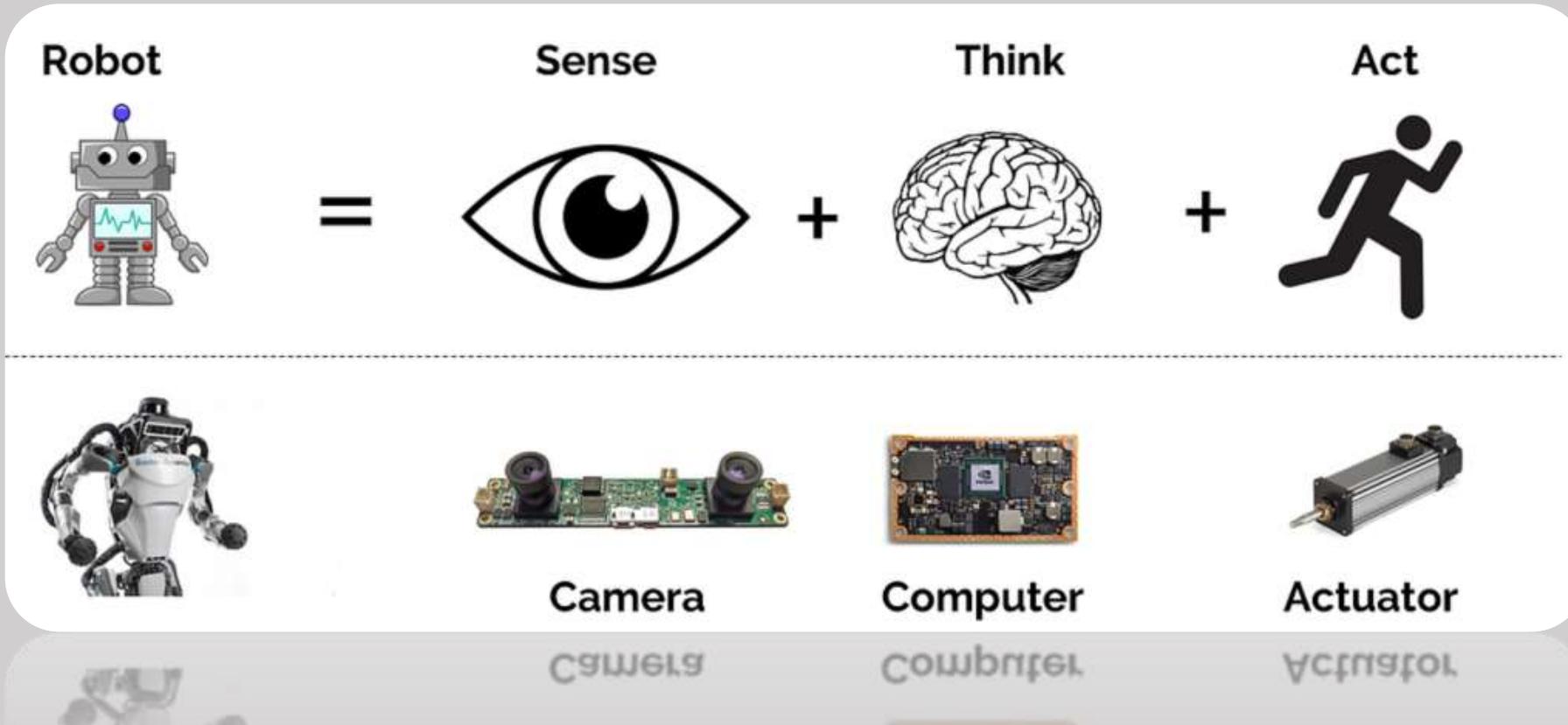


Robot Operating System

Getting started with Robot Operating System

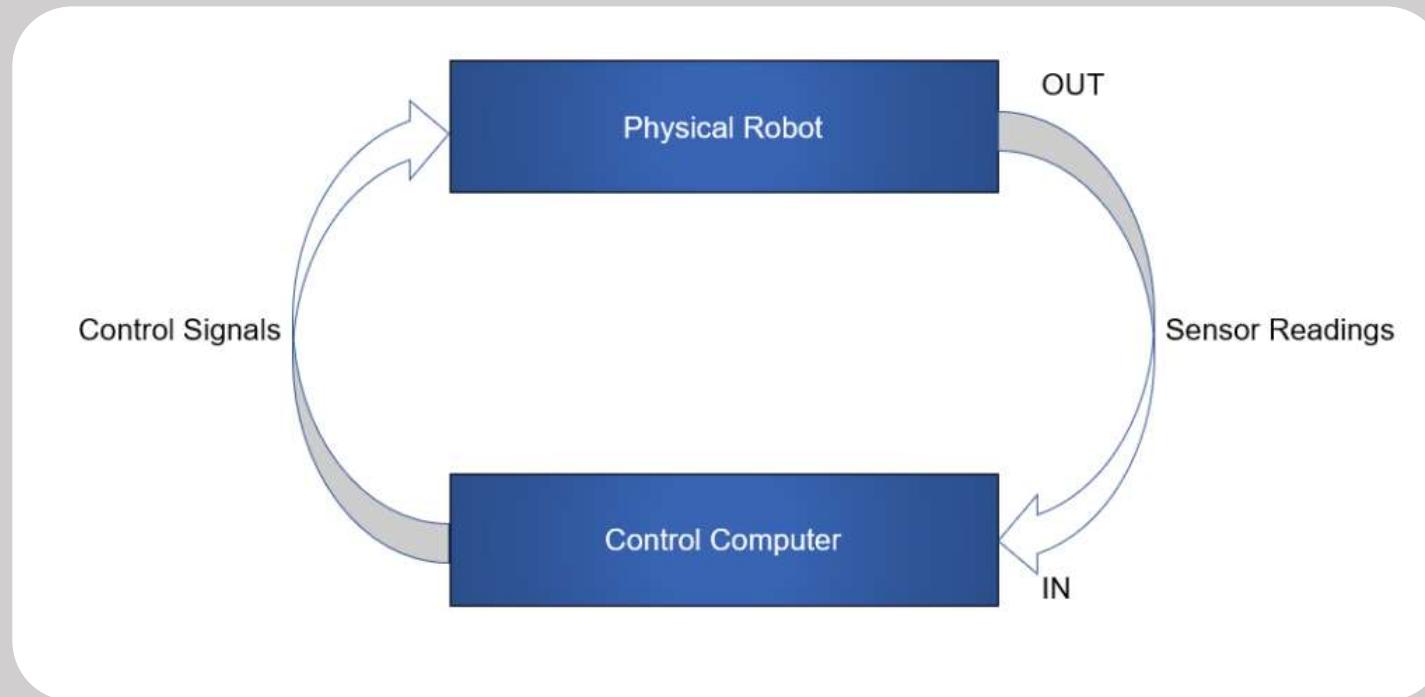
What is a Robot?

Basics blocks in a robot



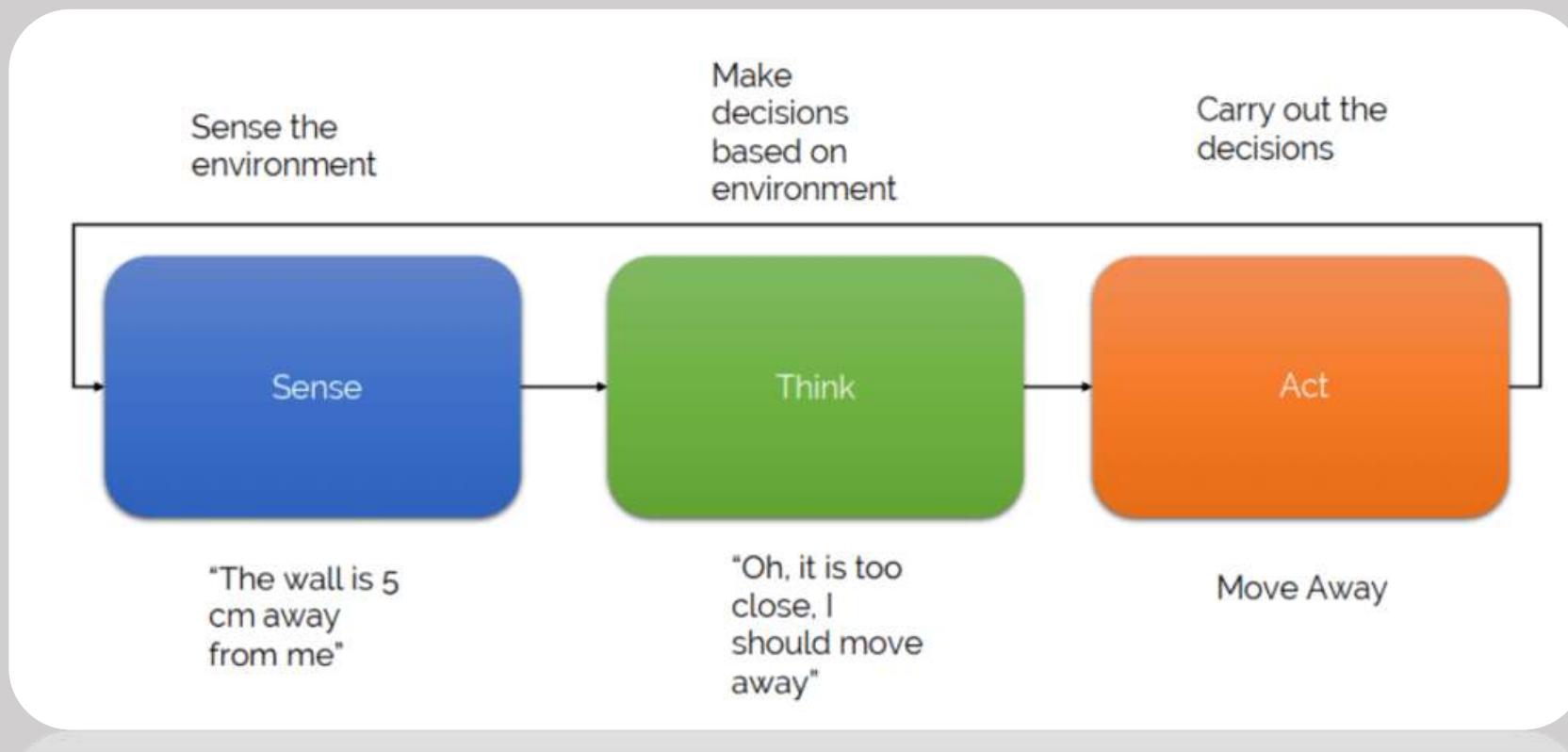
What is a Robot?

Robot feedback loop

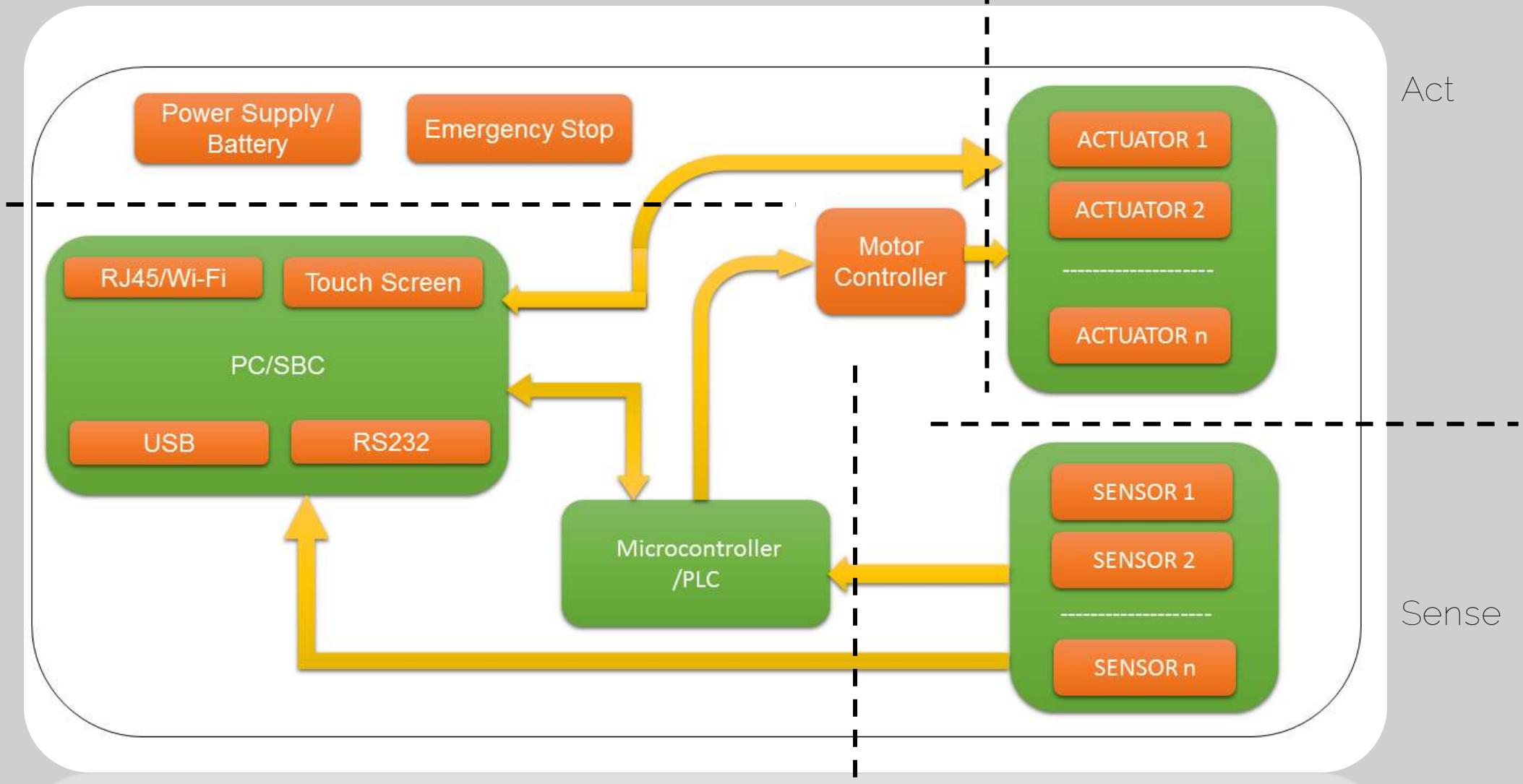


What is a Robot?

Block diagram of Obstacle avoidance robot



What is a robot? Inside view

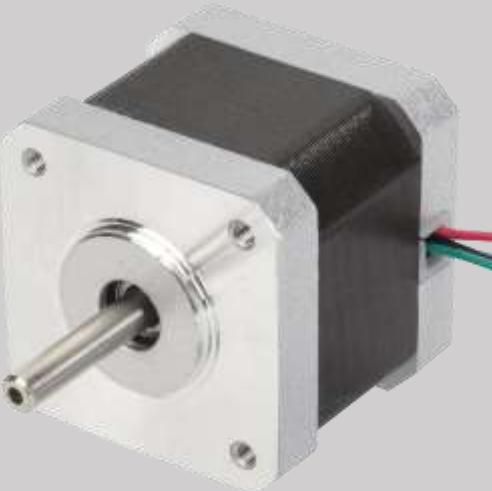


Robot Components

- Main Robot components
 - Sensors: Camera, Wheel encoders, Ultrasonic sensors, etc.
 - Actuators: Servo, Stepper, DC motors, etc.
 - Computer
 - PC: Intel NUC, Nvidia TX2
 - Microcontroller platform: Arduino board, Texas Launchpad, ARM controller based boards



RC Servo Motor



Stepper Motor



DC Gear Motor



Industrial Servo Motor



Dynamixel Smart Actuator



UTM - 30LX, Laser Scanner



Kinect, 3D Depth Sensor Intel RealSense, 3D Depth Sensor



Computing units for Robots

- X86, X86_64 Based PC
 - Intel NUC
 - Industrial PC
- Single board Computer
 - Nvidia TK1, TX1,TX2
 - Raspberry Pi
 - Odroid



Intel NUC



Raspberry Pi



Nvidia TX2



Odroid



Industrial PC

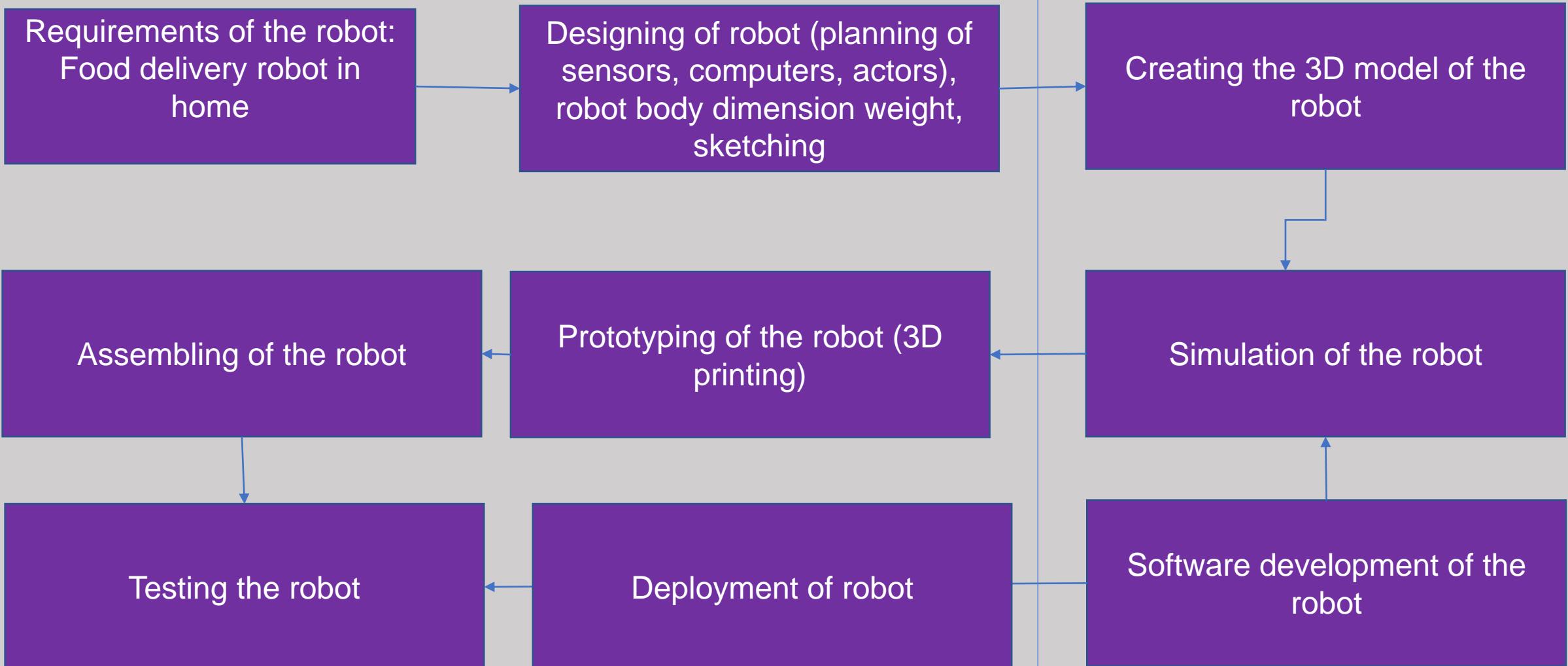
What is Robot Programming

- Robot Programming:
 - Modelling the behavior of a robot using a programming language
 - Robot sensor values as input to the model
 - Robot actuators as the output of the model.
- Programming languages: C++, Python etc.

Programming: Different Approaches

- Use existing software frameworks to program robots
 - Quick prototyping
 - Good for academic research
 - Need fine tuning for robotic products
 - Example: Arduino framework (low level), Robot Operating System (high level)
- Create everything from scratch using a programming languages
 - Time consuming
 - Better in the long run
 - Preferred Programming languages: C++ , Python

Robot Development Cycle



FREE Robotic Software frameworks

- Robot Operating System: Communication Middleware
 - Website: <http://www.ros.org/>
- Note: ROS is not a real Operating System but a meta operating system
- Open-CV: Computer Vision library
 - Website: <https://opencv.org/>
- PCL: Point Cloud Library
 - Website: <http://pointclouds.org/>



FREE Robotic Software frameworks

- Gazebo: Robot simulator
 - Website: <http://gazebosim.org/>
- Open-Rave: Robot framework for motion planning
 - Website: <http://openrave.org/>



FREE Robotic Software frameworks

- Webots: Robot simulator
 - Website: <https://cyberbotics.com/>
- V-REP: Robot simulator
 - Website: <http://www.coppeliarobotics.com/>



Popular Robotic Programming Languages

- C++



- Python



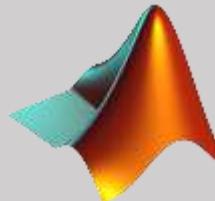
- Java



- C#/ .NET



- MATLAB



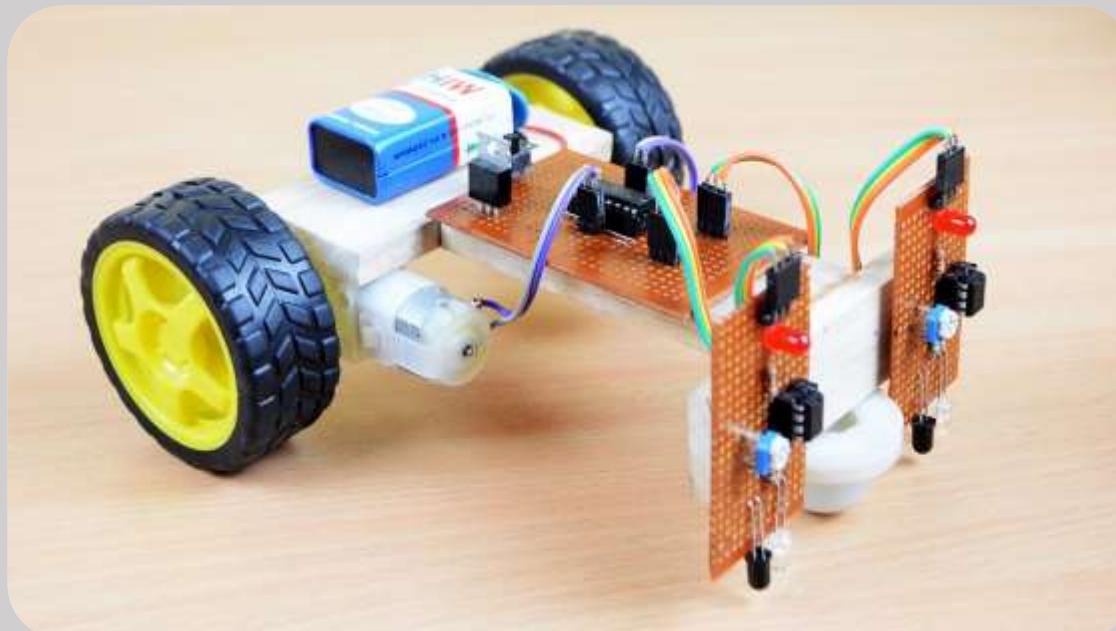


Why we need an OS in Robots?

