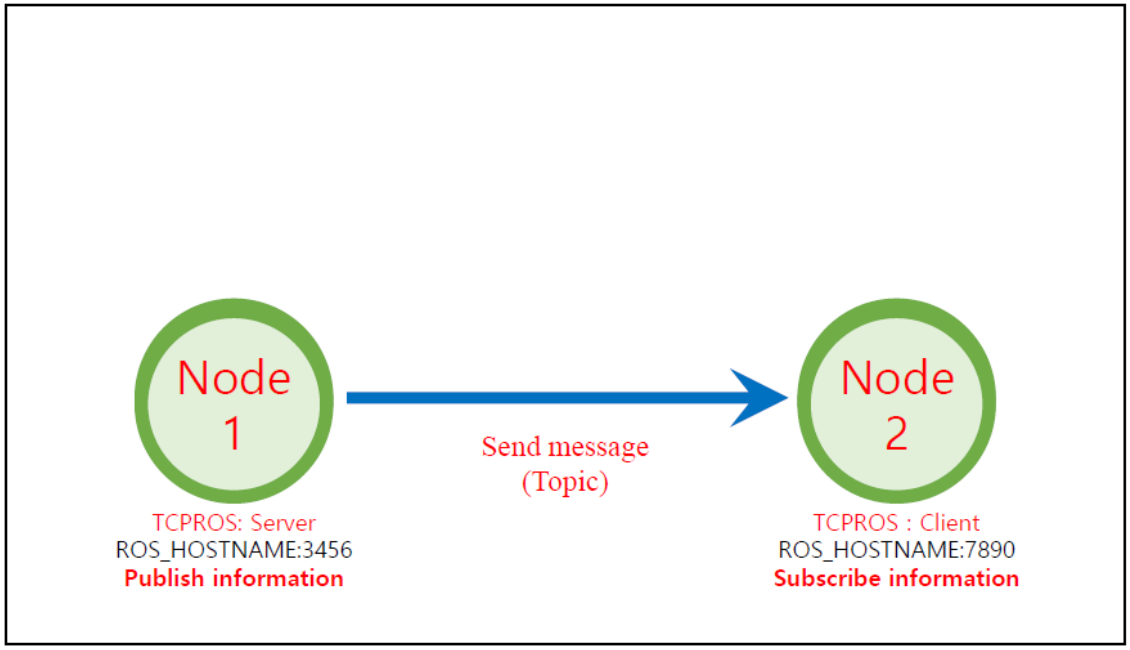
8. Send message

- The publisher node sends a message to the subscriber node (topic)



Message Communication

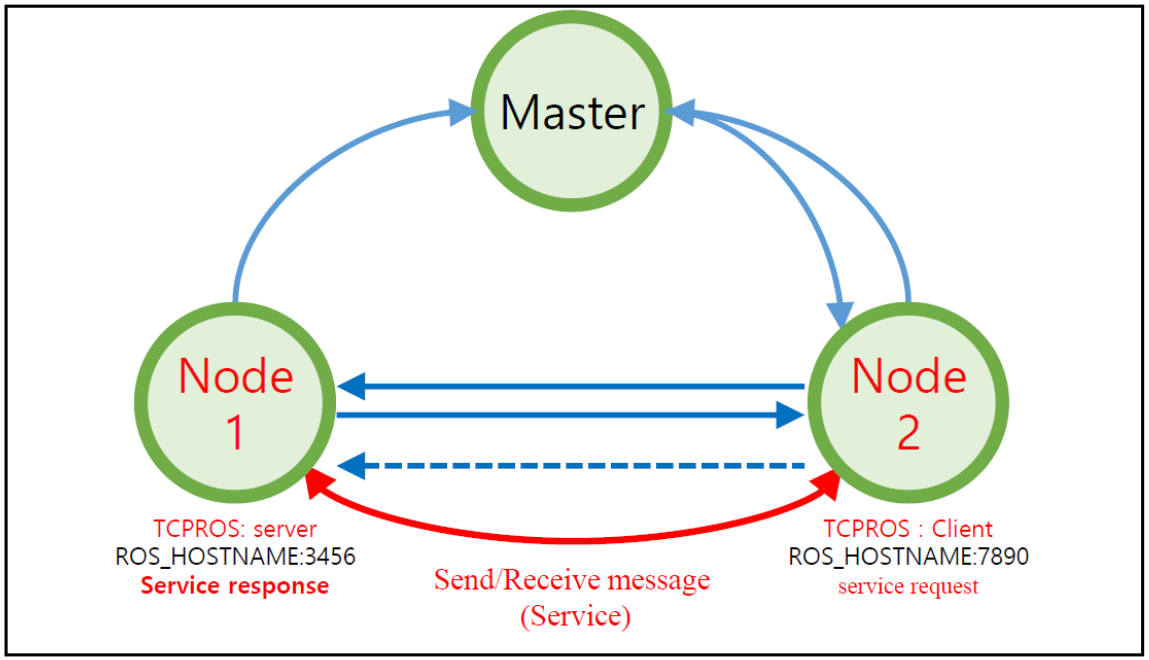
- In Topic mode, messages are continuously transmitted unless the connection is terminated. That is, continuity.



Message Communication

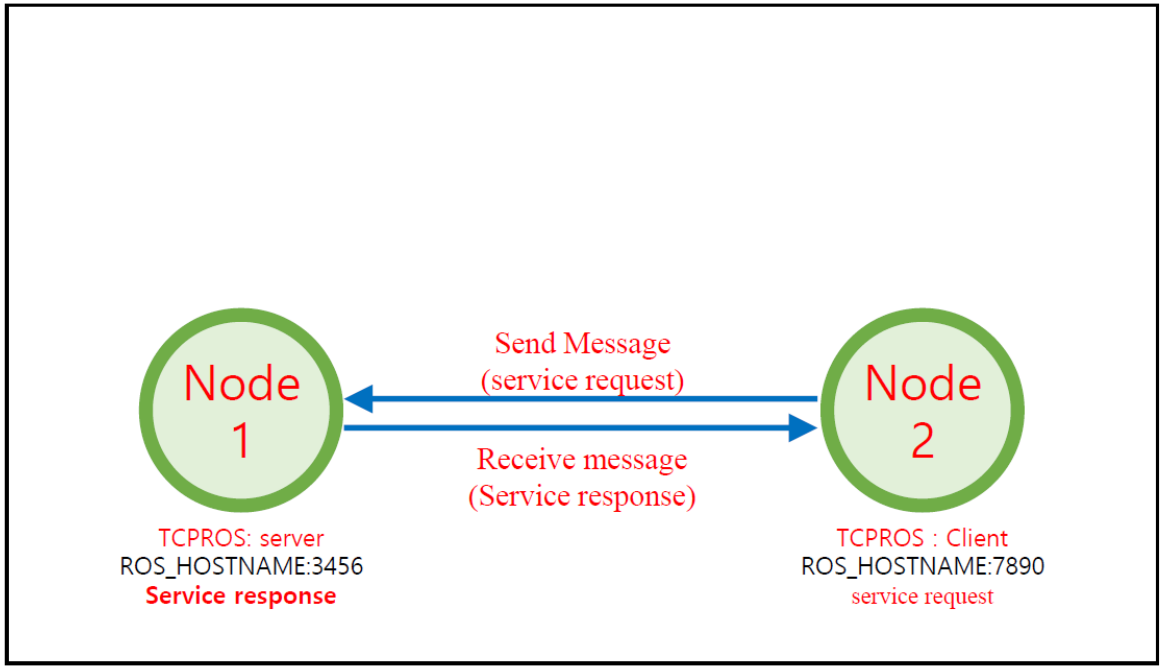
9. Service Request and Response

- For only once, service request and service response are performed and disconnected from each other.