Discussing the importance of an OS in a robot

When we don't need an OS in a robot?

- Brain only have a microcontroller
- Brain only using digital circuits

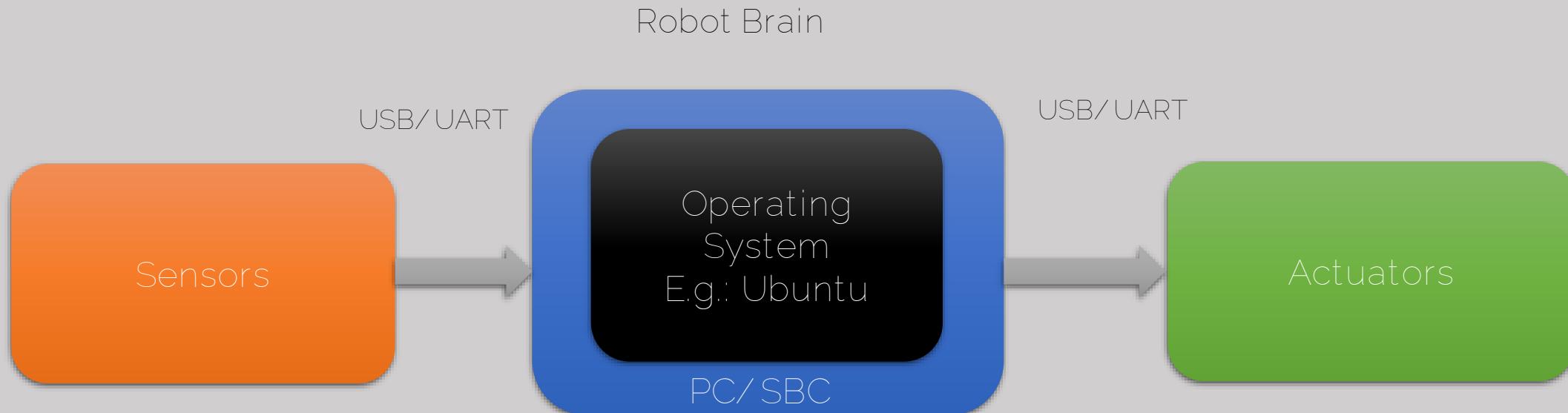


Why we need an OS in Robots?



Why we need an OS in Robots?

- Operating system (OS) is running on robot brain/PC
- OS is responsible for scheduling the tasks in a computer
- It can receive input from robot sensors | Perform computation | Take decision | Send command to actuator



Why we need an OS for Robots?

- Robot software is running on Operating System
- Low level control
- Support for robot framework
 - E.g.: Robot Operating System (ROS), Open-CV, PCL etc.
- More flexibility in programming robots
- Secure
- Realtime processing



List of Operating System used in Robots

- GNU/Linux variants
 - Ubuntu (<https://www.ubuntu.com/>)
 - Debian (<https://www.debian.org/>)
- Windows CE, 7 & 10



- Mac OS X



- QNX Real time operating system (<https://blackberry.qnx.com/en>)

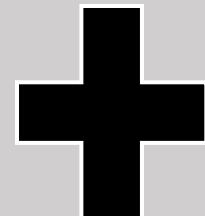


- VxWorks: (<https://www.windriver.com/products/vxworks/>)



Prerequisites to learn ROS

- Knowledge in Linux terminal commands
- Knowledge in C++ or Python
- Do you have these skills ??





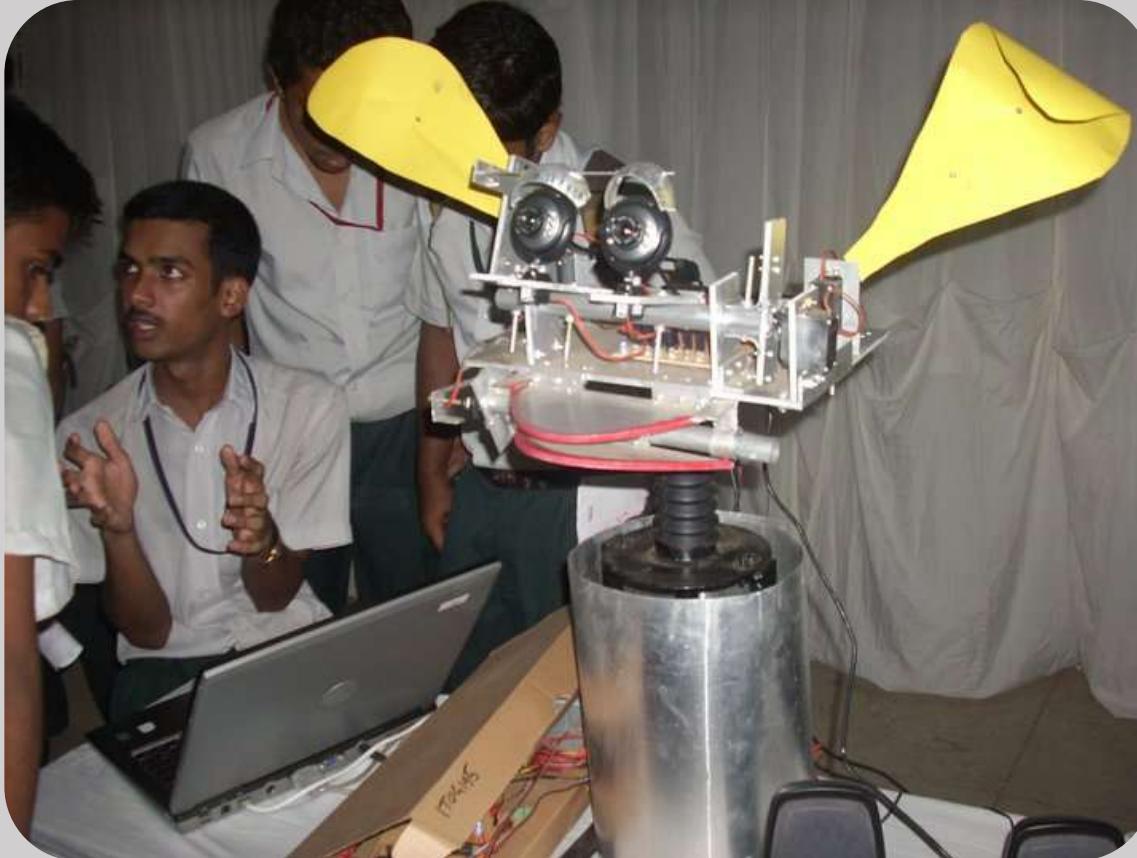
What is Robot Operating System (ROS)

Discussing brief idea of ROS

Let's start with a project



A robot working without R.O.S



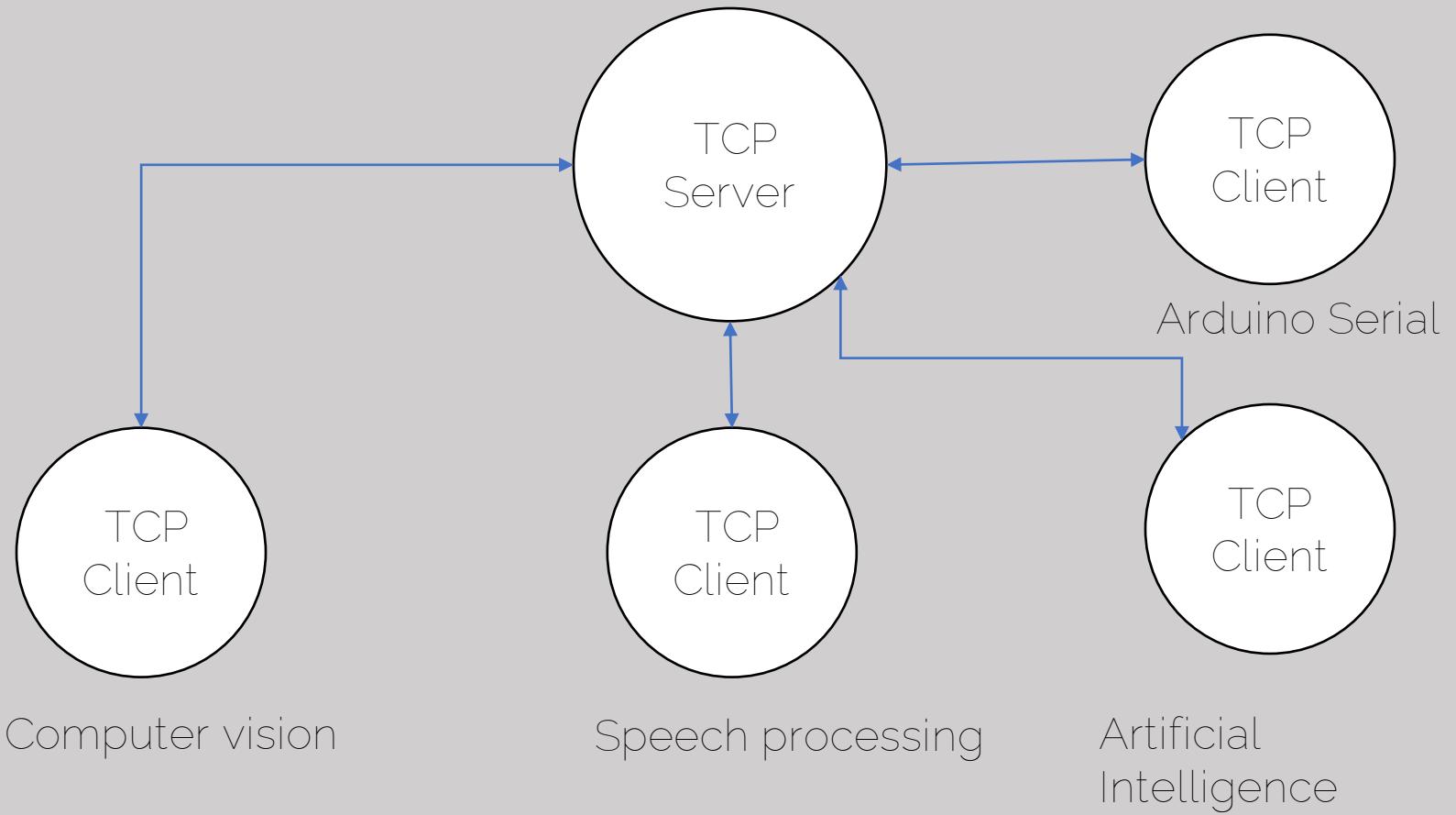
- Type of Robot : Social Robot
- Sensors : Dual VGA camera, Microphone
- Actuators : 12 RC – Servos
- Robot Brain : 2 X Laptops
- Capabilities :
 - Interacting with people using natural language using AI, speech recognition/synthesis and express emotions
 - Face detection, motion detection and object detection
- Programming languages : Python, C++
- Communication of processes using Python Server, client architecture



Challenges Faced !!!!

- Interprocess communication from different machines
- Synchronization of robot task
- Lack of Modularity
- Difficult to reuse the code
- Difficult to develop application on top of the existing codes

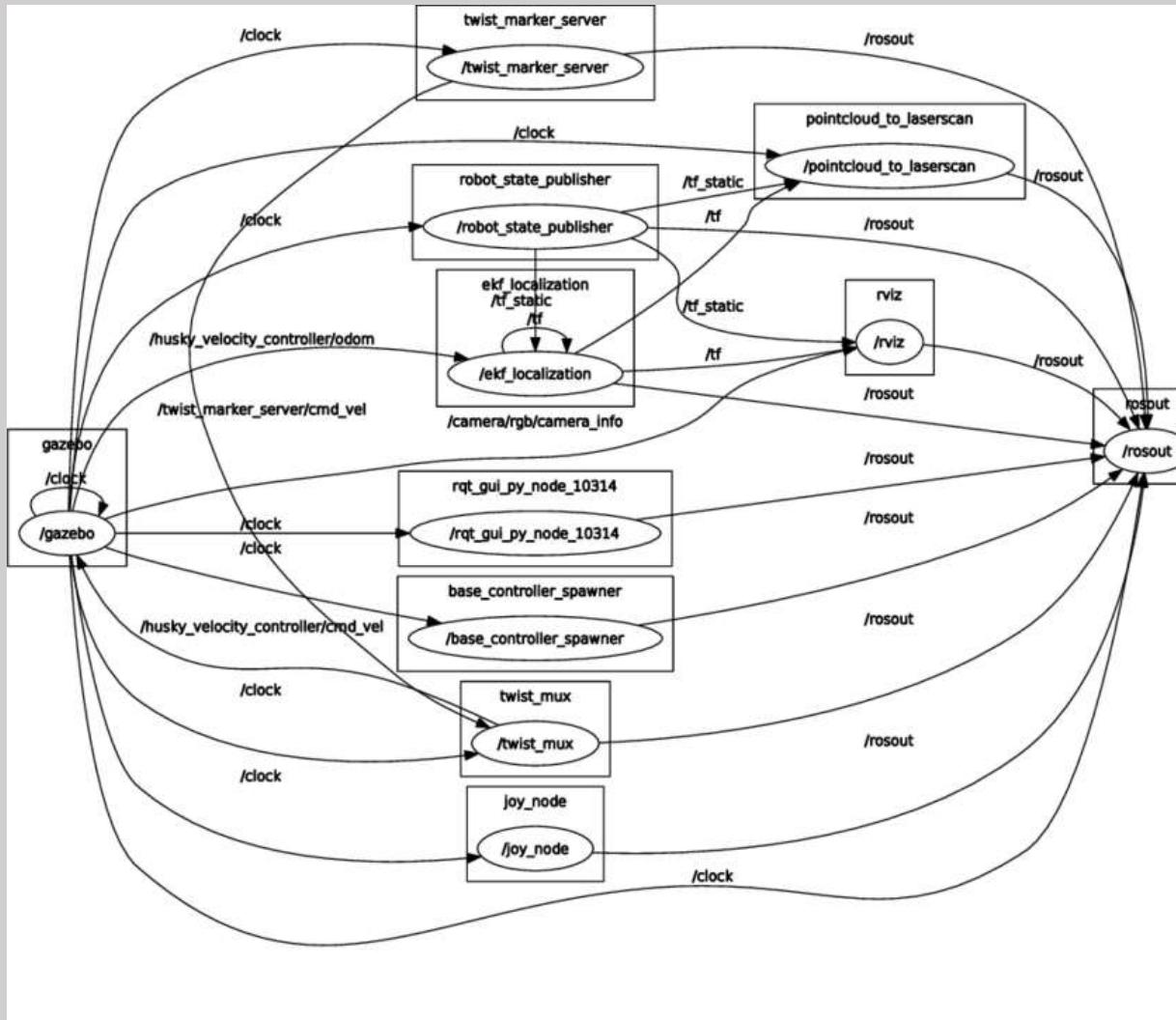
Software Communication diagram



A robot working on R.O.S



A robot working on R.O.S



- Type of Robot : Autonomous Mobile Robot(Chefbot)
- Sensors : Kinect 3D sensor, Microphone, Ultrasonic, IMU
- Actuators : DC-Geared motor with Encoder
- Robot Brain : Intel NUC mini PC
- Capabilities :
 - Autonomous navigation :One of the application is food serving
 - Interacting with people using natural language using AI, speech recognition/synthesis and express emotions
- Programming languages : Python, C++
- Completely running on R.O.S platform



Issues Faced !!!!

- Interprocess communication from different machines
- Synchronization of robot task
- Lack of Modularity
- Difficult to reuse the code
- Difficult to develop application on top of the existing codes

What is Robot Operating System?

- Open Source Robotic Software framework
- Robotics Middleware, not a real operating system
- Need a host operating system to run
- Website (<http://ros.org>)



Installing ROS on Ubuntu

- Ubuntu Version : 20.04 LTS 64 bit, Focal
- ROS Distribution : ROS Noetic Ninjemys(Stable)



+



Installing ROS on Ubuntu

- Installation link : <http://wiki.ros.org/noetic/Installation>

1. Installation

1.1 Configure your Ubuntu repositories

Configure your Ubuntu repositories to allow "restricted," "universe," and "multiverse." You can [follow the Ubuntu guide for](#) instructions on doing this.

1.2 Setup your sources.list

Setup your computer to accept software from packages.ros.org.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

Mirrors: Source Debs are also available

1.3 Set up your keys

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B  
172B4F42ED6FBAB17C654
```

If you experience issues connecting to the keyserver, you can try substituting `hkp://pgp.mit.edu:80` or `hkp://keyserver.ubuntu.com:80` in the previous command.

Alternatively, you can use curl instead of the apt-key command, which can be helpful if you are behind a proxy server:

Testing the ROS installation

- \$ roscore – Test this command in terminal to verify ROS installation

```
turtlebot@turtlebot-X200CA:~$ roscore
... logging to /home/turtlebot/.ros/log/6ef6185c-9127-11e4-83da-0c84dc11754b/roslaunch-turtlebot-X200CA-9168.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.0.8:45853/
ros_comm version 1.11.9

SUMMARY
=====

PARAMETERS
* /rostdistro: indigo
* /rosversion: 1.11.9

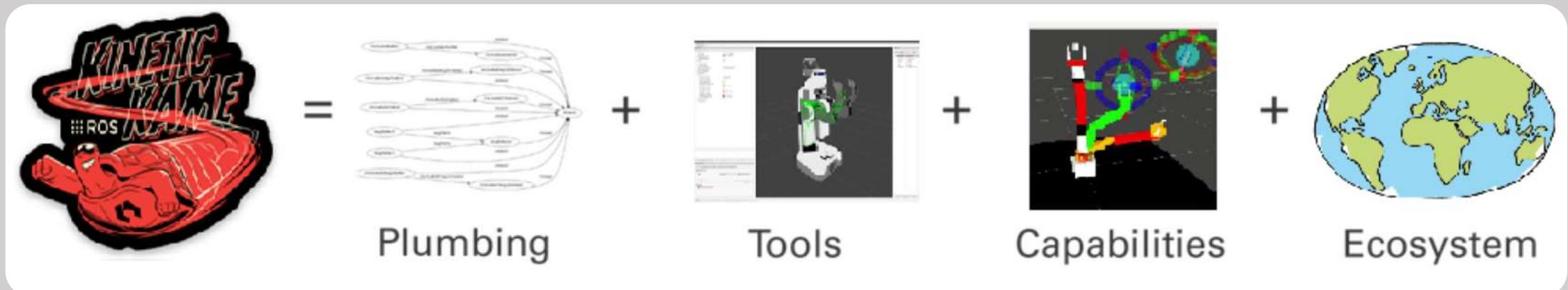
NODES

auto-starting new master
process[master]: started with pid [9180]
ROS_MASTER_URI=http://192.168.0.8:11311/

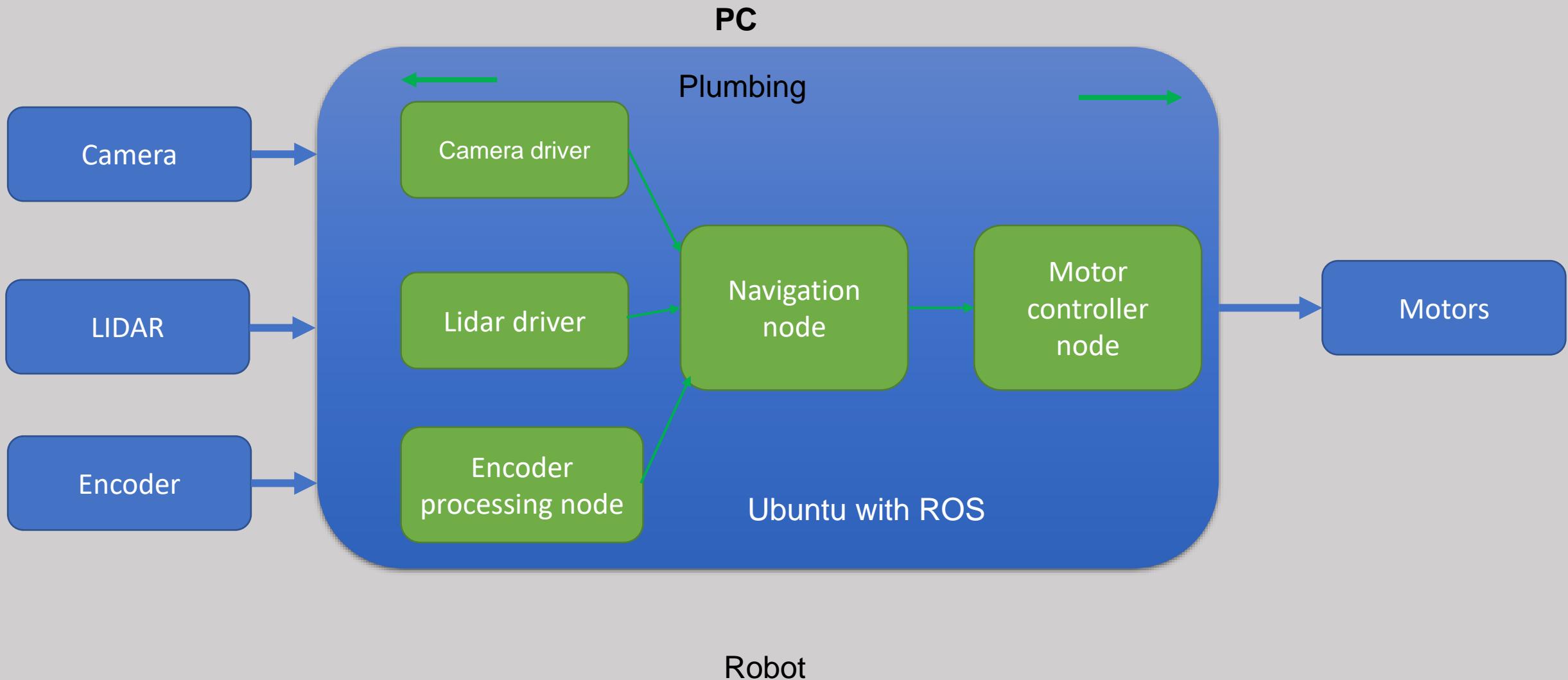
setting /run_id to 6ef6185c-9127-11e4-83da-0c84dc11754b
process[rosout-1]: started with pid [9193]
started core service [/rosout]
```

ROS Equation

- ROS Equation

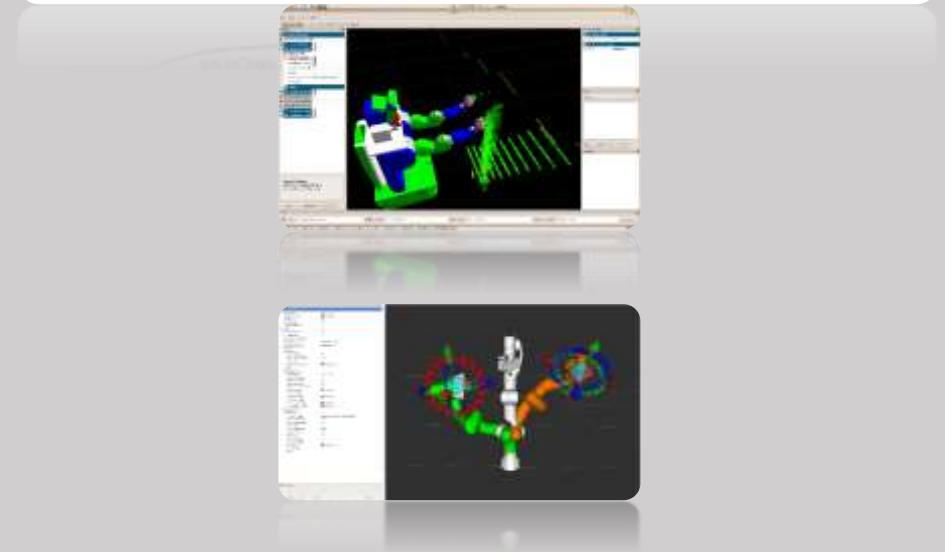
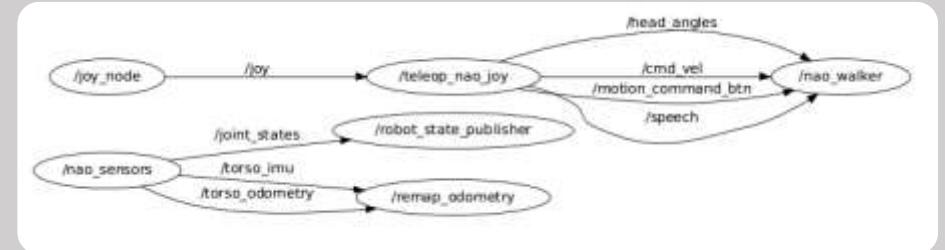


What is Plumbing?



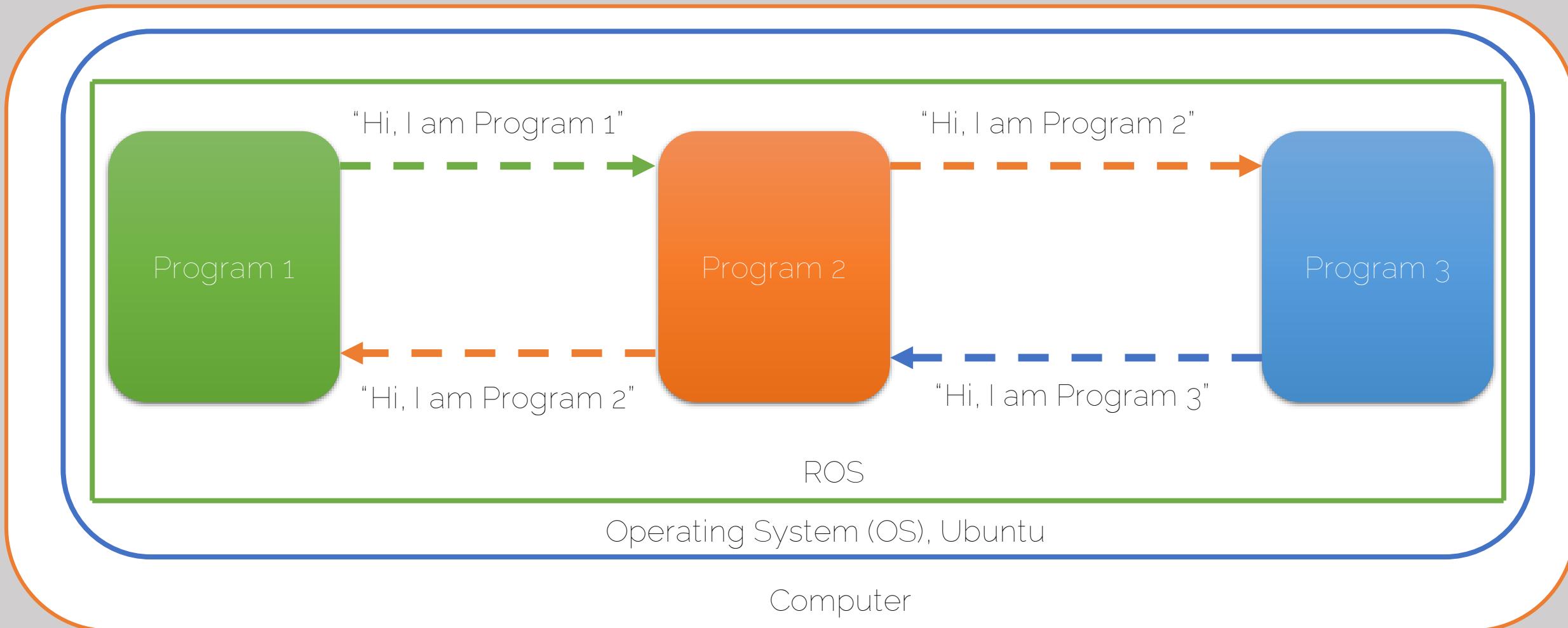
What is Robot Operating System?

- Plumbing/Communication middleware:
 - Inter-process communication
- Tools:
 - Visualization and Debugging of robot data
 - Example: Rviz, Rqt
- Capabilities:
 - Robot manipulation and navigation
 - Example: MoveIt!, and navigation stack
- Ecosystem:
 - World wide ROS developers
 - ROS Wiki
 - ROS Q & A



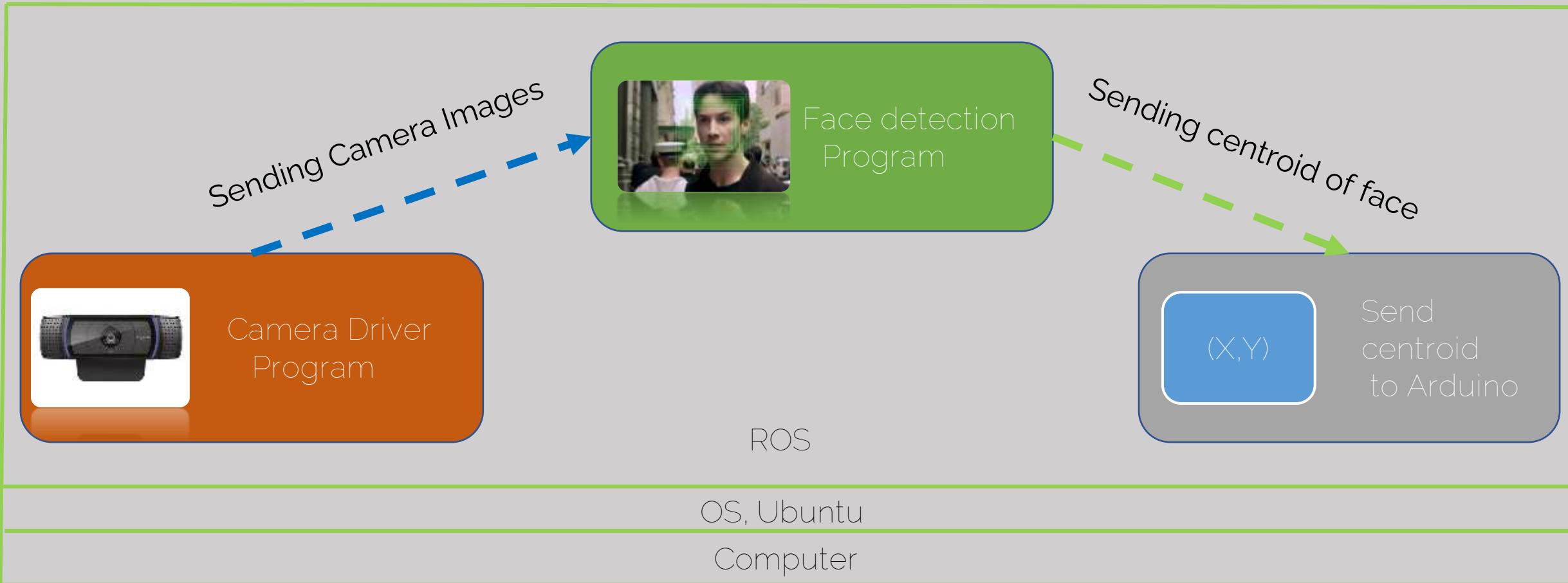
Working of ROS: Plumbing

- Inter-process Communication



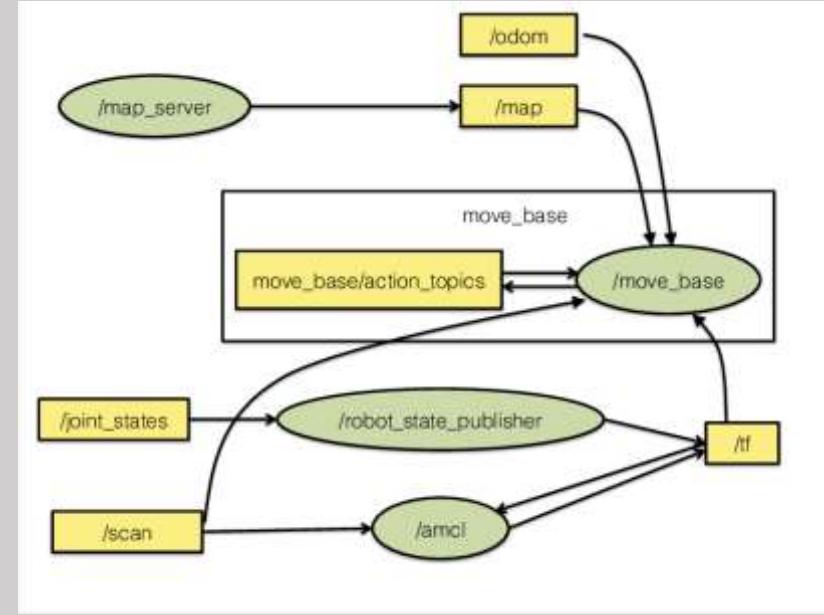
Working of ROS: Plumbing/Middleware

- Example: Face Detection in ROS

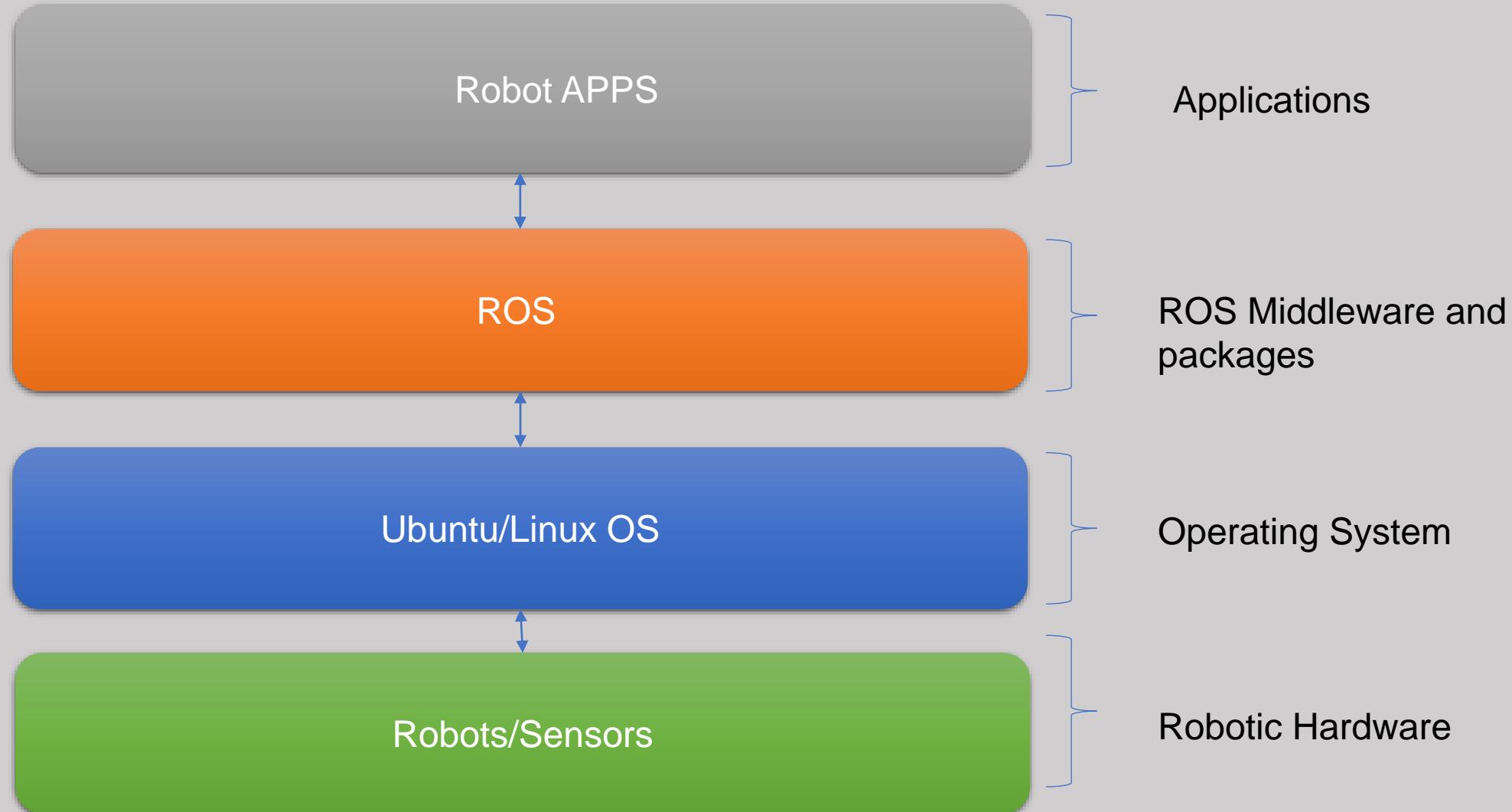




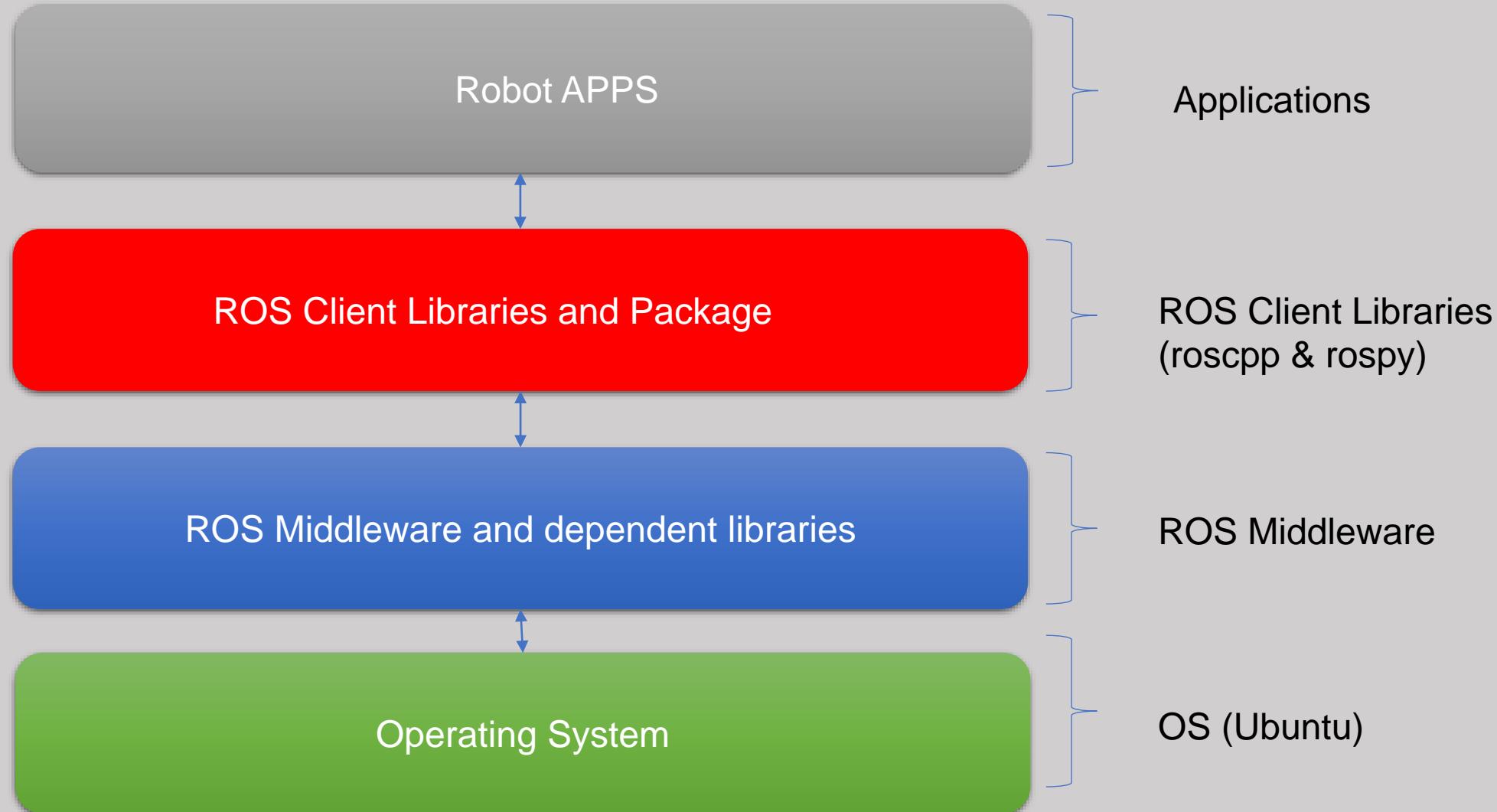
How ROS Works?



ROS Software Architecture