Message Communication

- Unlike the topic, the service connects only once and disconnected after a service request and a service response are performed. That is, it is one-time.



Message Communication Concept

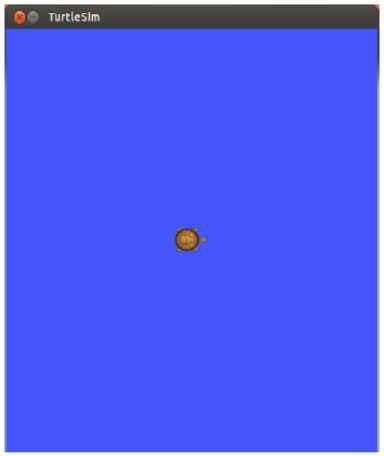
- **turtlesim package**

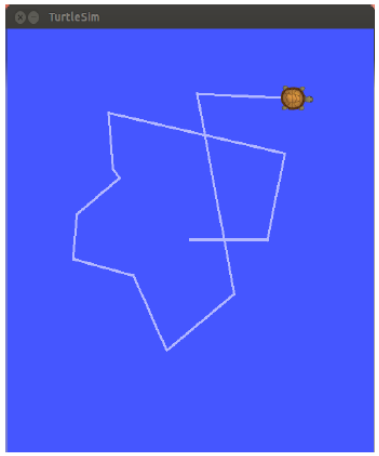
  - roscore

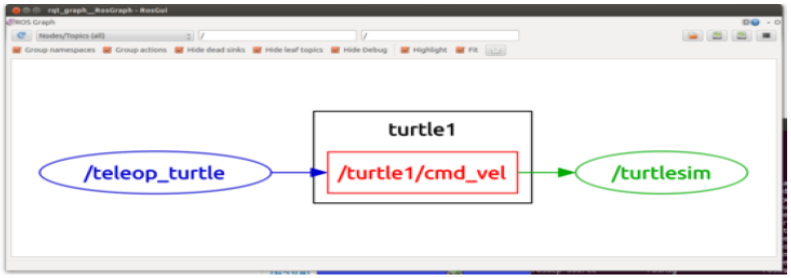
  - rosrun turtlesim turtlesim_node

  - rosrun turtlesim turtle_teleop_key

  - rosrun rqt_graph rqt_graph

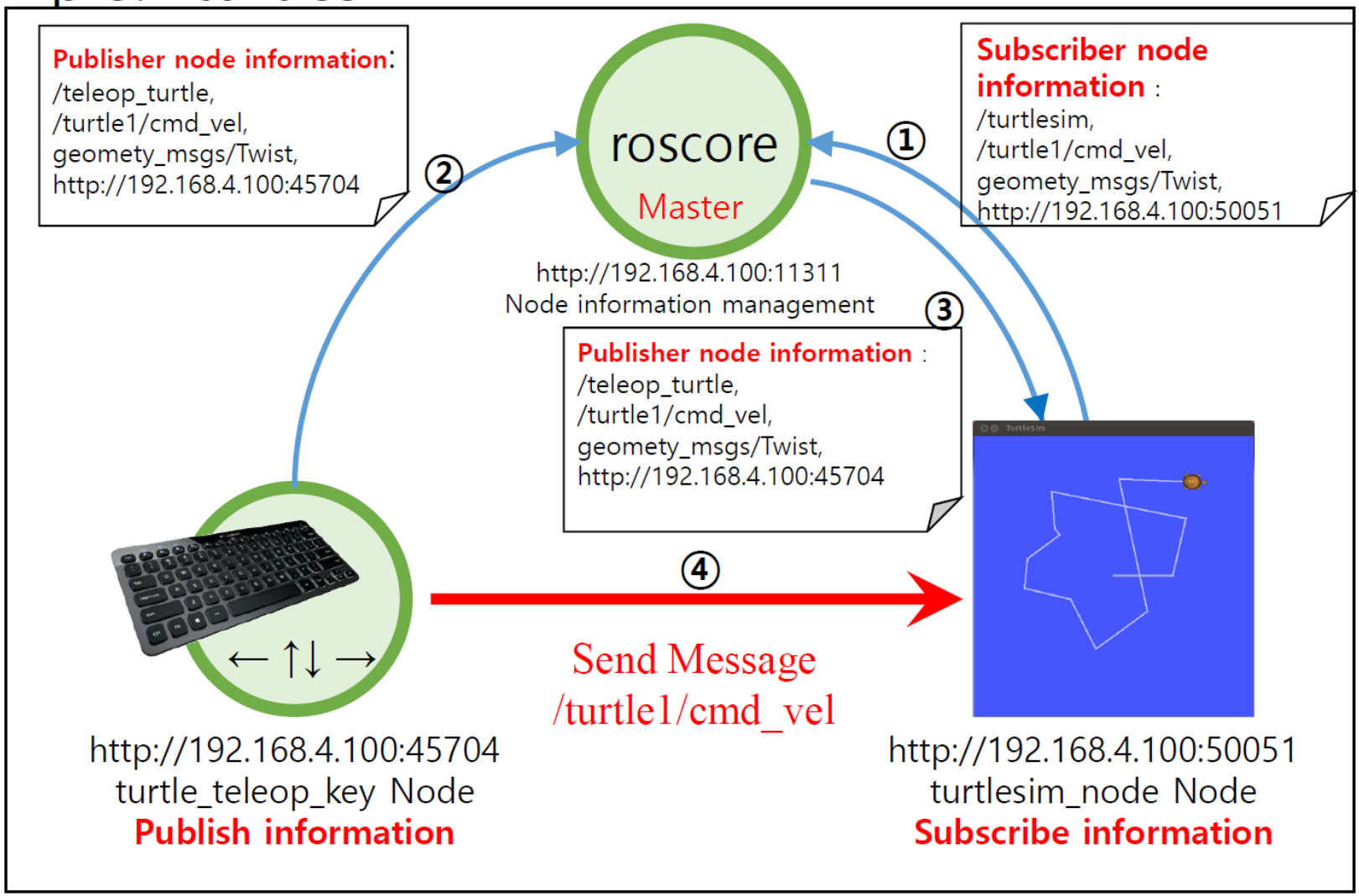




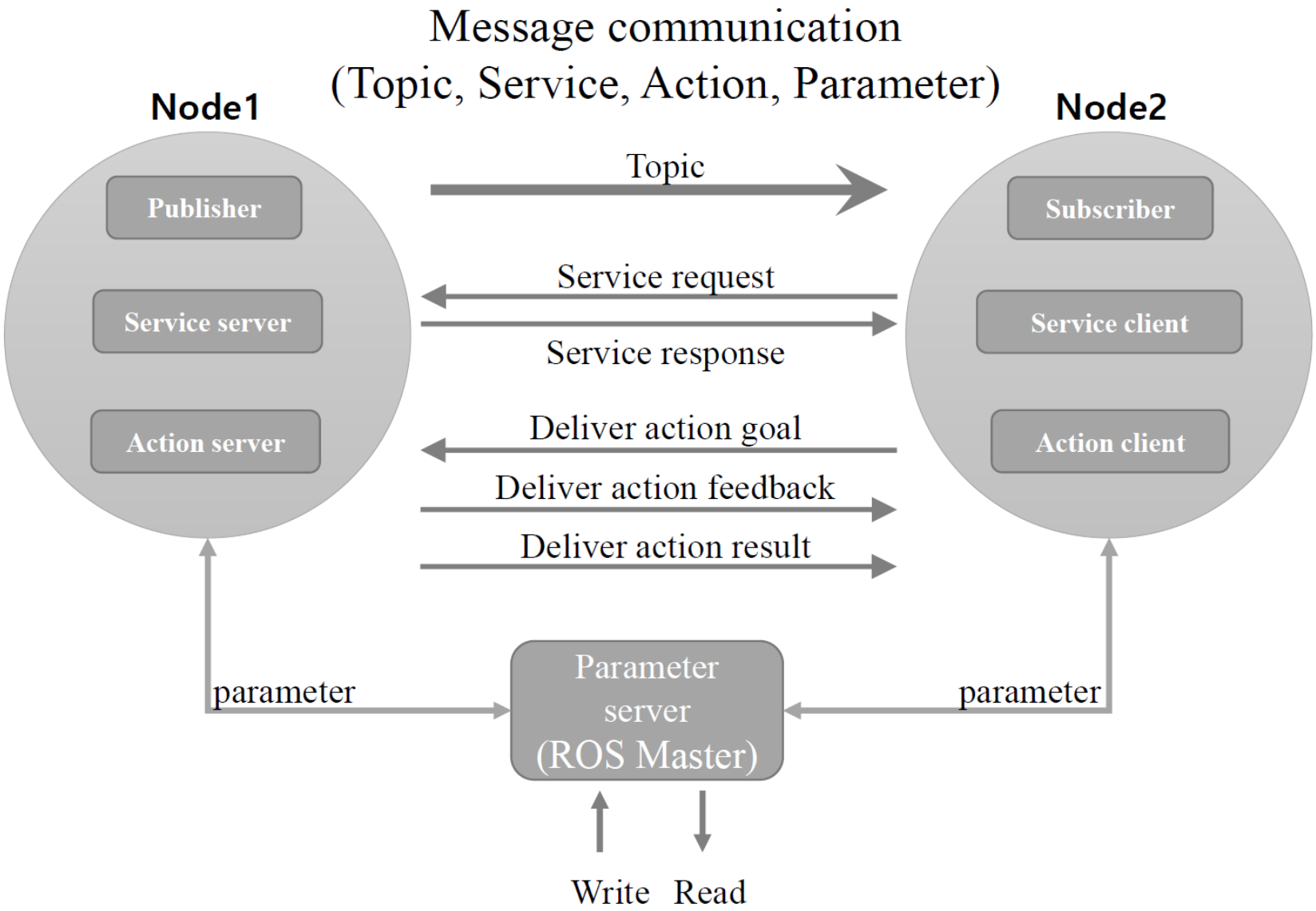


Message Communication Concept

- 10. Example! turtlesim



ROS Message


Message communication
(Topic, Service, Action, Parameter)
Node1
Node2
Publisher
Service server
Action server
Subscriber
Service client
Action client
Topic
Service request
Service response
Deliver action goal
Deliver action feedback
Deliver action result
parameter
Parameter server (ROS Master)
parameter
Write
Read

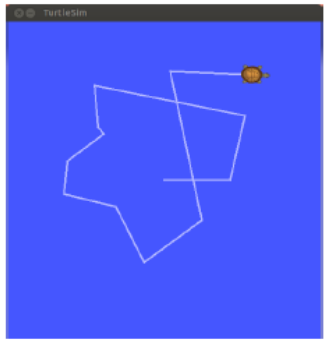
ROS Message

- Message is a type of data travel around nodes
  - Topics, services, and actions all use messages
    - http://wiki.ros.org/msg
    - http://wiki.ros.org/common_msgs
  - **Simple type**
    - ex) integer, floating point, boolean
    - http://wiki.ros.org/std_msgs
  - **A simple data structure containing messages in a message**
    - ex) geometry_msgs/PoseStamped
    - http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html
  - **An array data structure in which messages are listed**
    - ex) float32[ ] ranges
    - ex) sensor_msgs/LaserScan
    - http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html

ROS Message
(ex: geometry_msgs/Twist)

http://192.168.4.100:45704
turtle_teleop_key Node
Publish Information

Send message
/turtle1/cmd_vel
(geometry_msgs/Twist)



http://192.168.4.100:50051
turtlesim_node Node
Subscribe information

[geometry_msgs/Twist]