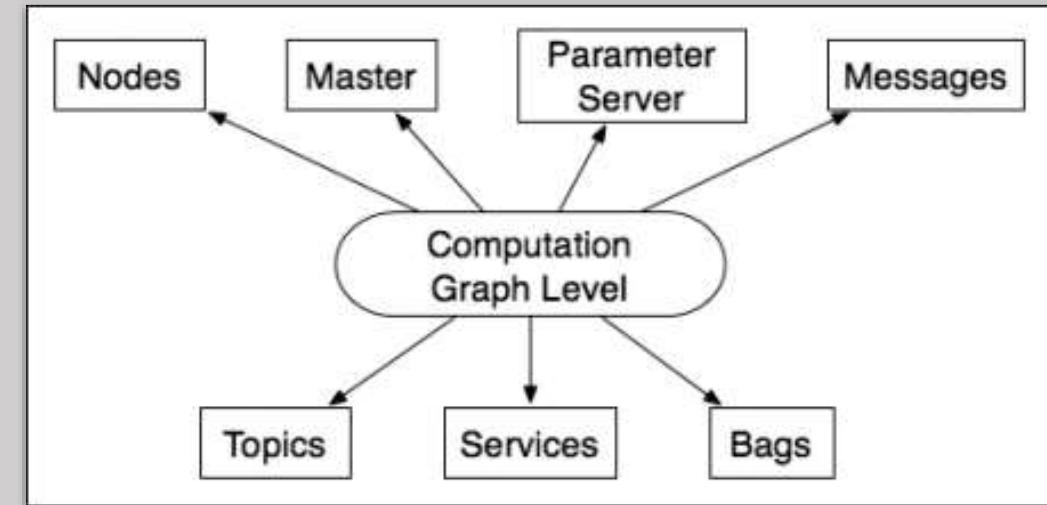


ROS Software Architecture

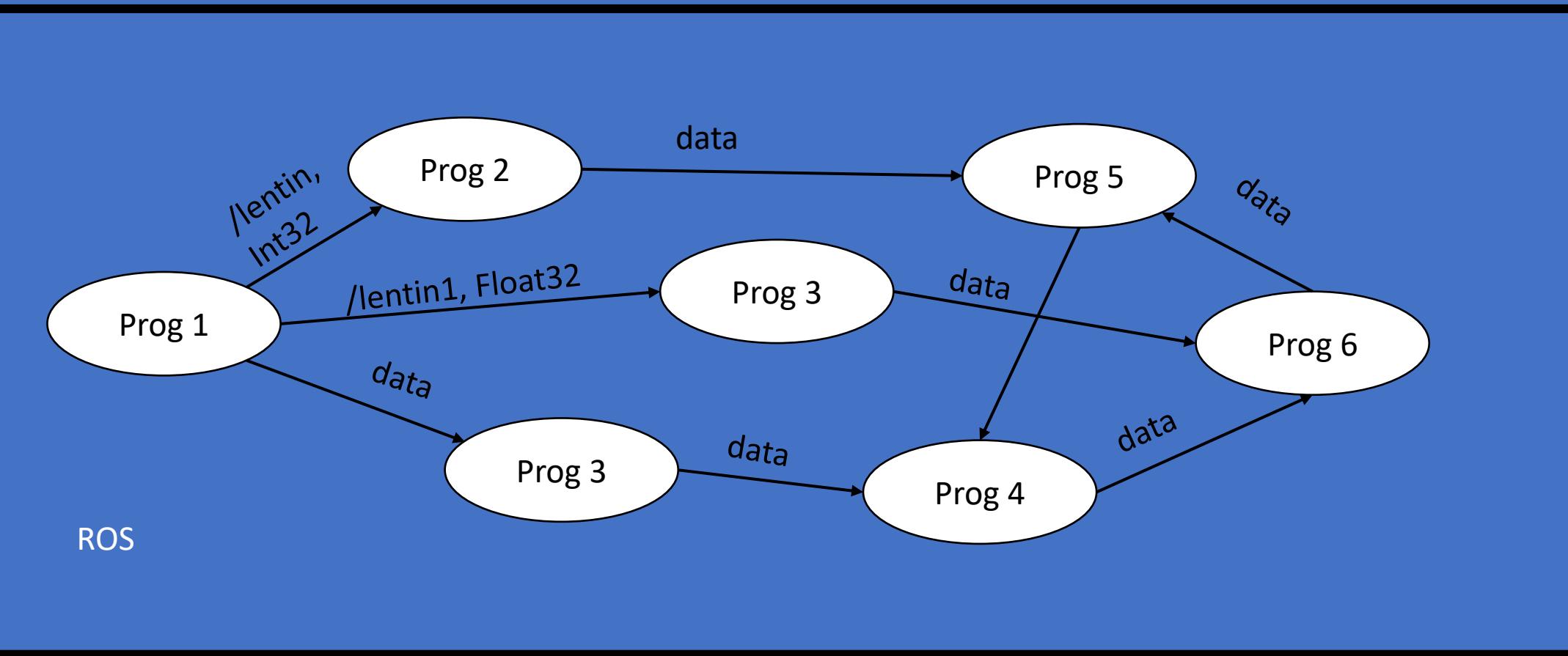


What is a ROS Computation Graph? [Demo]

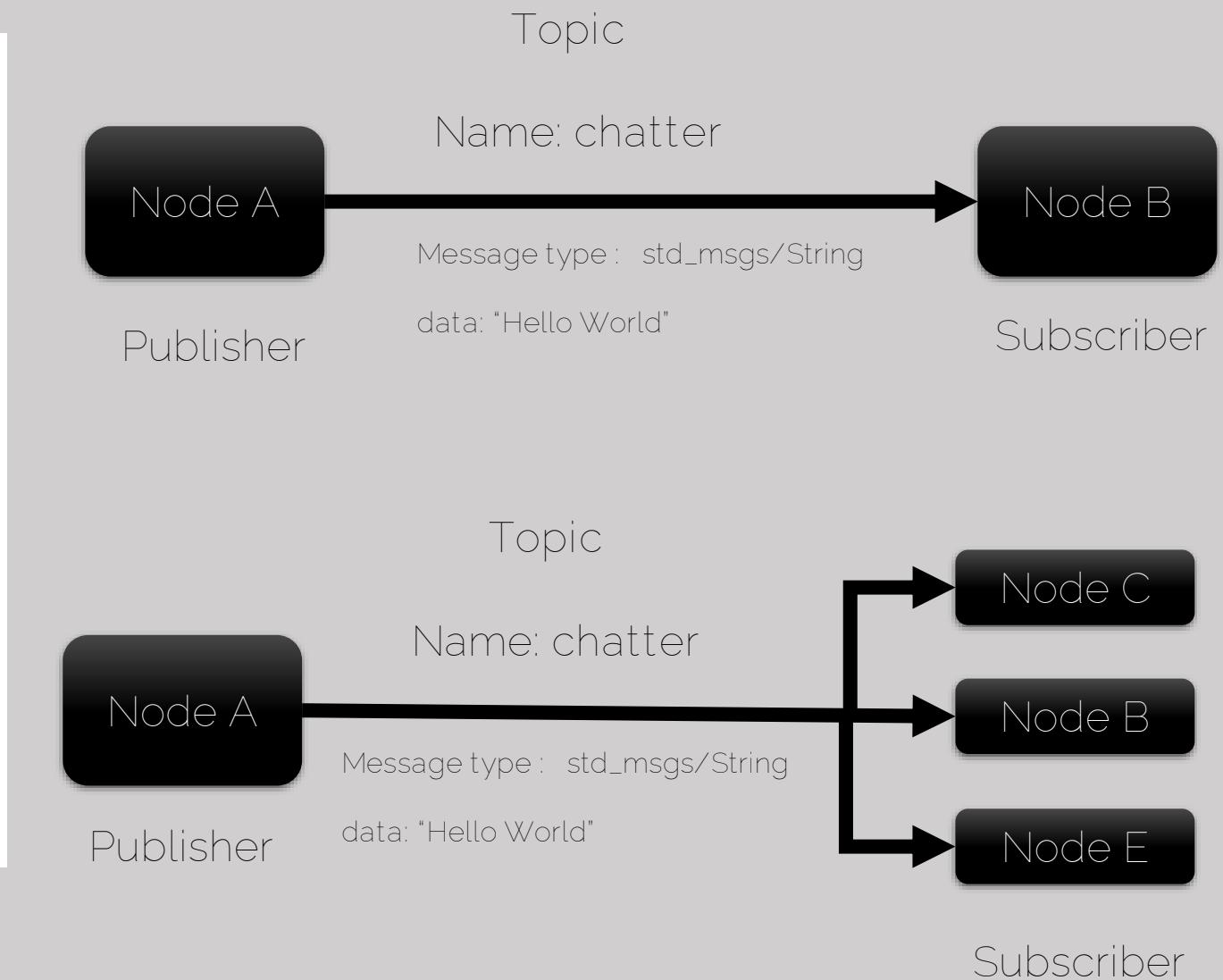
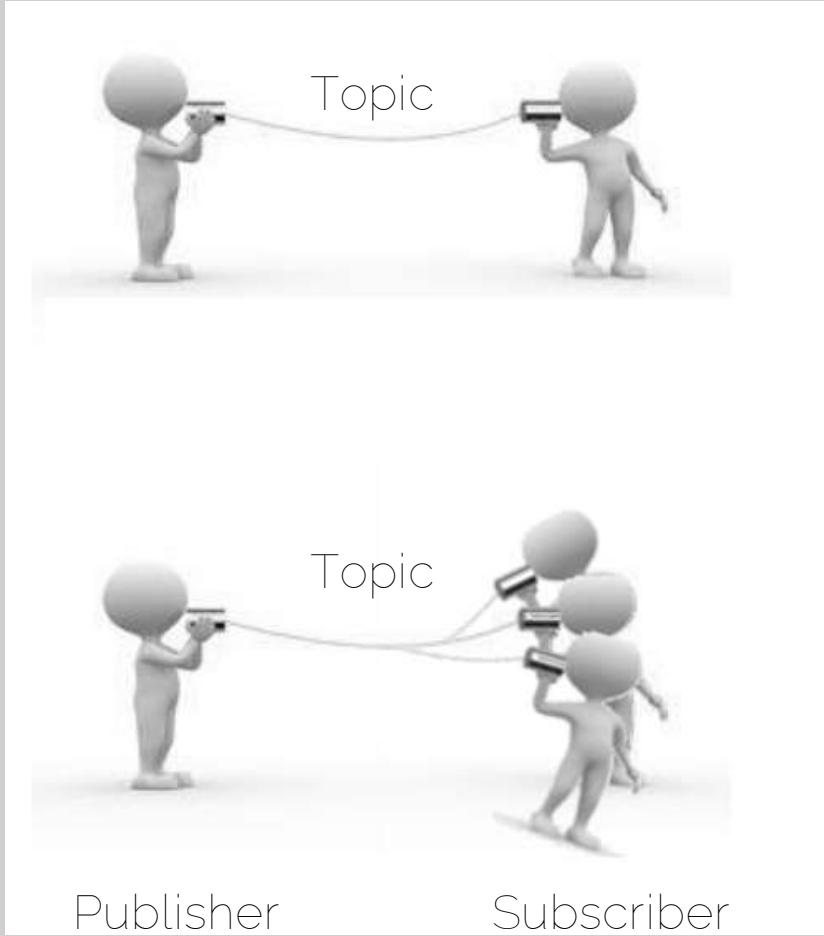
- ROS Computation Graph:
 - *It is the network of ROS process which is doing the computation in ROS together by connecting peer-to-peer*
 - Various concept in the graph are
 - Nodes
 - Master
 - Parameter Server
 - Messages
 - Topics
 - Actions
 - Services
 - Bags



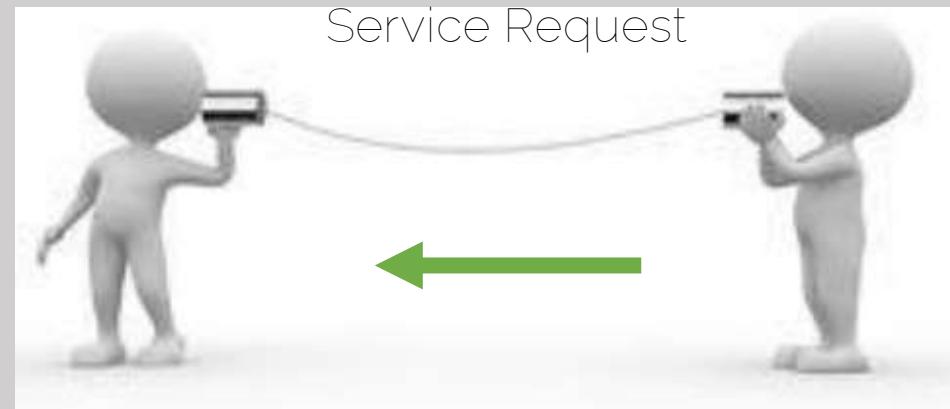
ROS Computational Graph



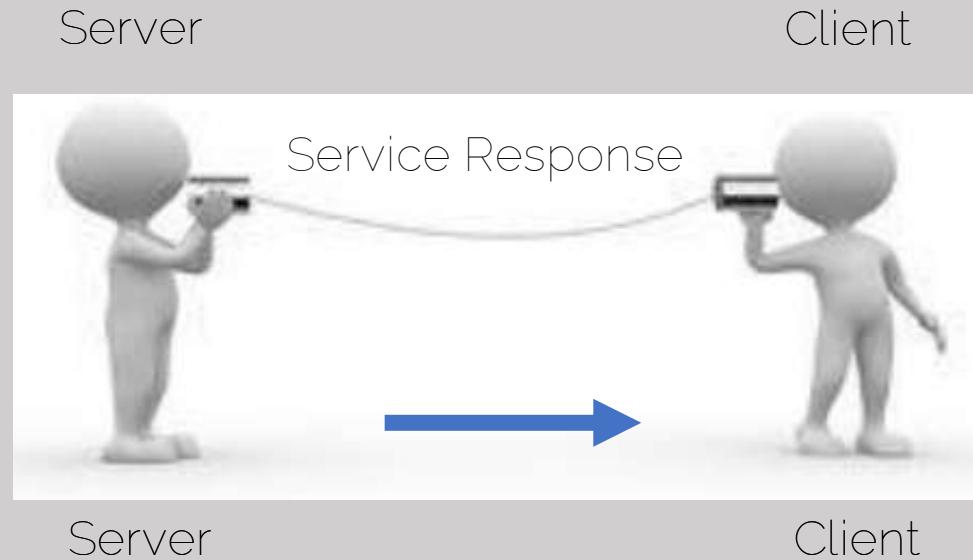
What is a ROS Topic?



What is a ROS Service?



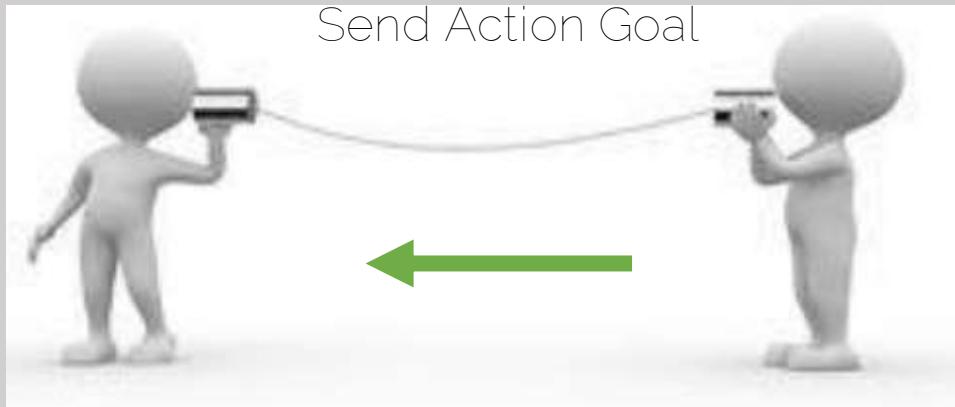
Hey
Server,
What is
 $2+2$



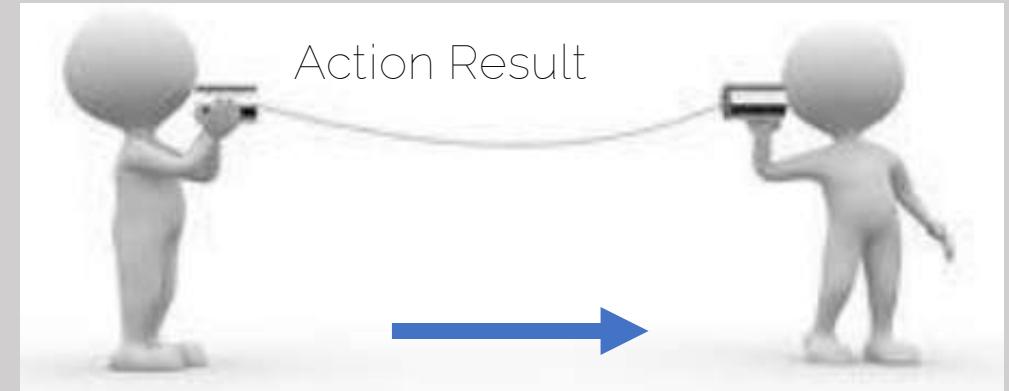
Thanks for
contacting
me, The
answer is 4

What is a ROS Action?

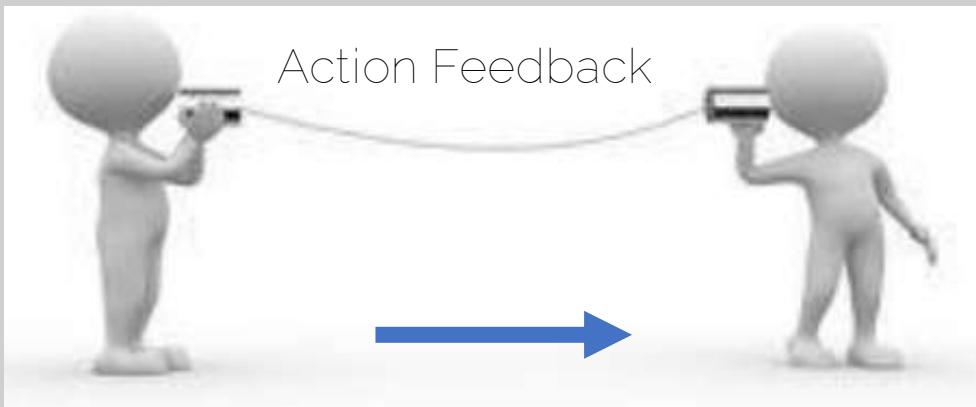
1)



3)



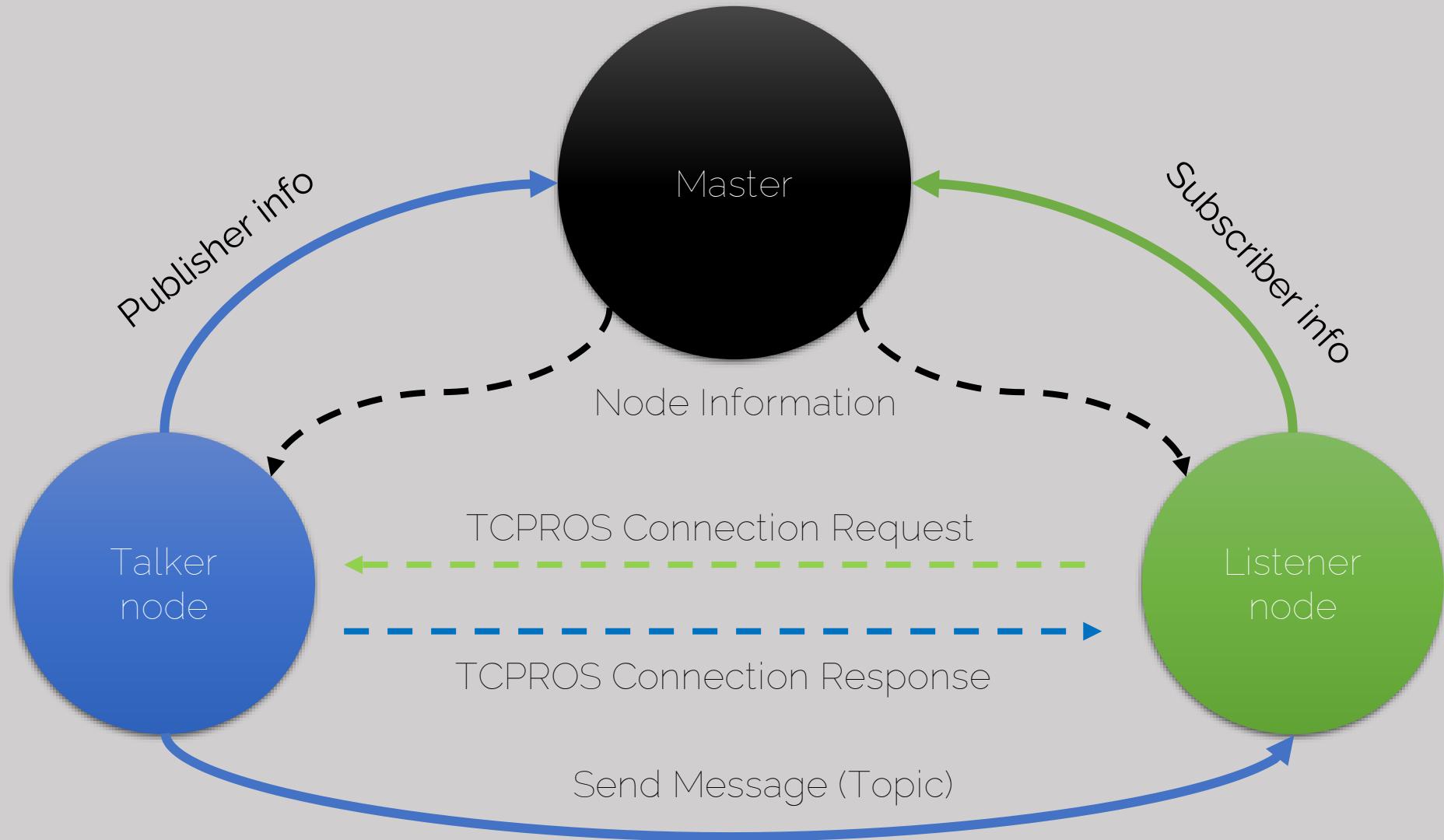
2)



Server

Client

ROS Communication: Topic



ROS Communication: Topic

1. Running ROS Master

```
$ roscore
```

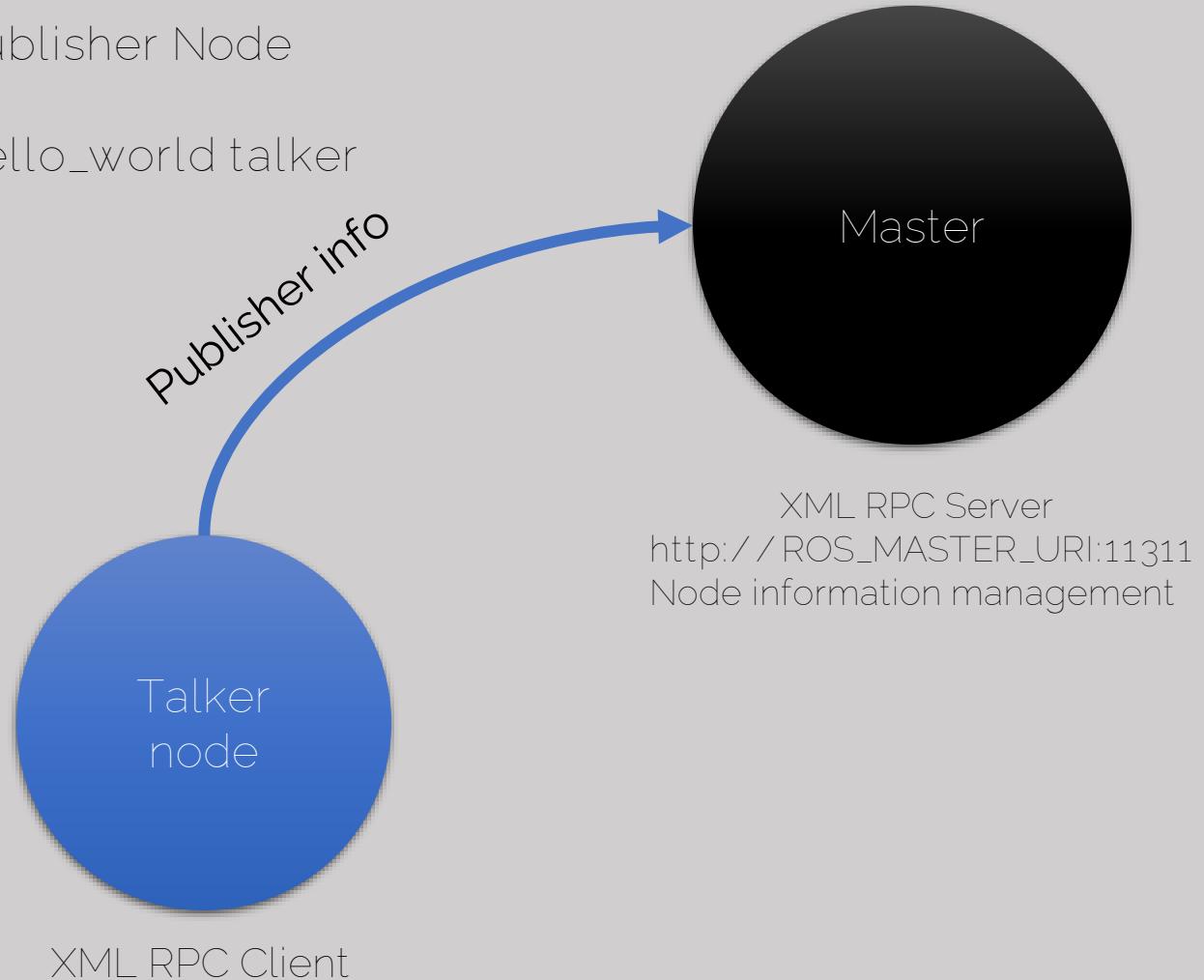


XML RPC Server
`http://ROS_MASTER_URI:11311`
Node information management

ROS Communication: Topic

2. Running a Publisher Node

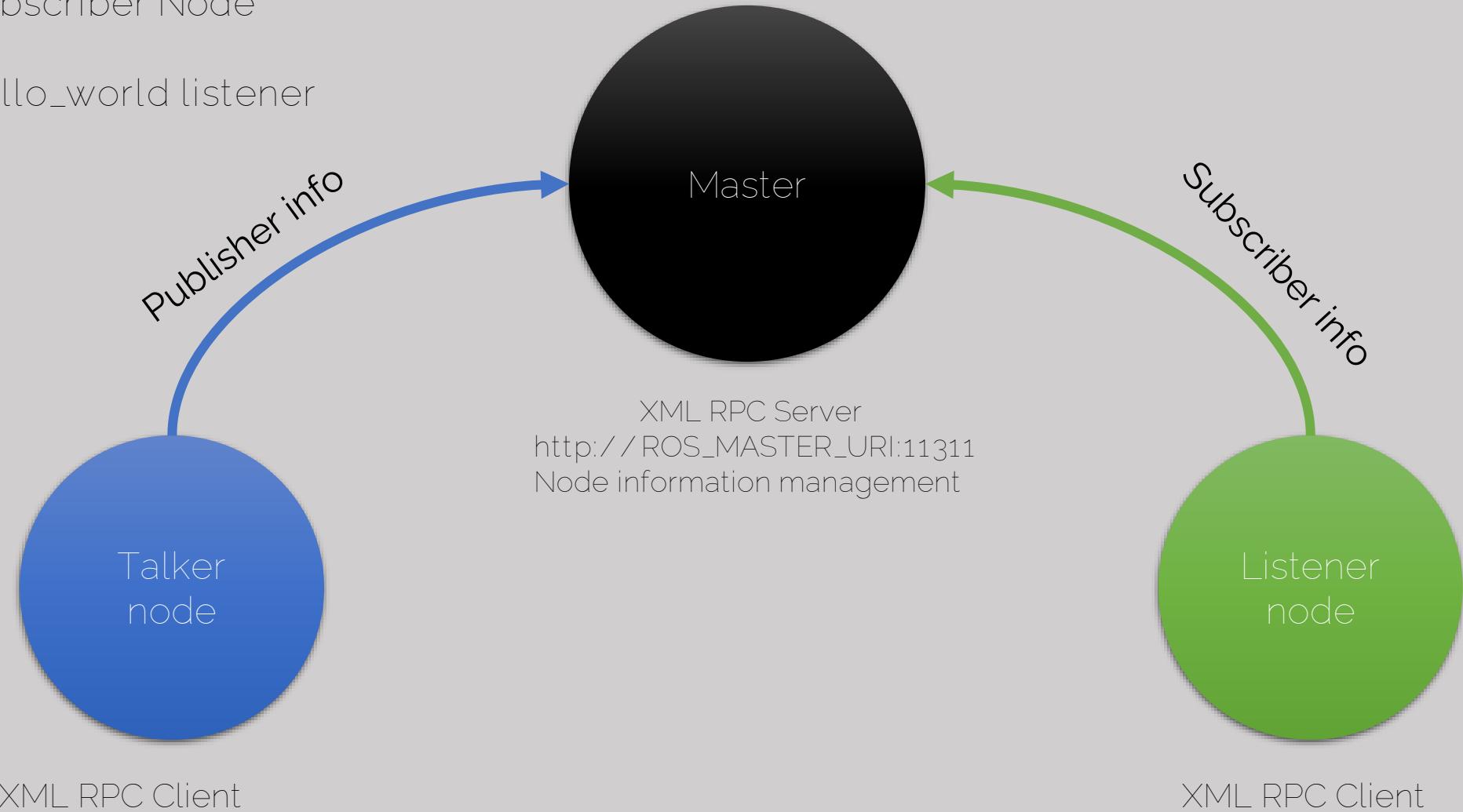
E.g. `$ rosrun hello_world talker`



ROS Communication: Topic

3. Running a Subscriber Node

E.g. `$ rosrun hello_world listener`

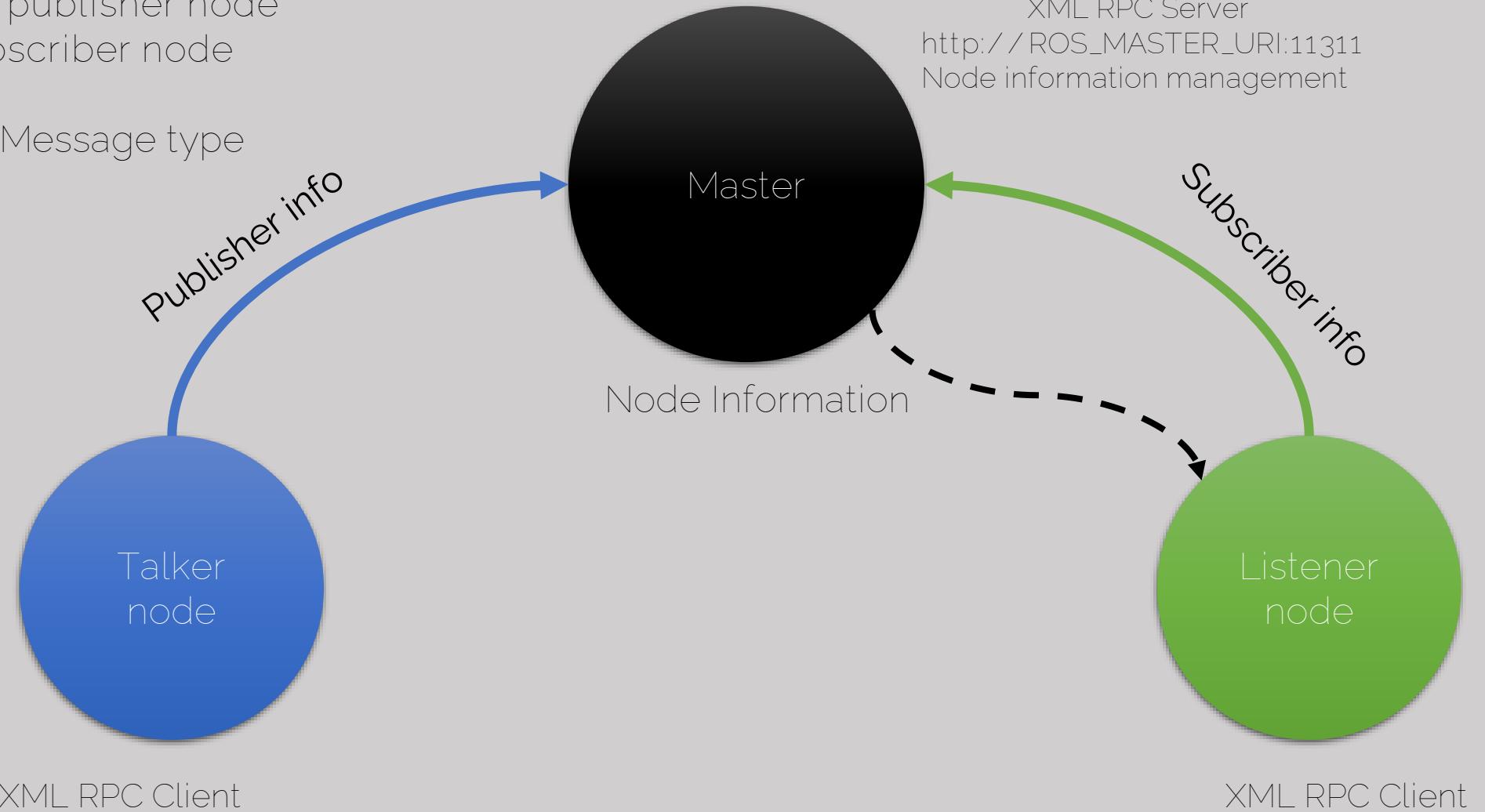


ROS Communication: Topic

4. Master Sending publisher node information to subscriber node

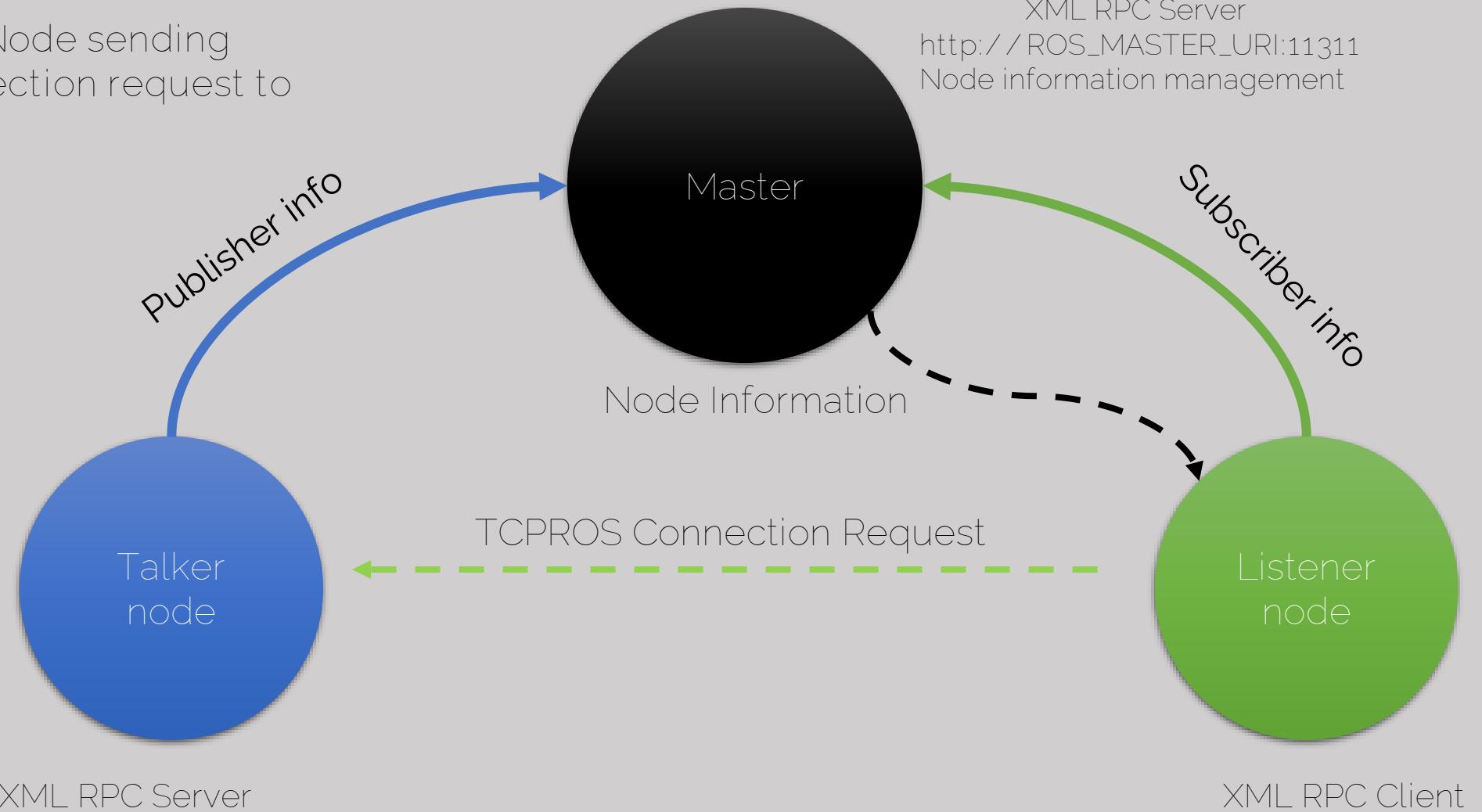
Node info: Topics, Message type etc.

XML RPC Server
`http://ROS_MASTER_URL:11311`
Node information management



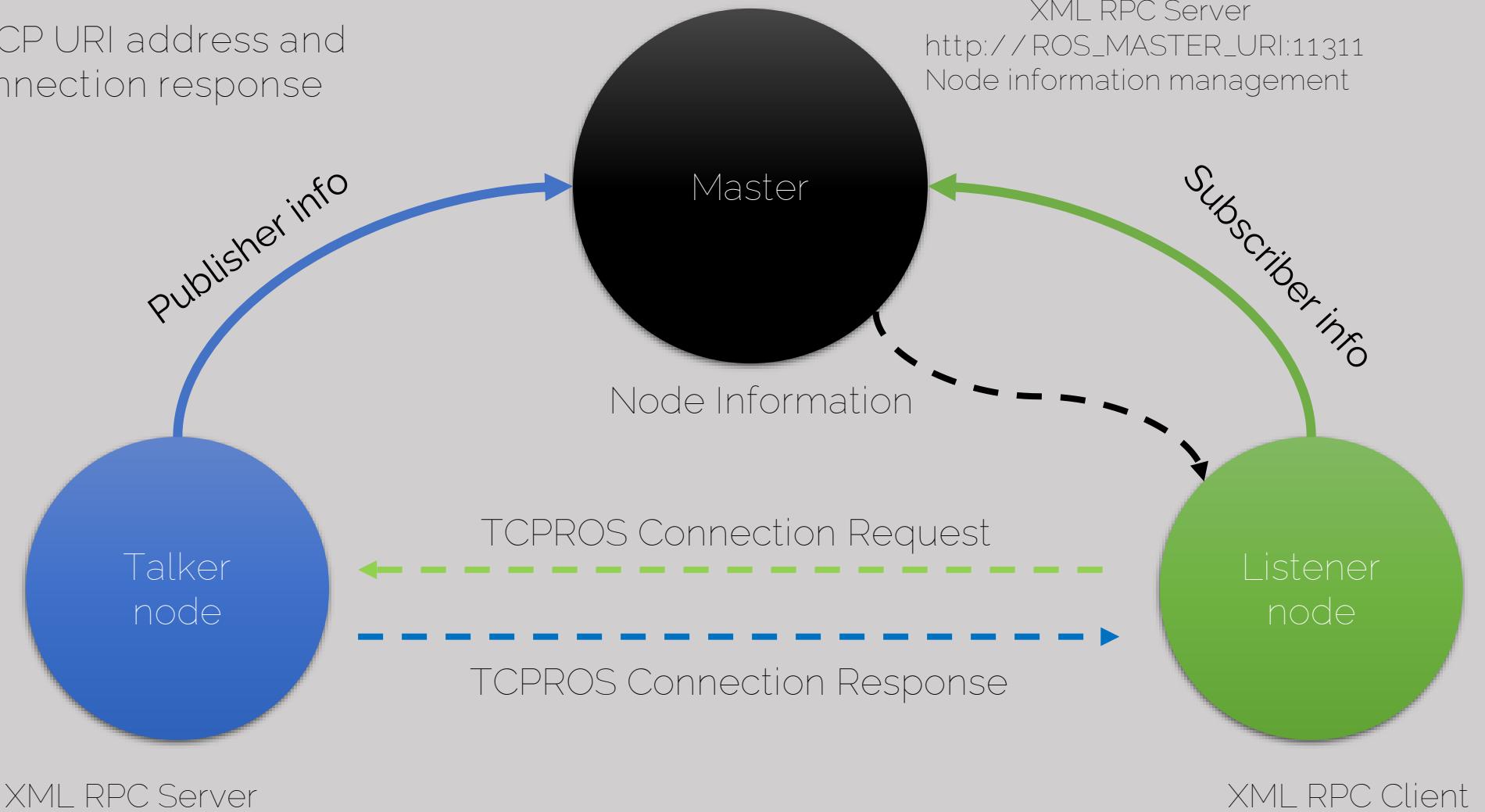
ROS Communication: Topic

5. Subscriber Node sending
TCPROS connection request to
Publisher



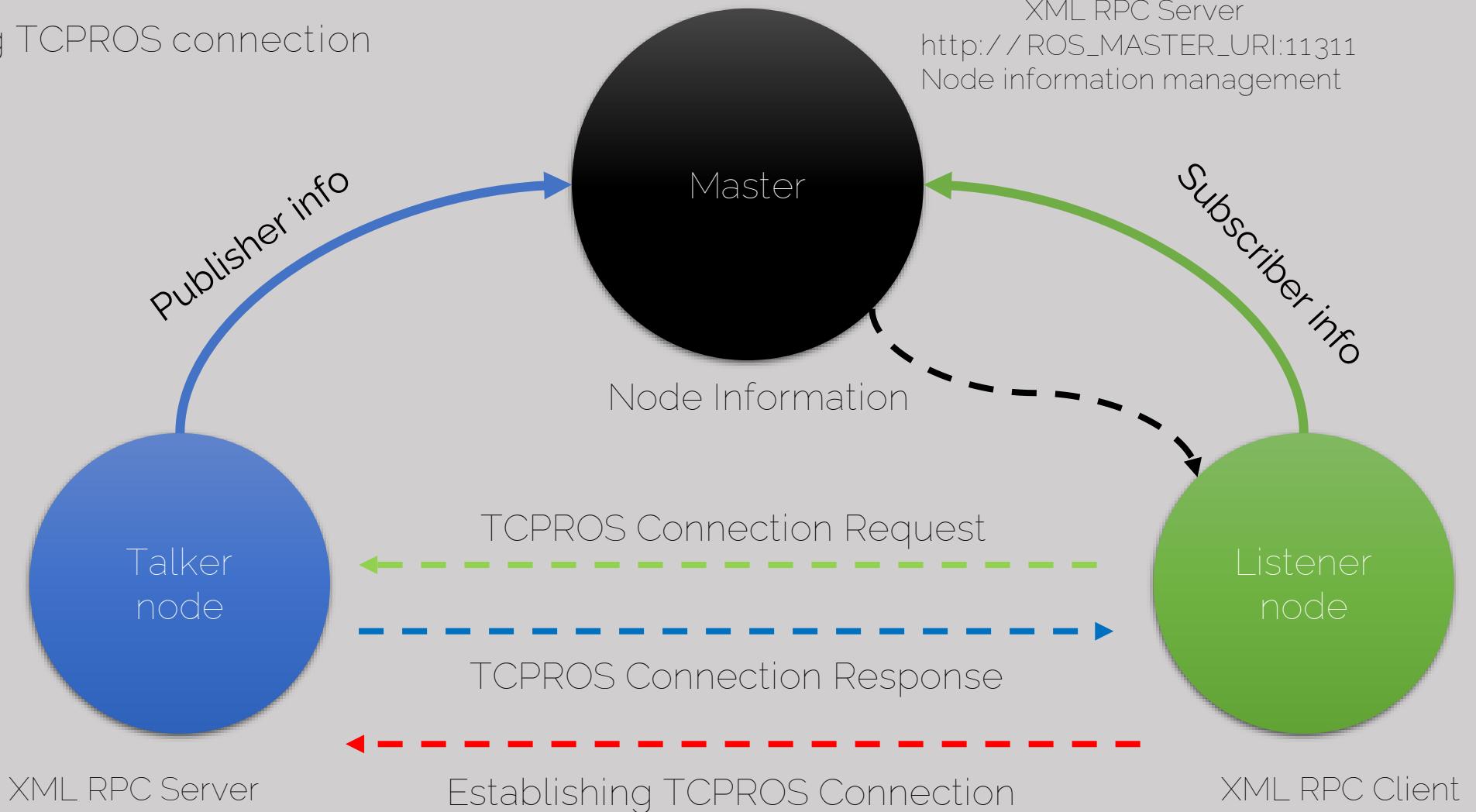
ROS Communication: Topic

6. Returning TCP URI address and port as the connection response



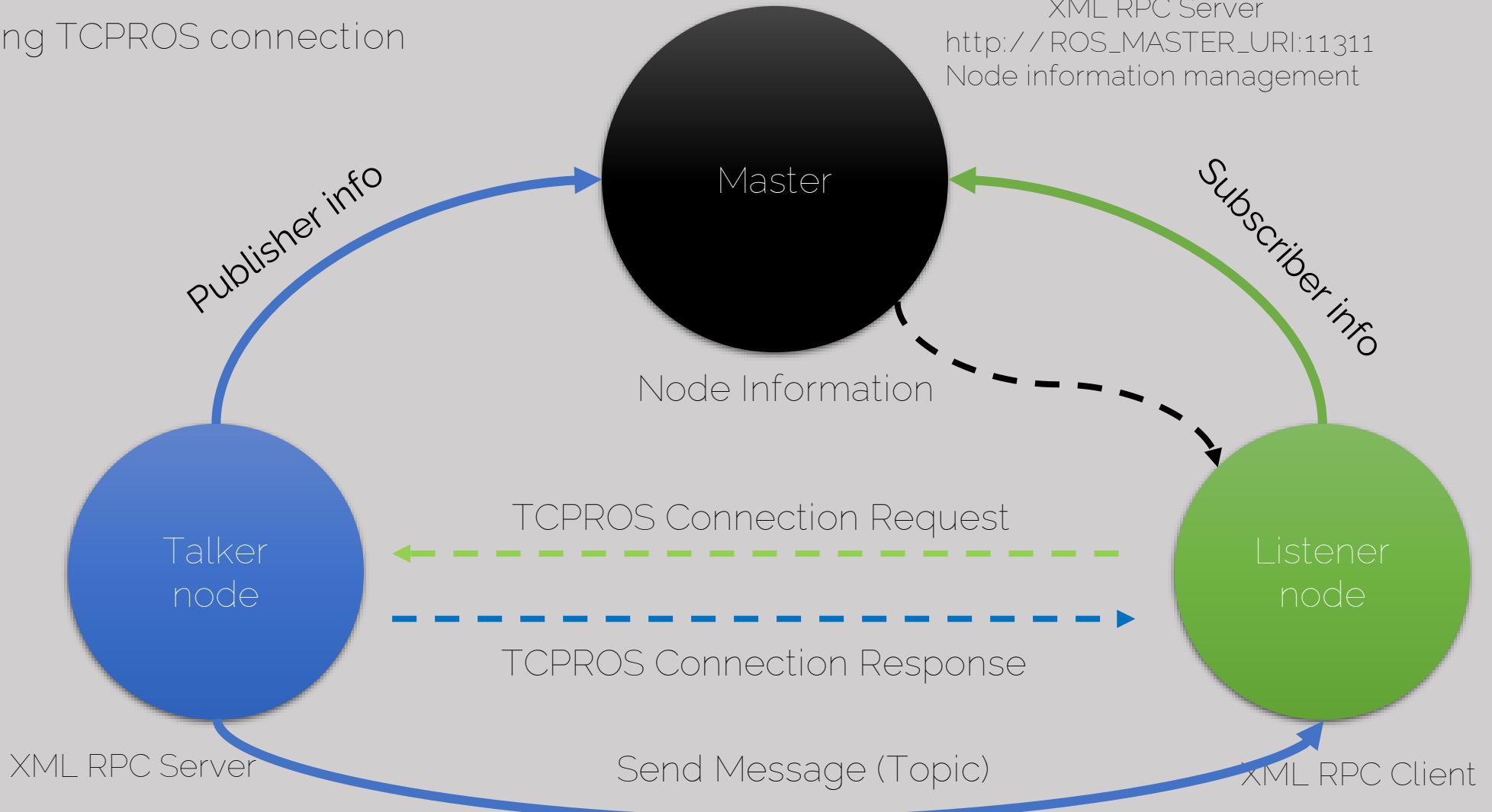
ROS Communication: Topic

7. Establishing TCPROS connection



ROS Communication: Topic

8. Establishing TCPROS connection



ROS Communication: Topic

9. Publishing and Subscribing topic



roscpp tutorials: talker & listener

The image shows four terminal windows arranged in a 2x2 grid, illustrating ROS communication between a talker and a listener.