Vector3 linear
Vector3 angular

[geometry_msgs/Vector3]
float64 x
float64 y
float64 z

[geometry_msgs/Vector3]
float64 x
float64 y
float64 z

http://docs.ros.org/api/geometry_msgs/html/msg/Twist.html

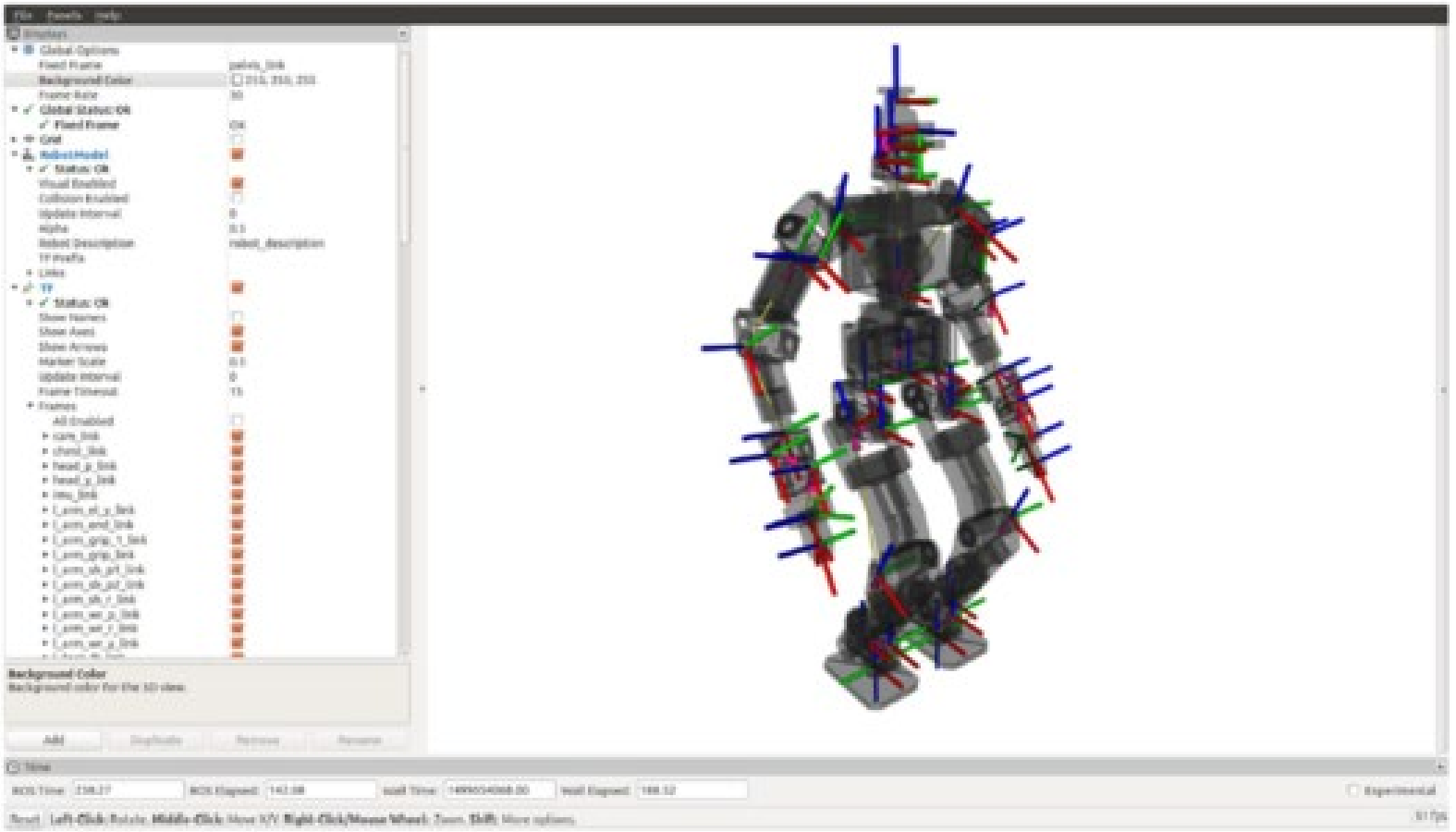
Names

- Name
  - **A unique identifier** for a Node or a message (topic, service, action, parameter)
  - ROS supports abstract data types called **graphs**
  - **Global**
    - Use the name as is or prepend a slash (/) to the name.
  - **Private**
    - Prepend a tilde (~) to the name
  - An example is covered in Chapter 7, ROS Basic Programming, roslaunch.

Node	Relative (default)	Global	Private
/node1	bar -> /bar	/bar -> /bar	~bar -> /node1/bar
/wg/node2	bar -> /wg/bar	/bar -> /bar	~bar -> /wg/node2/bar
/wg/node3	foo/bar -> /wg/foo/bar	/foo/bar -> /foo/bar	~foo/bar -> /wg/node3/foo/bar

Coordinate transformation(TF, transform)

- Relative coordinate transformation of each joint
  - Indicates the relationship between joints in the form of tree structure



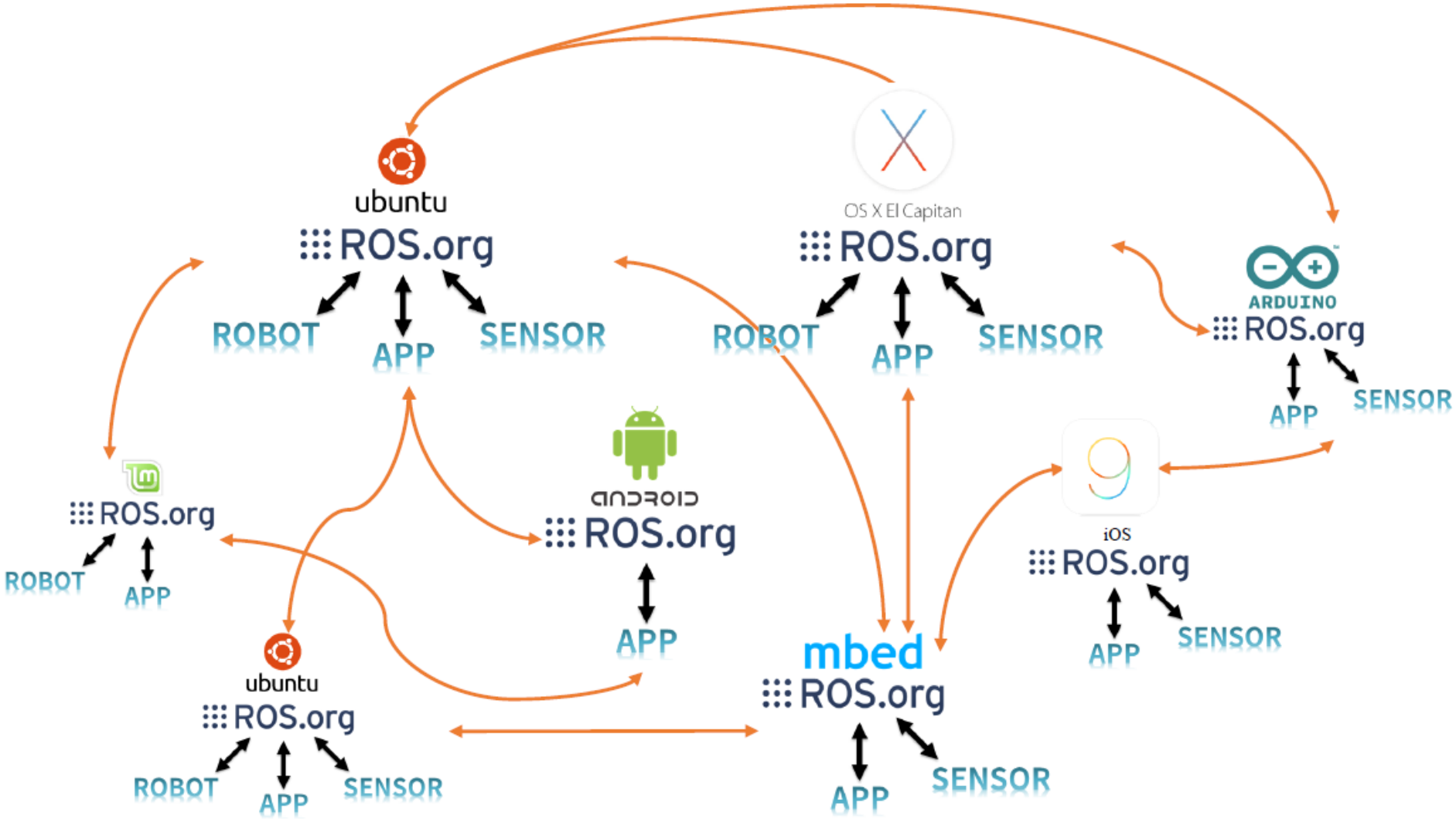
Client Library

- Supports various programming languages
  - roscpp, rospy,roslisp
  - rosjava, roscs, roseus, rosgo, roshask, rosnodejs, RobotOS.jl, roslua, PhaROS, rosR, rosruby, Unreal-Ros-Plugin
  - MATLAB for ROS
  - LabVIEW for ROS