- Top Left Terminal:** Displays the output of the `roscore` command. It shows the master node starting up, setting its `/run_id`, and launching the `rosout` service.
- Top Right Terminal:** Displays the output of the `talker` node. It prints "hello world" followed by a timestamp and sequence number (359 to 366) at regular intervals.
- Bottom Left Terminal:** Displays the output of the `listener` node. It prints "I heard: [hello world 363]" and continues to receive messages from the talker.
- Bottom Right Terminal:** An empty terminal window with the prompt `robot@robot-pc:~$`.

see
TIME: [1285832303.341520000]: I heard: [hello world
362]
[INFO] [1285832303.341520000]: I heard: [hello world
363]

roscpp tutorials: talker & listener

[Code explanation]

- \$ roscore

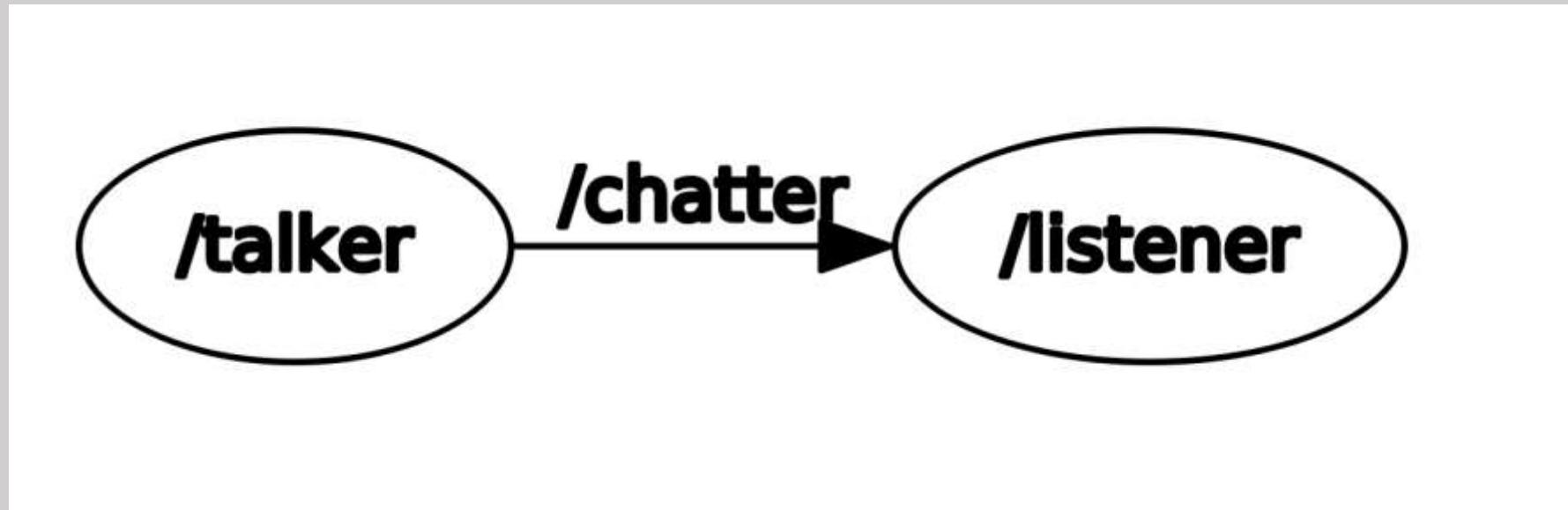
[Publisher]

- \$ rosrun roscpp_tutorials talker

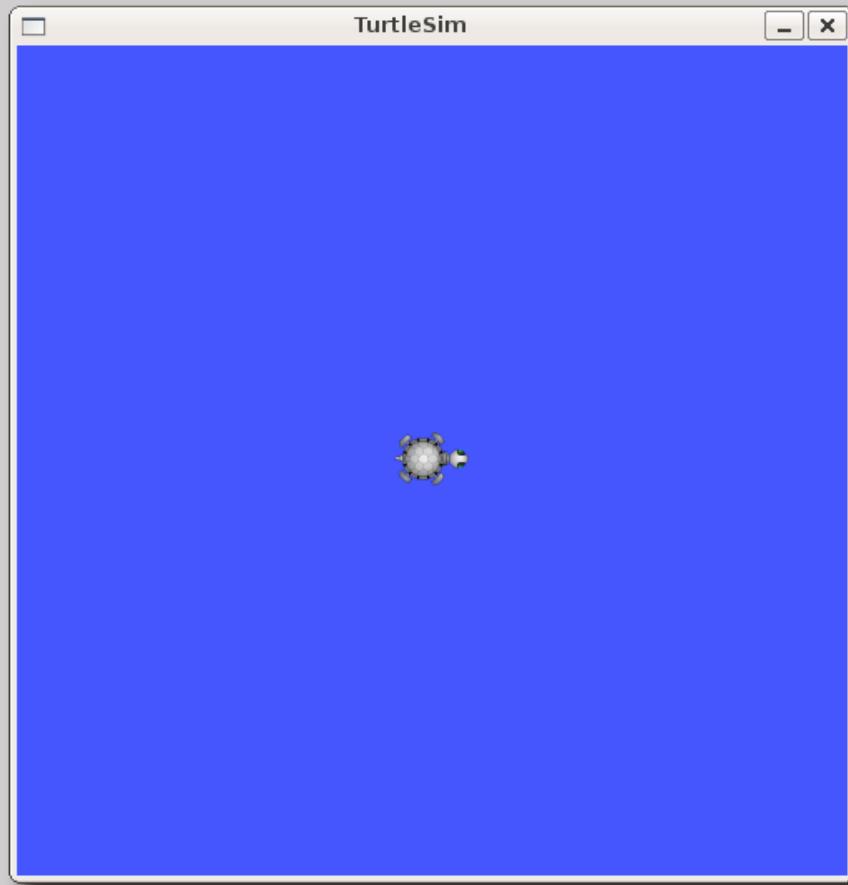
[Subscriber]

- \$ rosrun roscpp_tutorials listener

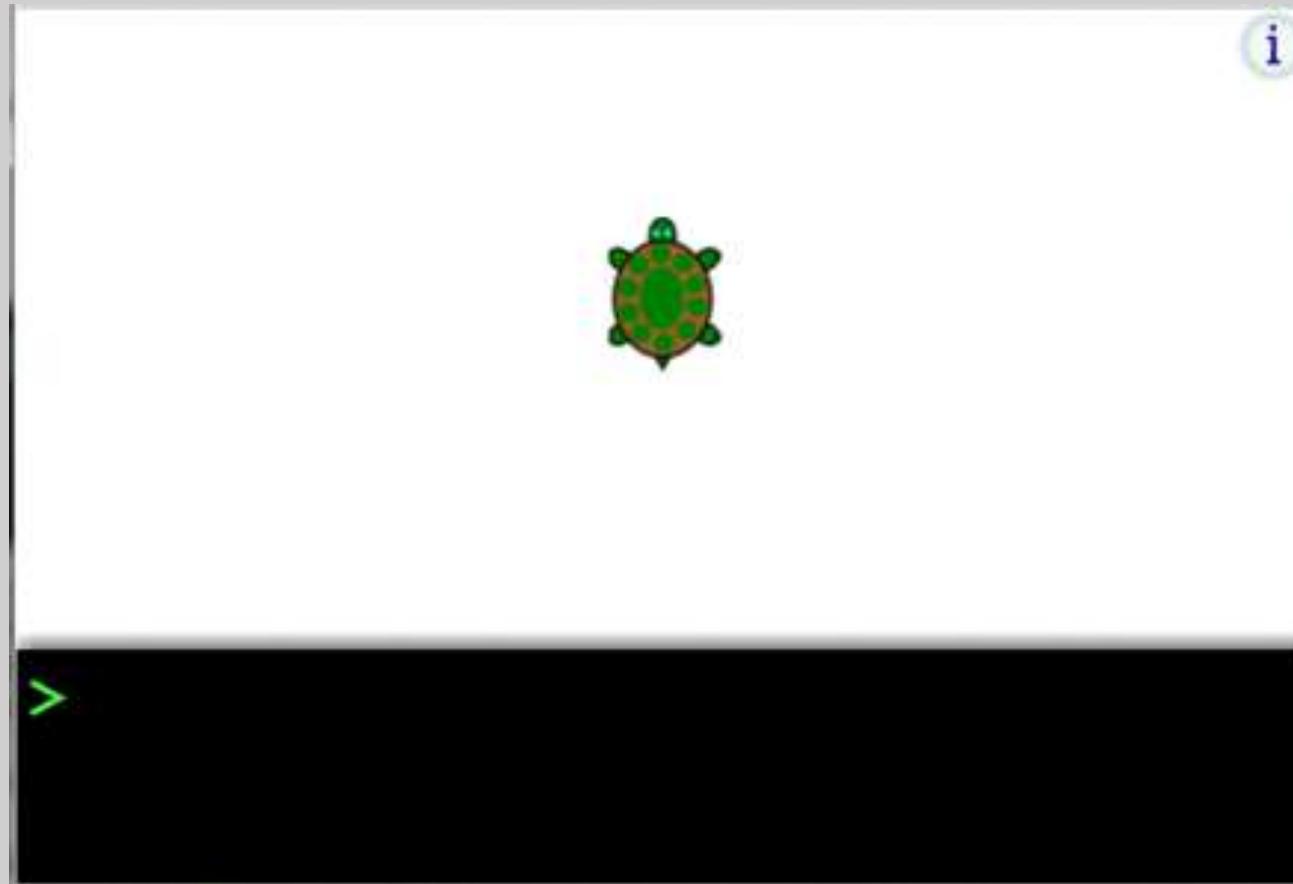
roscpp tutorials: talker & listener



Turtlesim Demo

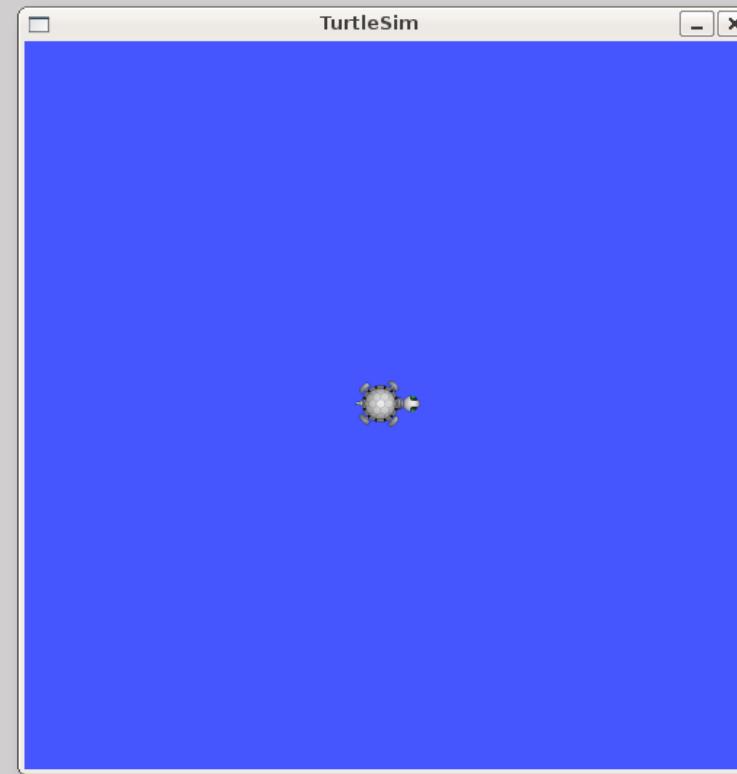


Do you remember Logo
Programming?



Learn ROS using Turtlesim

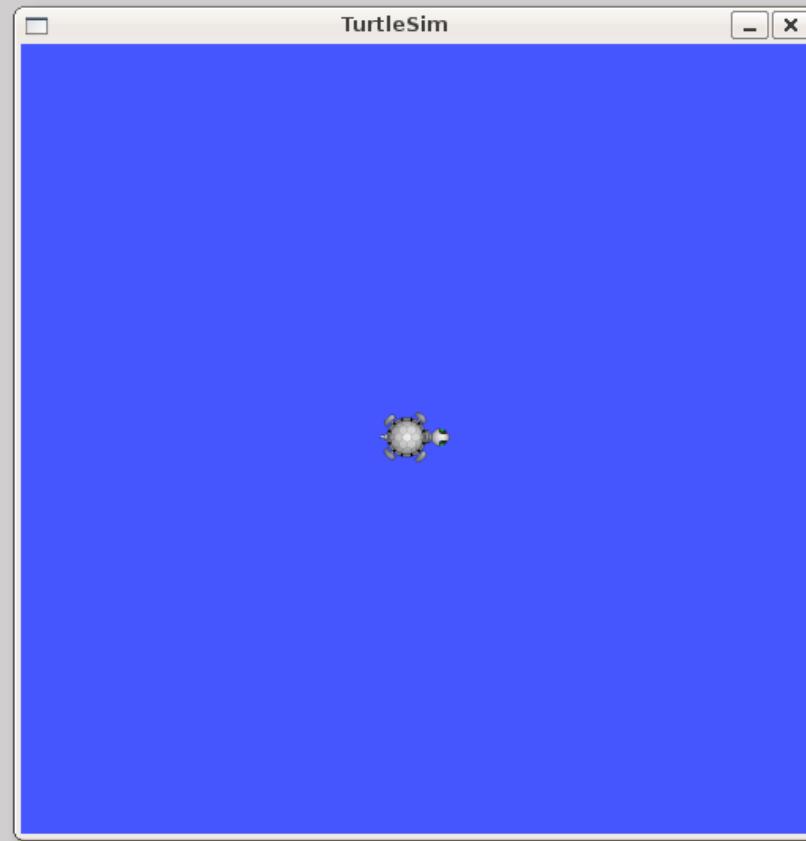
- <http://wiki.ros.org/turtlesim>



Launching turtlesim

- `$ roscore` [In first terminal]
- `$ rosrun turtlesim turtlesim_node` [In second terminal]

Launching turtlesim



Moving turtle

- `$ rosrun turtlesim turtle_teleop_key`

Moving turtle



Inspecting ROS Topic

```
robot@robot-pc:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
robot@robot-pc:~$ rostopic echo /turtle1/cmd_vel
^Crobot@robot-pc:~$ rostopic echo /turtle1/cmd_vel
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
--
```

Inspecting ROS Parameter

```
robot@robot-pc:~$ rosparam list
/background_b
/background_g
/background_r
/rosdistro
/roslaunch_uris/host_robot_pc_45153
/rosversion
/run_id
robot@robot-pc:~$ rosparam get /background_b
255
robot@robot-pc:~$
```

```
robot@robot-pc:~$ 522
```

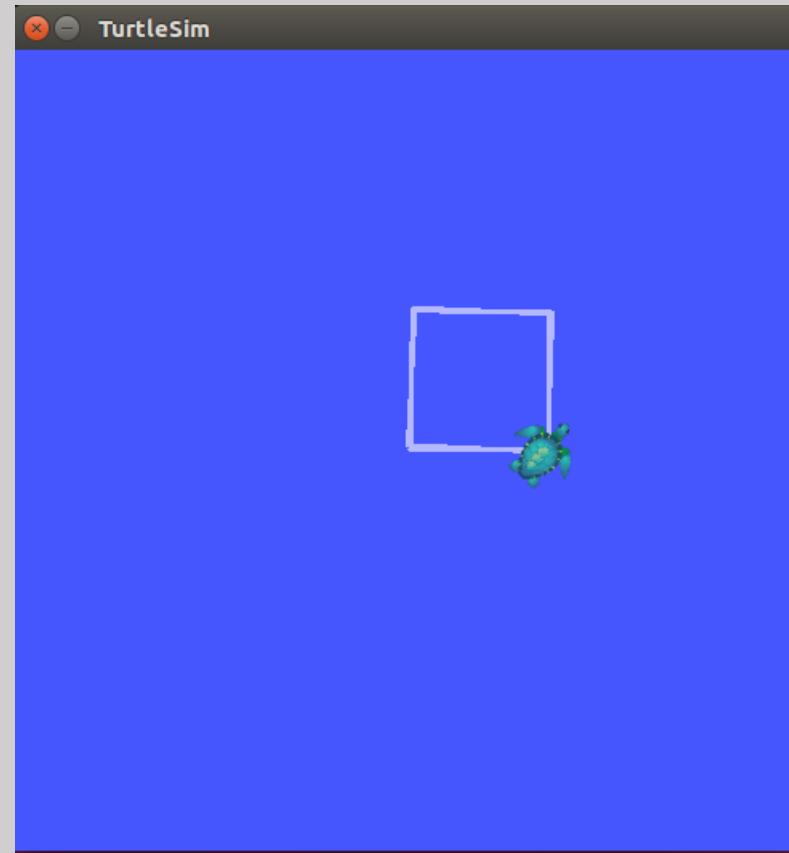
Communication graph



Turtlesim Demo: Draw Square

- `$ roscore` [In first terminal]
- `$ rosrun turtlesim turtlesim_node` [In second terminal]
- `rosrun turtlesim draw_square`

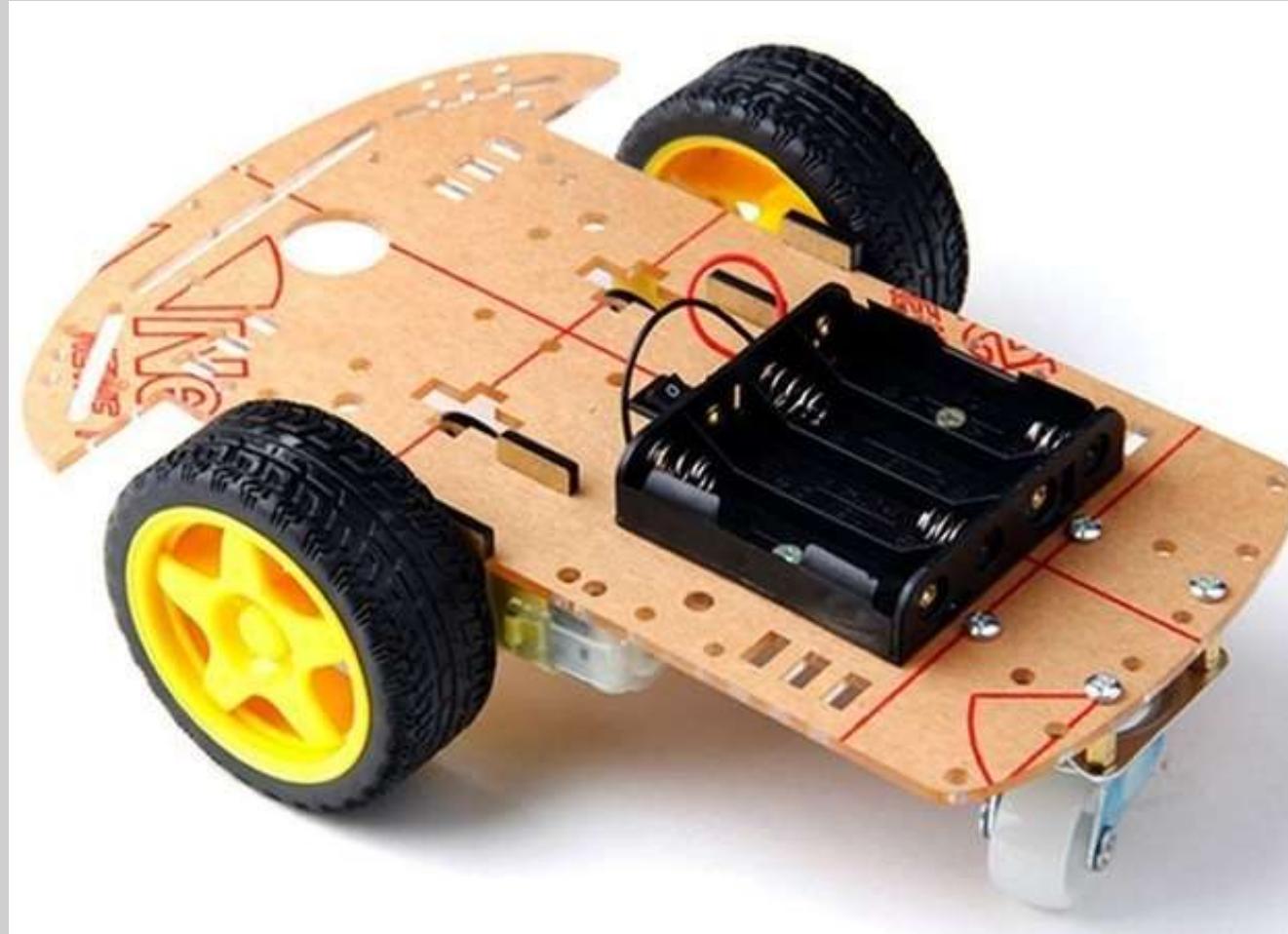
Turtlesim Demo: Draw Square



Working of ROS: Tools

- Rviz (ROS Visualizer)
 - 3D Visualization tool in ROS
 - <http://wiki.ros.org/rviz>
- Rqt (ROS Qt)
 - GUI framework in ROS based on Qt
 - <http://wiki.ros.org/rqt>
- ROS Command line tools
 - rostopic
 - roslaunch
 - <http://wiki.ros.org/ROS/CommandLineTools>

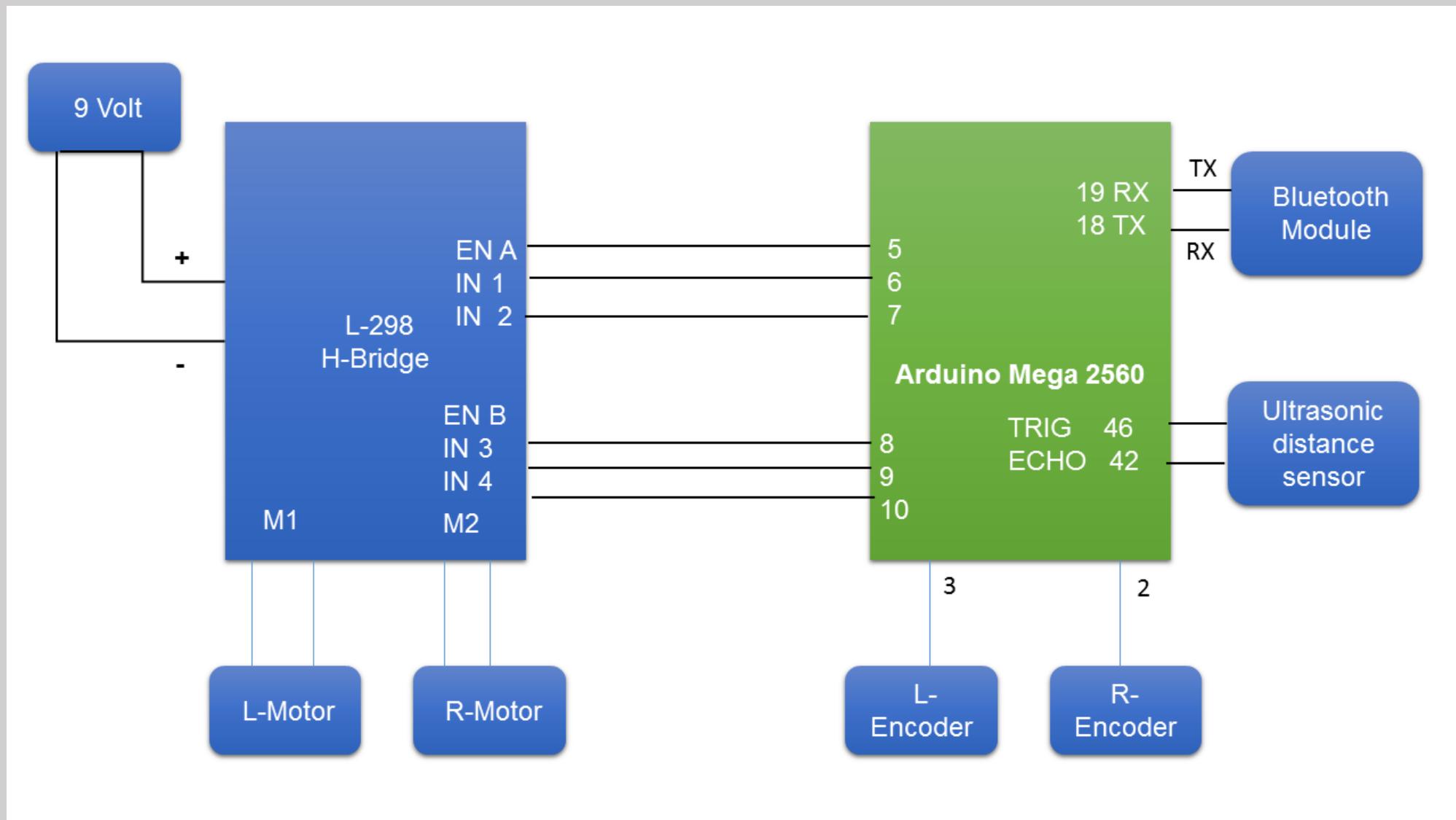
Interfacing Mobile robot to ROS



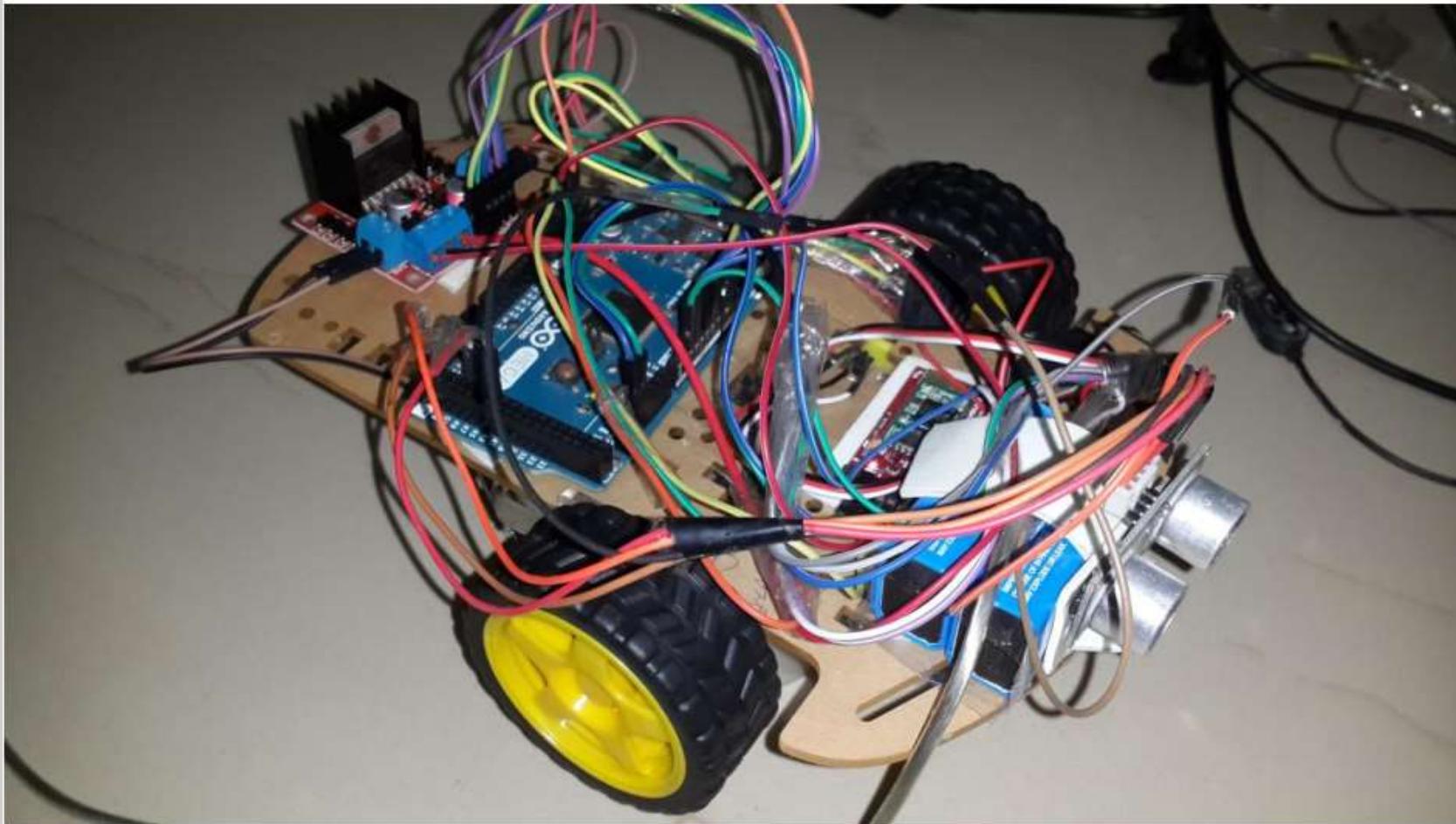
Interfacing Mobile robot to ROS



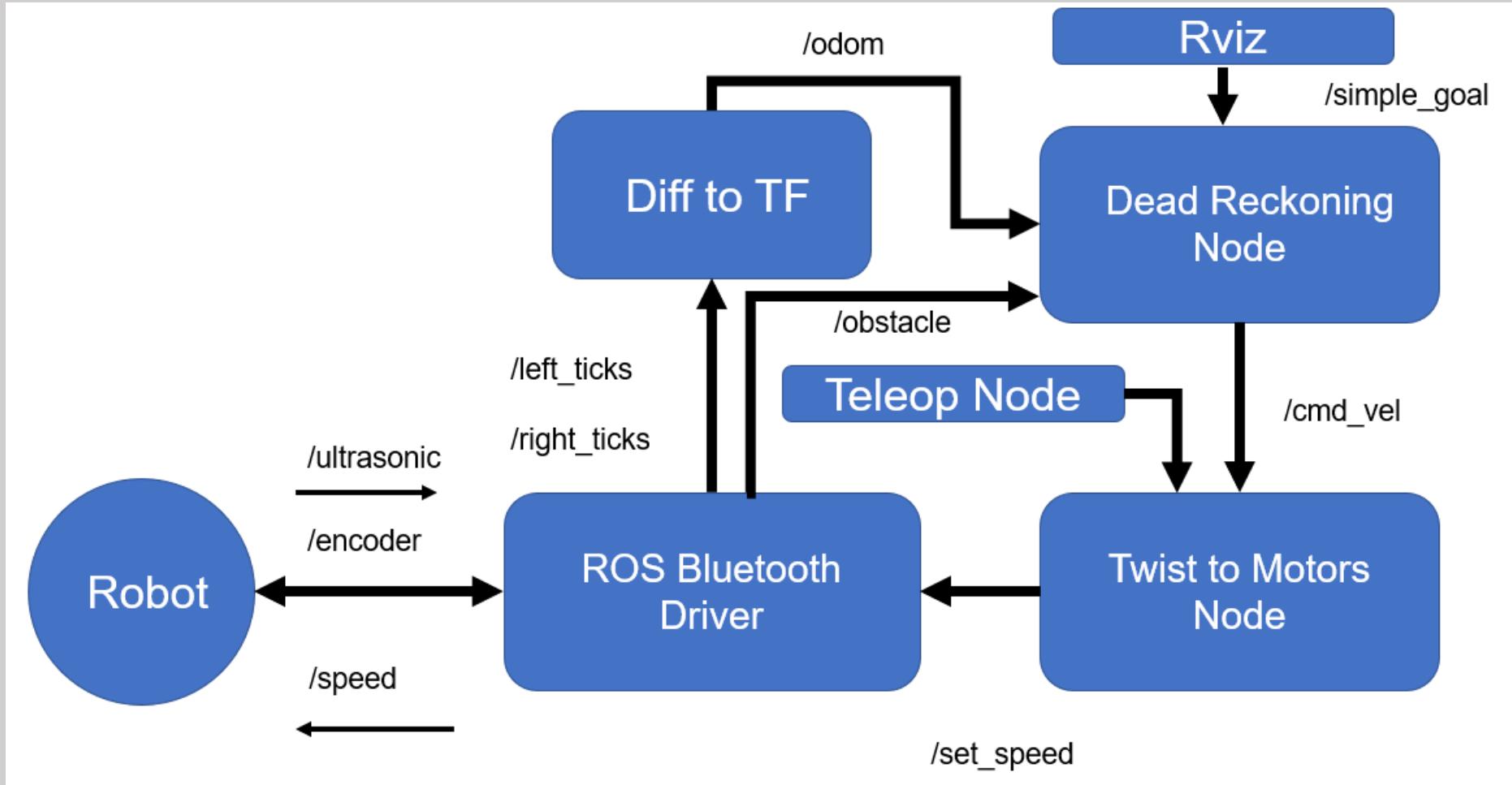
Interfacing Mobile robot to ROS



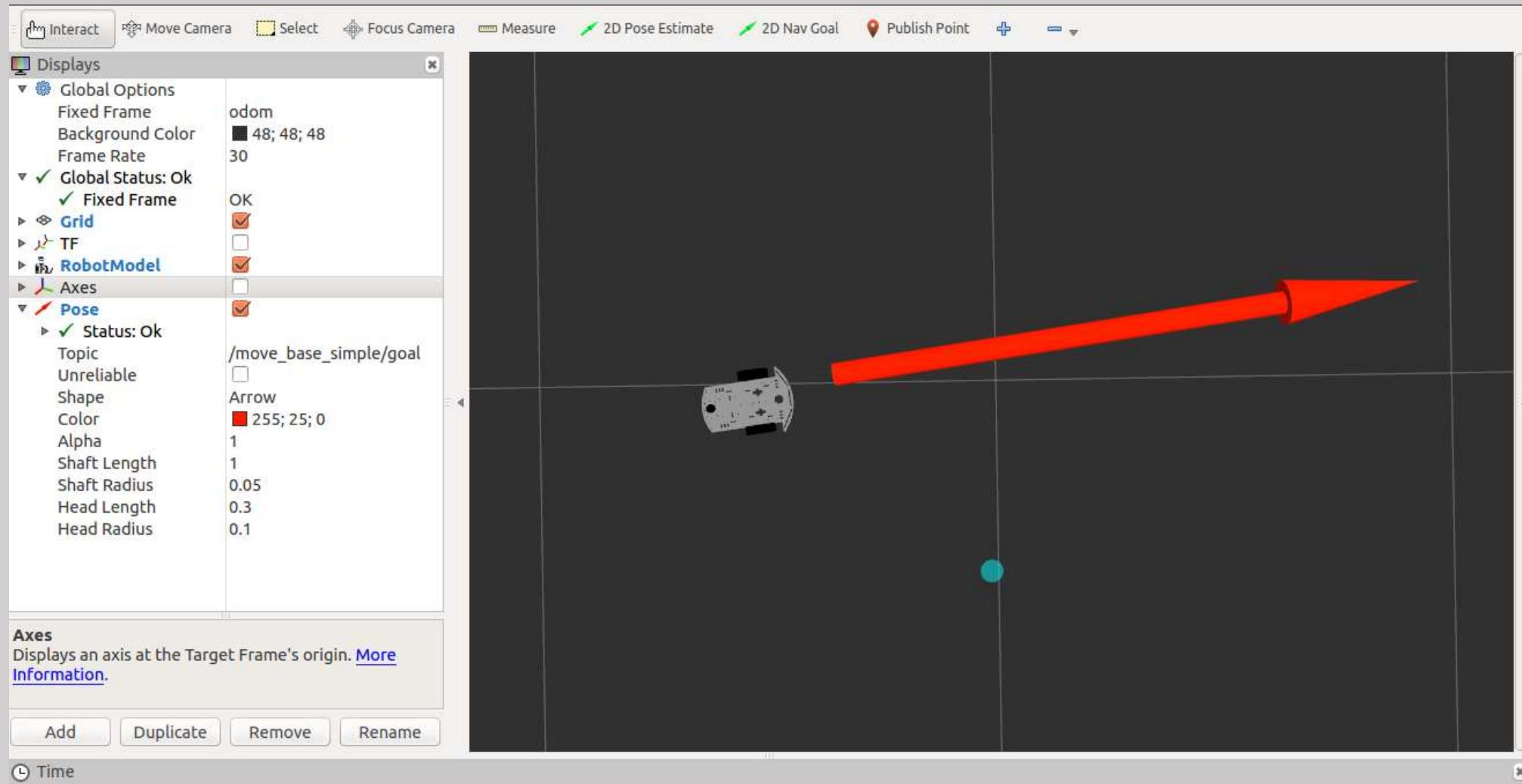