http://wiki.ros.org/Client%20Libraries

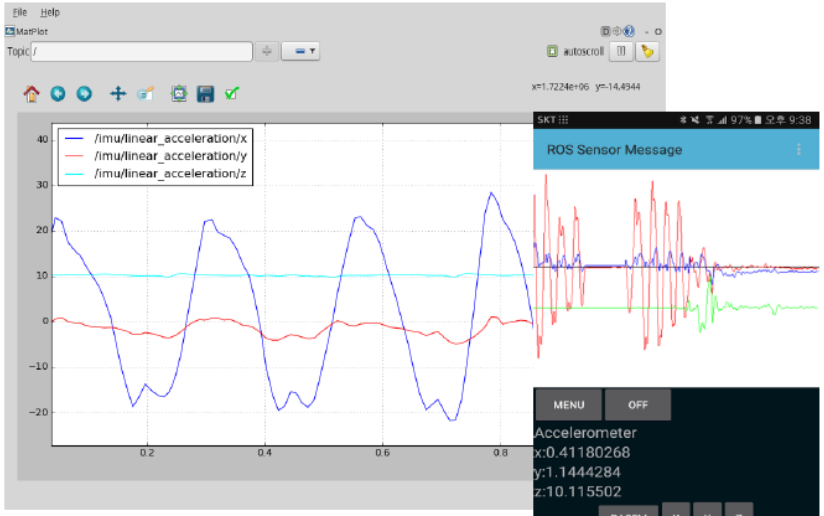
https://commons.wikimedia.org/wiki/File:Prog-languages.png

Communication between Heterogeneous Devices



Communication between Heterogeneous Devices

- Example 1: Transferring images remotely (see Chapter 8, Camera)
- Example 2: Checking the acceleration value of your Android smartphone on your PC (APP)
- Example 3: Controlling TurtleBot with Android Smartphone (APP)



[PC] [Smartphone]

[PC] [Smartphone]

Command-Line Tools

The following command-Line Tools are used throughout the ROS course.

rospack, roscd, rospd, rosls, rosed, roscp, rosdep, roswf, catkin_create_pkg, wstool, catkin_make, roscore, rosrun, roslaunch, rosnode, rostopic, rosservice, rosparam, rosmsg, rossrv, rosbag, tf_echo

Please refer to the distributed ROS cheat sheet.

https://github.com/oroca/oroca_ros_tutorials/raw/master/ROScheatsheet_indigo_catkin.pdf

ROS Shell Commands

command	importance	Command description	Detailed description
roscd	★★★	ros+cd(changes directory)	Move to the directory of the specified ROS package
rosls	★☆☆	ros+ls(lists files)	Check the list of files in the ROS package
rosed	★☆☆	ros+ed(editor)	Edit files in the ROS package
roscp	★☆☆	ros+cp(copies files)	Copy Files in the ROS Package
rospd	☆☆☆	ros+pushd	Add Directories to the ROS Directory Index
rosd	☆☆☆	ros+directory	Check the ROS directory index

ROS Execution Commands

command	importance	Command description	Detailed description
roscore	★★★	ros+core	- master (ROS name service) - rosout (record logs) - parameter server (Manage parameters)
rosrun	★★★	ros+run	Run the Node
roslaunch	★★★	ros+launch	Run multiple nodes with options
rosclean	★★☆	ros+clean	Check or delete ROS log files

ROS Information Commands

command	importance	Command description	Detailed description
rostopic	★★★	ros+topic	Check ROS topic information
rosservice	★★★	ros+service	Check ROS service information
rosnode	★★★	ros+node	Check ROS node information
rosparam	★★★	ros+param(parameter)	Check and modify ROS parameter information
rosbag	★★★	ros+bag	ROS message recording, playback
rosmsg	★★☆	ros+msg	Check ROS message information
rossrv	★★☆	ros+srv	Check ROS service information
rosversion	★☆☆	ros+version	Check ROS Package, Release Version Information
roswtf	☆☆☆	ros+wtf	ROS System Inspection

ROS Catkin Commands

command	importance	Detailed description
catkin_create_pkg	★★★	Automatically generates a package
catkin_make	★★★	Builds based on the Catkin build system
catkin_eclipse	★★☆	Converts the package created by the Catkin build system for Eclipse
catkin_prepare_release	★★☆	Log cleanup and version tagging for release
catkin_generate_changelog	★★☆	Creates or Updates the CHANGELOG.rst file for release
catkin_init_workspace	★★☆	Initializes the workspace of Catkin build system
catkin_find	★☆☆	Search catkin

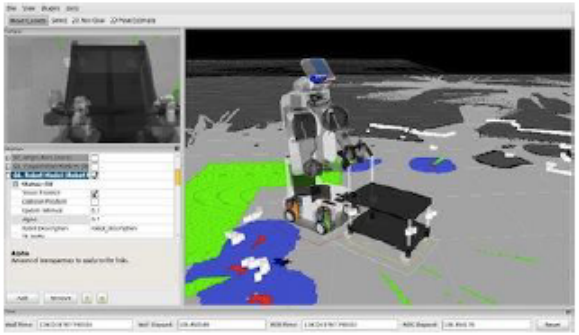
ROS Package Commands

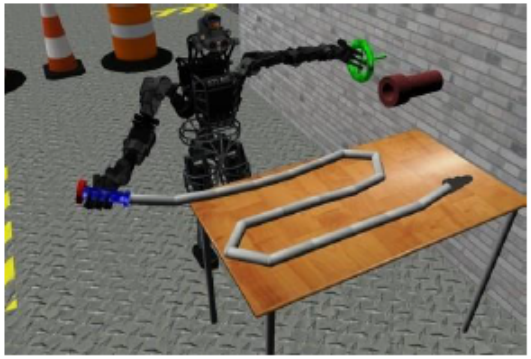
command	importance	Command description	Detailed description
rospack	★★★	ros+pack(age)	Displays information related to the ROS package
rosinstall	★★☆	ros+install	Installs additional package for ROS
rosdep	★★☆	ros+dep(endencies)	Installs dependency files for the package
roslocate	☆☆☆	ros+locate	Acquires information regarding ROS package
roscreate-pkg	☆☆☆	ros+create-pkg	Automatically generates ROS package (used in old rosbuild system)
rosmake	☆☆☆	ros+make	Builds the ROS package (formerly used by the rosbuild system)

Various Development Tools for ROS

- Provides various development tools needed for robot development
- Improving the efficiency of robot development

- Command-Line Tools
  - Robot access only with commands provided by ROS without GUI & Use almost all ROS features
- RViz
  - Provides powerful 3D visualization tool
  - Visualizes sensor data such as laser, camera, etc.
  - Represents robot configuration and planned motion
- RQT
  - Provides Qt-based framework for developing graphic interface
  - Displays nodes and connection information between them (rqt_graph)
  - Floats encoder, voltage, or number that changes over time (rqt_plot)
  - Records data in message form and play back (rqt_bag)
- Gazebo
  - 3D simulator which supports physics engine, robot, sensor, environmental model, etc.
  - High compatibility with ROS



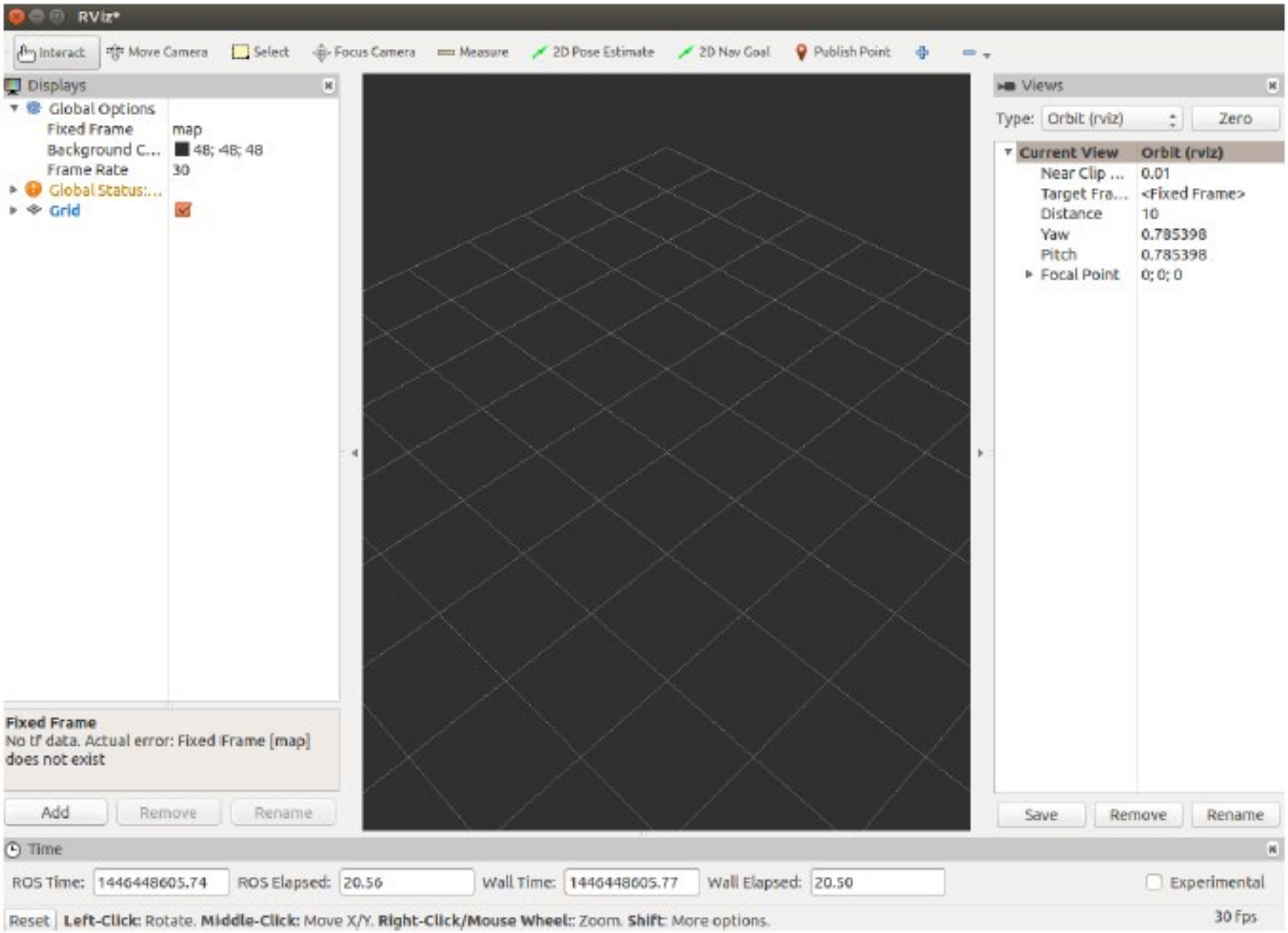


http://www.ros.org/core-components/

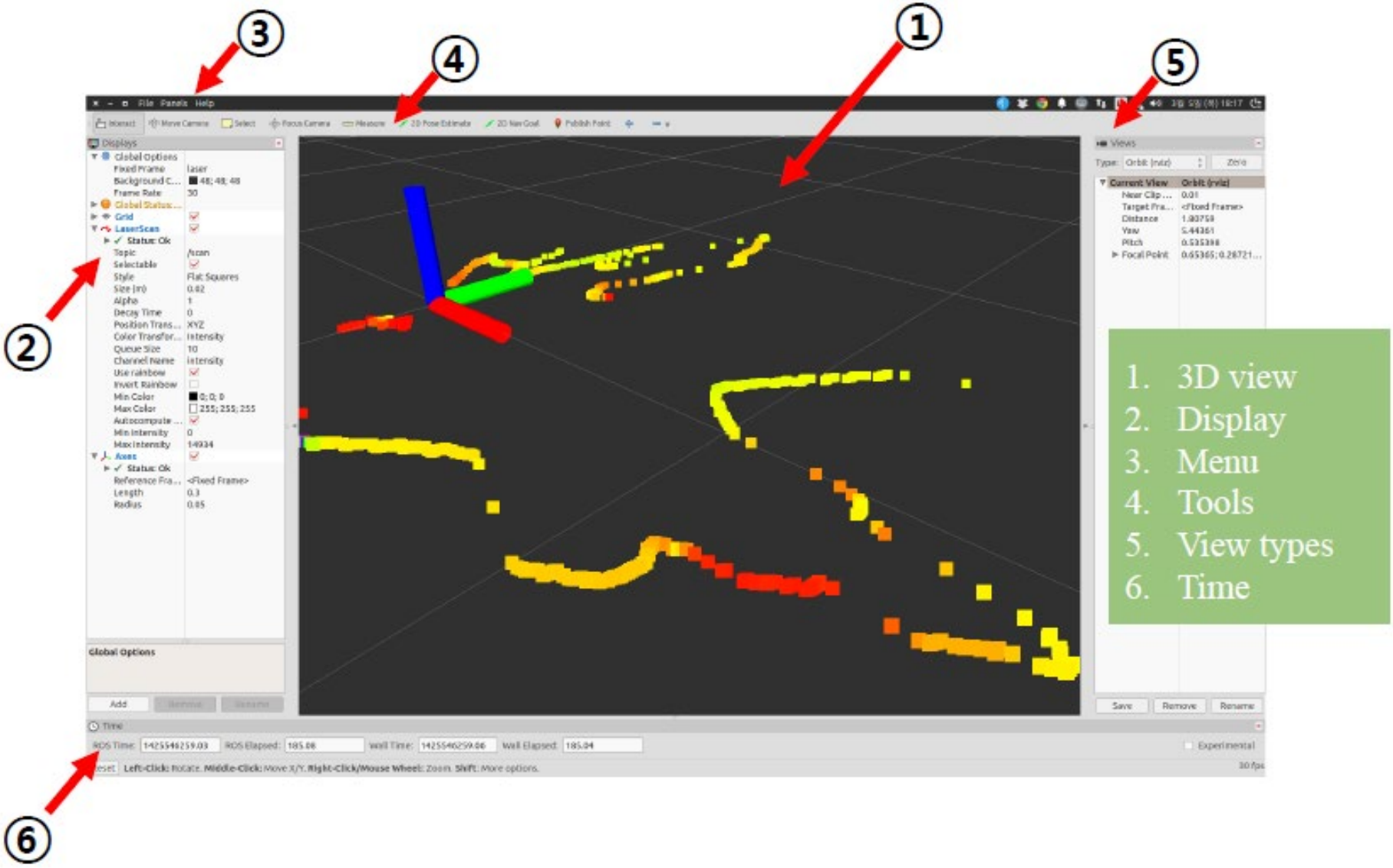
RViz(ROS Visualization Tool)

- 3D visualization tool for ROS
  - Visualization of sensor data
  - Distance data of LASER distance sensor(LDS)
  - Point cloud data from depth camera such as 'RealSense', 'Kinect', 'Xtion', etc.
  - Image data of camera
  - Inertia data of IMU sensor ...
- Represents robot configuration and planned motion
  - URDF (Unified Robot Description Format)
- Navigation
- Manipulation
- Tele-operation

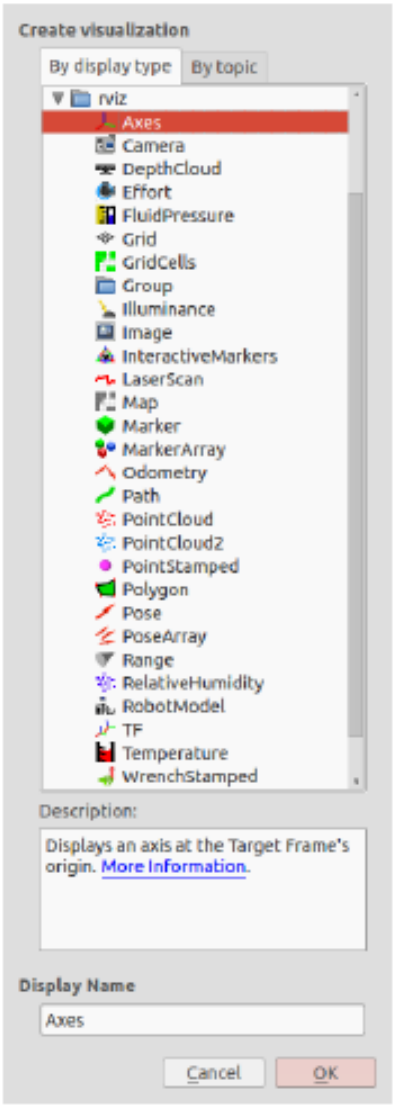
RViz initial screen (not configured yet)



Screen Configuration of Rviz (for LDS)



Display type of RViz (Click ‘ADD’in Display Menu)



- Axes
- Camera
- Depth cloud
- Effort
- Fluid pressure
- Grid
- Grid cells (used for map)
- Group
- Illuminance
- Video
- Interactive marker
- Laser scan
- Map
- Marker
- Marker array
- Odometry
- Path
- Point cloud
- Point cloud2
- Point stamped
- Polygon
- Pose
- Pose array
- Range
- Temperature
- Robot model
- TF
- Relative Humidity
- WrenchStamped

RQT: Plug-in Type Comprehensive GUI Tool for ROS

- Starting with the ROS Fuerte version, the existing 'rxbag', 'rxplot', 'rxgraph', etc. have been merged with 'rqt'. It is now available as **comprehensive GUI tool** for ROS with plug-ins such as 'rqt_bag', 'rqt_plot', 'rqt_graph', etc.
- Since 'rqt' is developed with 'Qt', users can freely add and develop **plugins**
- Let's take a look at **'rqt_image_view'**, **'rqt_graph'**, **'rqt_plot'**, **'rqt_bag'** which are representative plugins of 'rqt'

- In addition, there are plugins such as
- rqt_action, rqt_gui, rqt_plot, rqt_runtime_monitorrqt_bag, rqt_gui_cpp, rqt_pose_view, rqt_rvizrqt_bag_plugins, rqt_gui_py, rqt_publisher, rqt_service_callerrqt_capabilities, rqt_image_view, rqt_py_common, rqt_shellrqt_console, rqt_launch, rqt_py_console, rqt_srvrqt_controller_manager, rqt_logger_level, rqt_reconfigure, rqt_tf_treerqt_dep, rqt_moveit, rqt_robot_dashboard, rqt_toprqt_ez_publisher, rqt_msg, rqt_robot_monitor, rqt_topicrqt_graph, rqt_nav_view, rqt_robot_steering, rqt_web, etc. (wow.. ——;;)

RQT Plug-in #1

1. Action

- Action Type Browser | Check the data structure of action type

2. Configuration

- Dynamic Reconfigure | Change the GUI setting value to change the setting value provide by the nodes
- Launch | GUI version of 'roslaunch'

3. Introspection

- Node Graph | Graph view showing relationship diagrams and message flow of running nodes
- Package Graph | Graph view showing node dependencies
- Process Monitor | Check CPU utilization, memory usage, and number of threads of running nodes

4. Logging

- Bag | ROS data logging
- Console | Check for messages such as warning, error that occur on the nodes
- Logger Level | Select and display logger information such as Debug, Info, Warn, Error, Fatal

RQT Plug-in #2

5. Miscellaneous Tools

- Python Console | Python console screen
- Shell | Activate shell
- Web | Activate web browser

6. Robot

- Depending on the robot, add a plug-in such as a dashboard

7. Robot Tools

- Controller Manager | Plug-in required to control the controller
- Diagnostic Viewer | Check robot device and error
- Moveit! Monitor | Check 'Moveit!' data used in robot arm planning
- Robot Steering | Robot adjustment GUI tool, used in remote control to steer the robot
- Runtime Monitor | Check for errors and warning on nodes in real time

RQT Plug-in #3

8. Services

- Service Caller | Connect to the running service server and request service
- Service Type Browser | Check the data structure of the service type

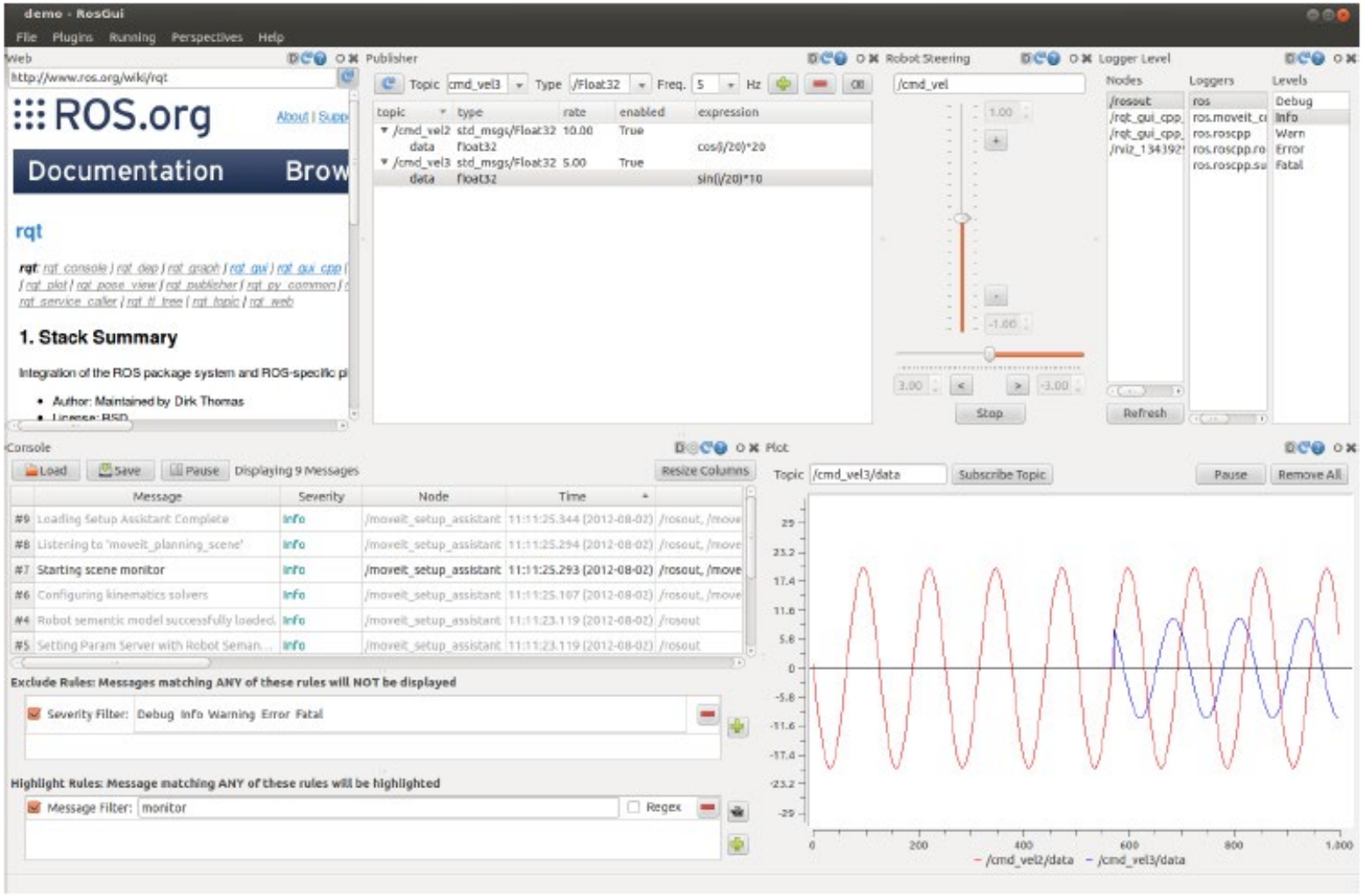
9. Topics

- Easy Message Publisher | Publish topic in GUI environment
- Topic Publisher | Create and publish topic
- Topic Type Browser | Check the data structure of the topic type
- Topic Monitor | Check the information of selected topic

10. Visualization

- Image View | Check image data of camera
- Navigation Viewer | Check location and target point of robot navigation
- Plot | 2D data plot GUI plug-in, 2D data plotting
- Pose View | Show current TF location and model location
- RViz | Rviz plug-in which is 3D visualization tool
- TF Tree | Graph view showing tf relation as a tree structure

RQT Example

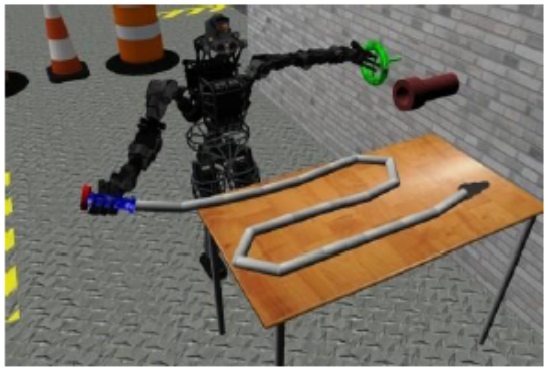


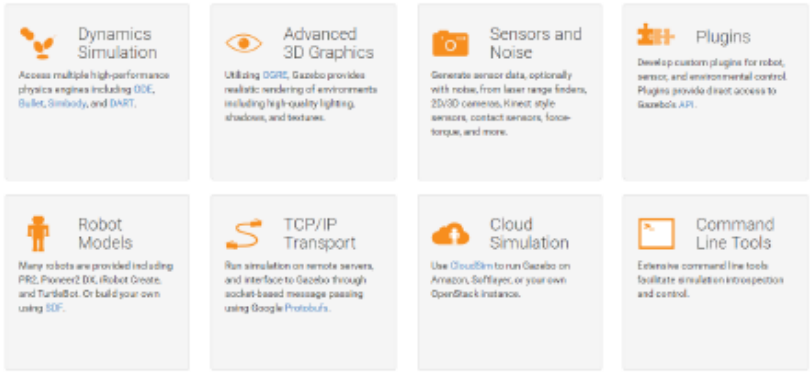
http://www.ros.org/core-components/

Gazebo

- Gazebo is 3D Simulator with Physics Engine, Robot model, Sensor, Environment model, and so on. It helps you to get data similar to the one in real environment.
- Gazebo is regarded as the best simulator among recently-introduced Open Simulator. In addition, it is selected as an **official simulator for DARPA Robotics Challenge**
- It is highly compatible with ROS.

GAZEBO





http://gazebosim.org/

Gazebo

Reference

- R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to Autonomous Mobile Robots. MIT Press, 2nd Edition, 2011, ISBN-10: 0262015358.
- Y. Pyo, H. Cho, R. Jung, and T. Lim, ROS Robot Programming, ROBOTIS Co., Ltd., 2017, ISBN 979-11-962307-1-5
- J. O'Kane, A Gentle Introduction to ROS, CreateSpace Independent Publishing Platform, 2013, ISBN-13: 978-1492143239