Interfacing Mobile robot to ROS



Interfacing Mobile robot to ROS

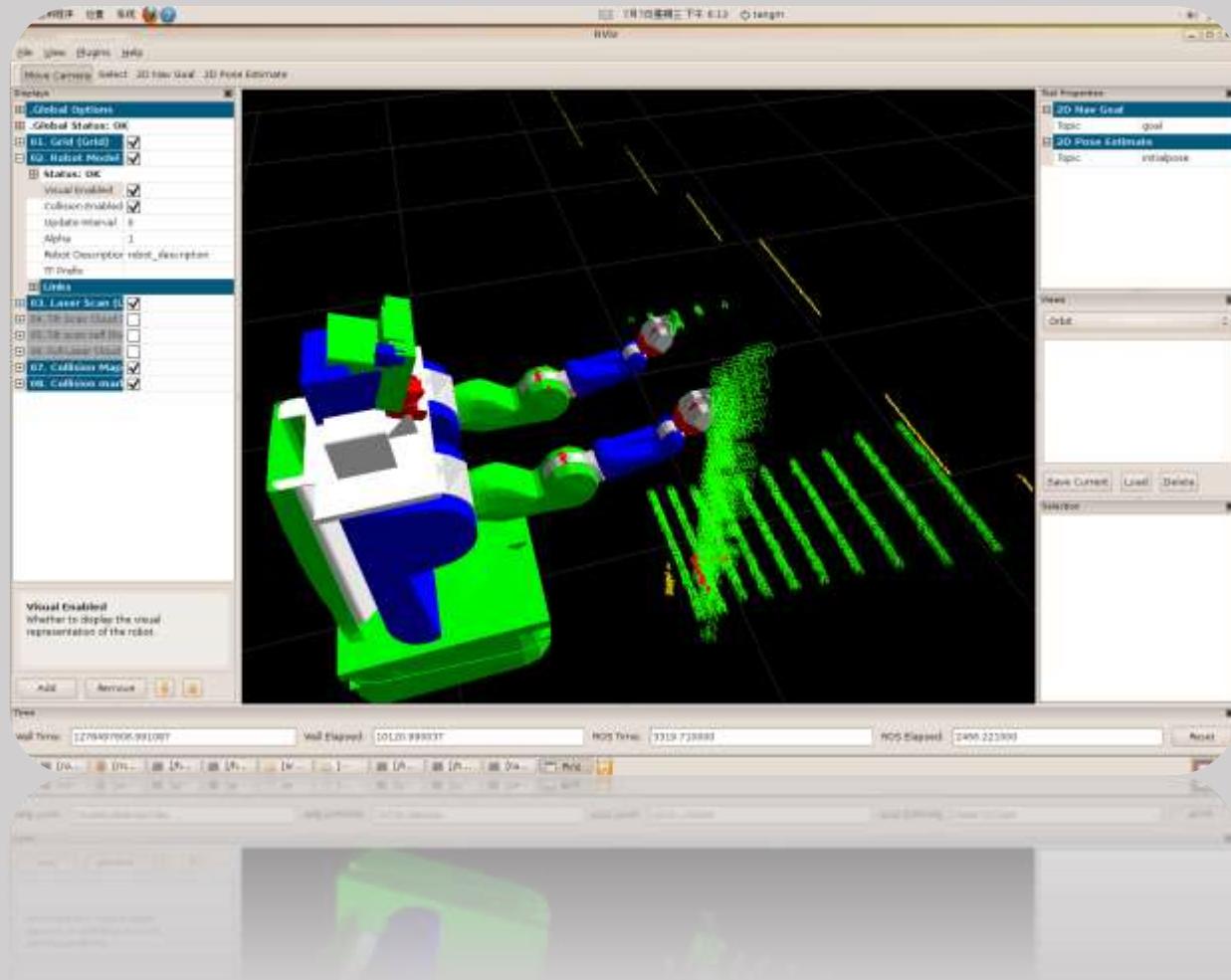


Interfacing Mobile robot to ROS

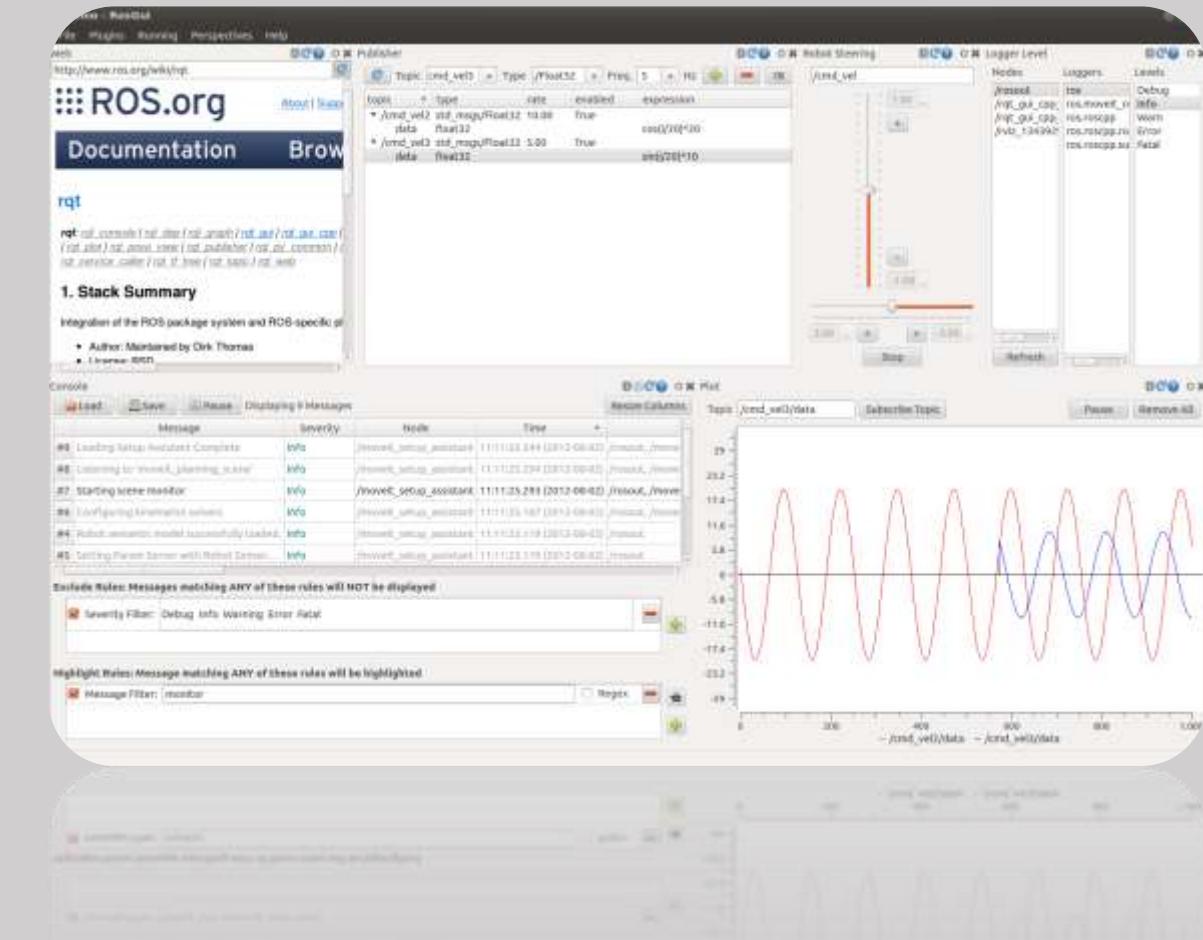


Working of ROS: Tools

Rviz



rqt

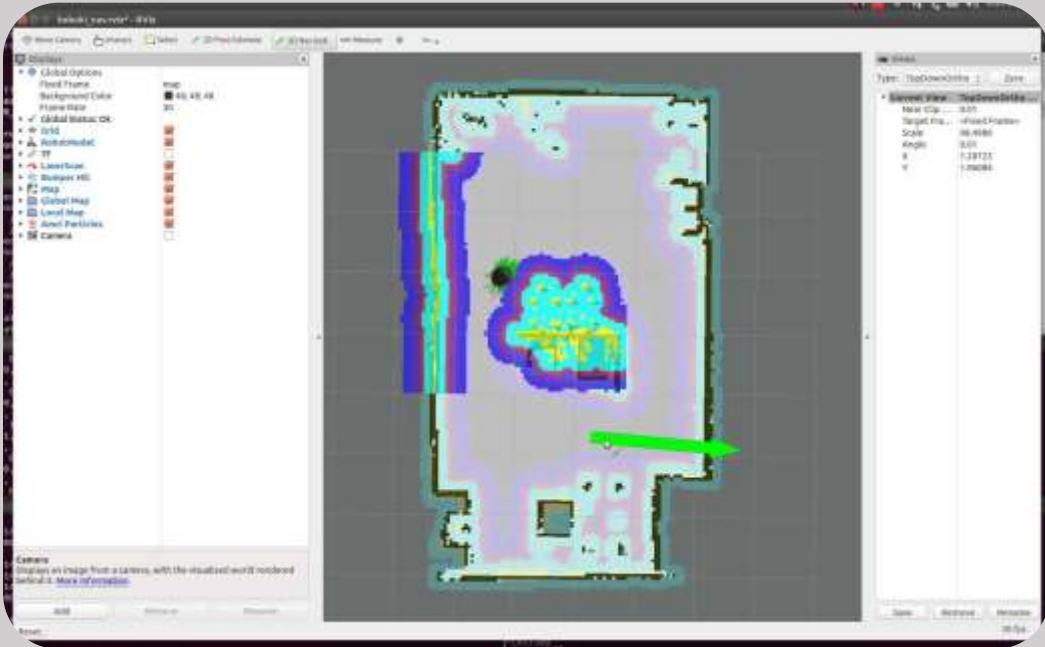


Working of ROS: Capabilities

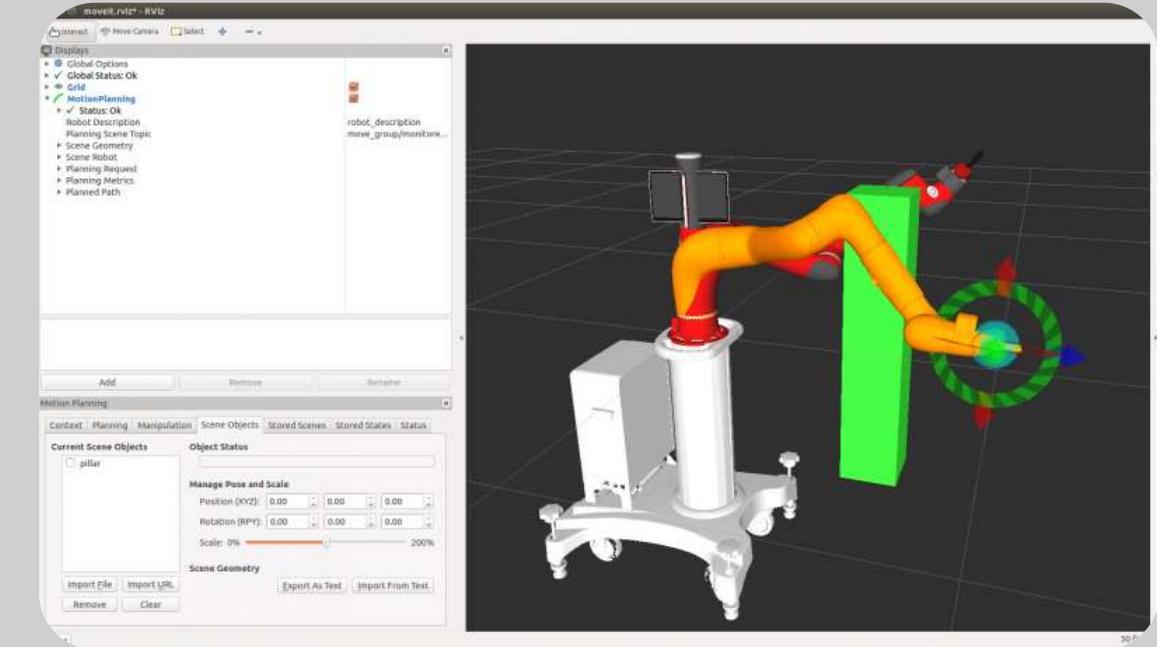
- ROS Navigation
 - Collection of software in ROS to move a mobile robot from A to B
- ROS MoveIt!
 - Software in ROS for planning the motion of a robotic arm

Working of ROS: Capabilities

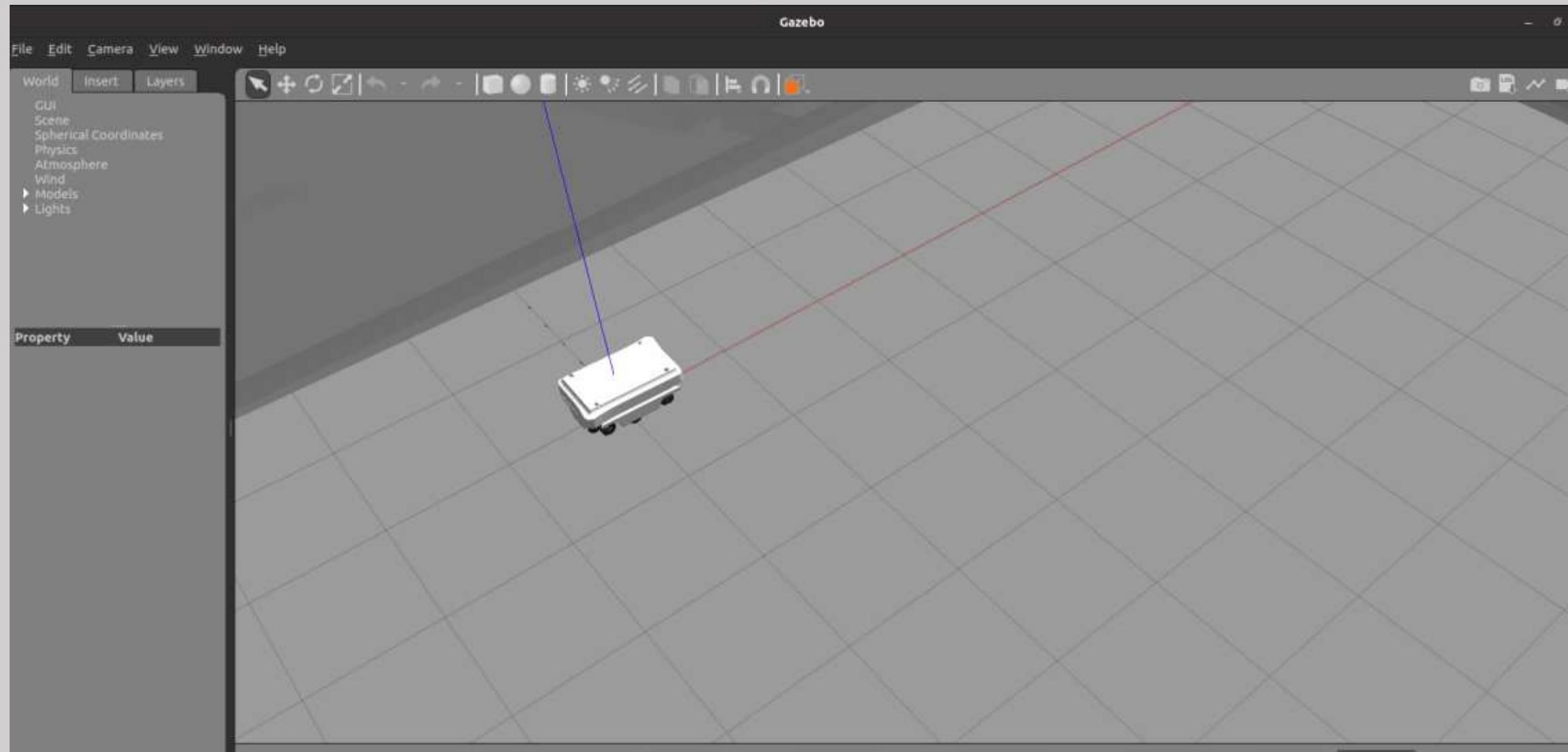
ROS Navigation



ROS MoveIt!



ROS Navigation



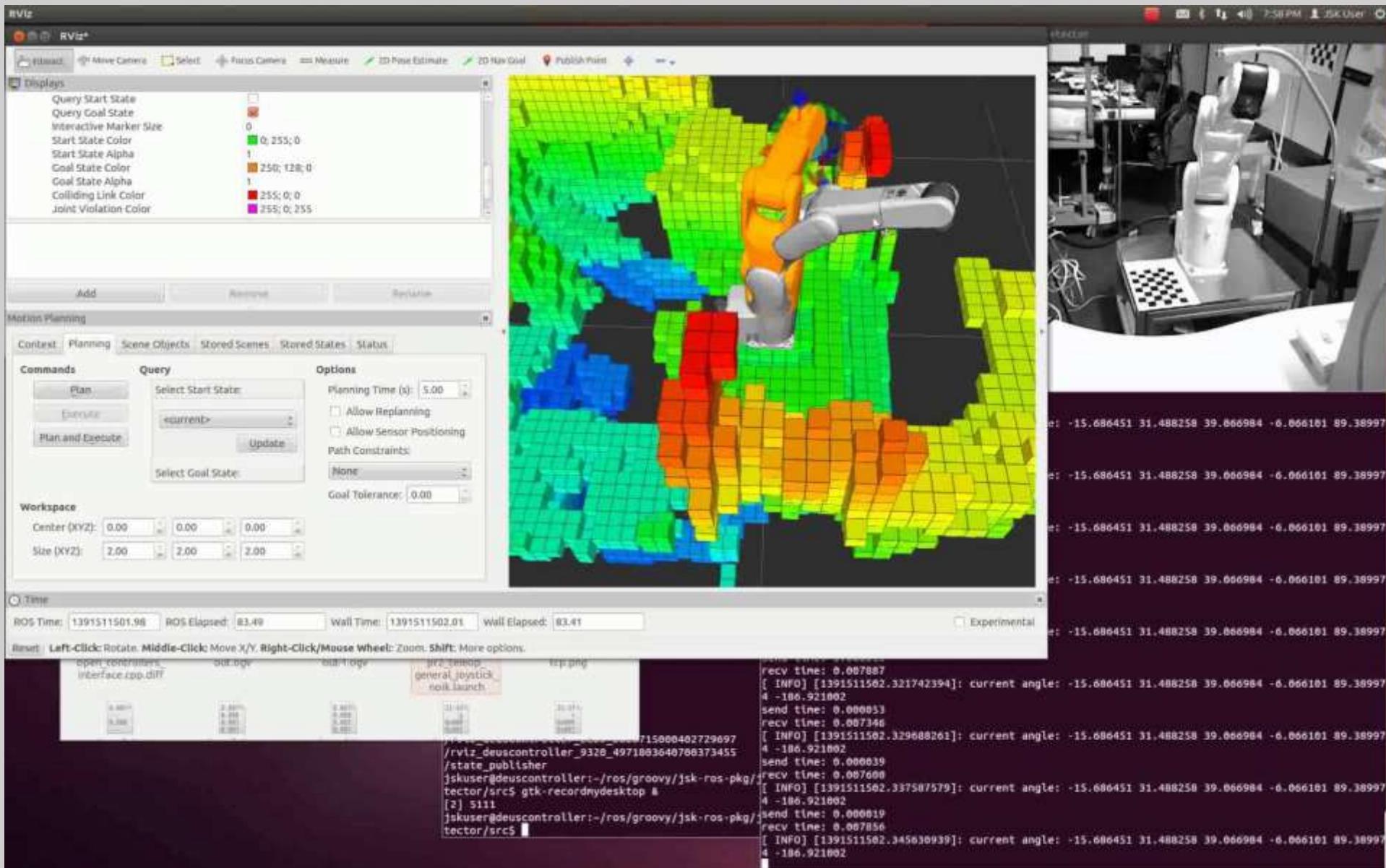
ROS Manipulation



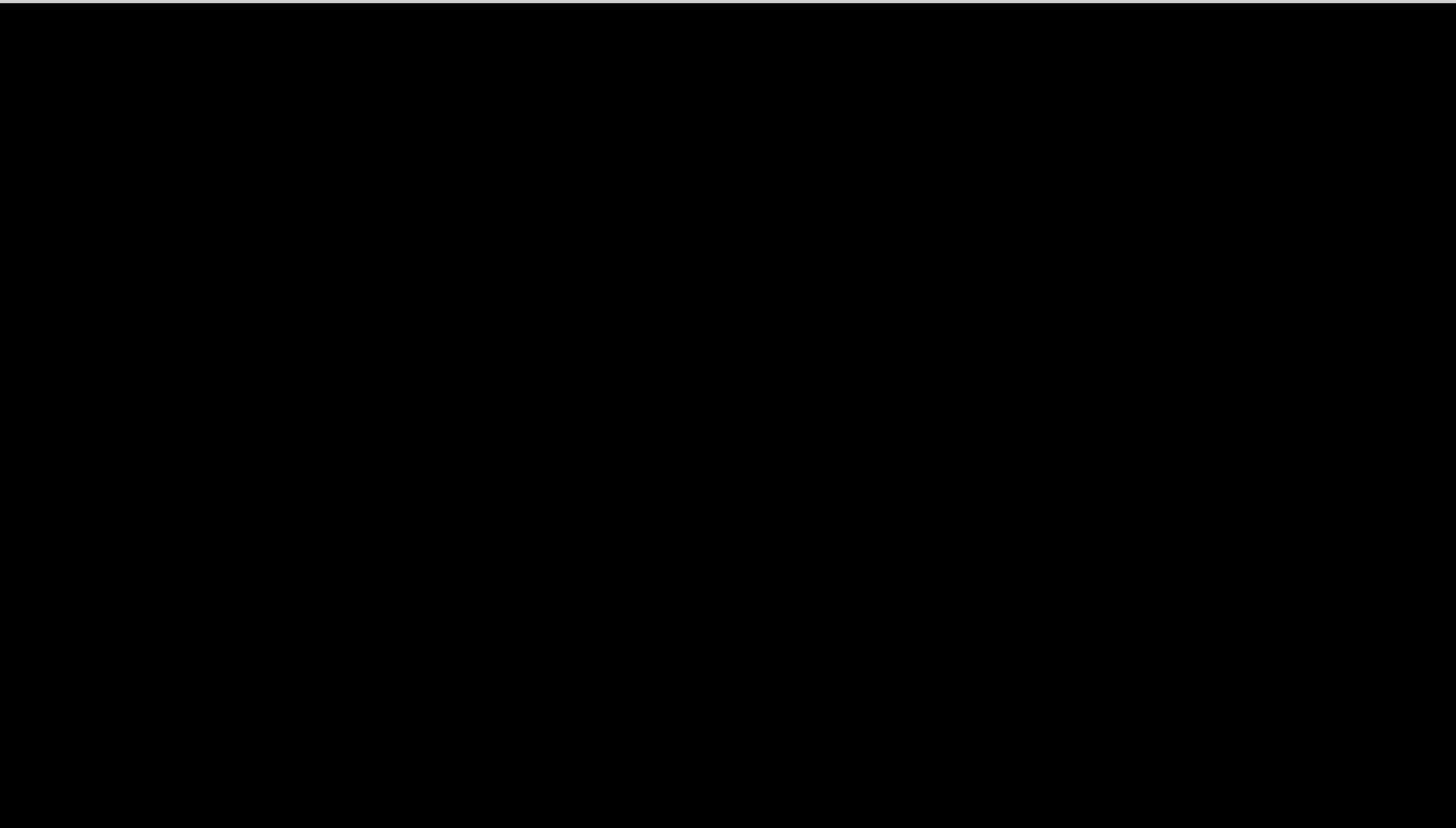
ROS Manipulation



ROS Perception



ROS Perception



Working of ROS: Ecosystem

- Worldwide ROS developers
 - <http://wiki.ros.org/Metrics>

| | | |
|-----|--|-------------------------|
| 1. |  United States | 100,711 (20.08%) |
| 2. |  China | 90,120 (17.97%) |
| 3. |  Japan | 45,834 (9.14%) |
| 4. |  Germany | 39,590 (7.89%) |
| 5. |  India | 20,632 (4.11%) |
| 6. |  South Korea | 16,683 (3.33%) |
| 7. |  United Kingdom | 12,784 (2.55%) |
| 8. |  Taiwan | 11,809 (2.35%) |
| 9. |  Canada | 11,685 (2.33%) |
| 10. |  France | 11,651 (2.32%) |



Working of ROS: Ecosystem

- ROS Wiki: <http://wiki.ros.org>
- ROS Q&A: <https://answers.ros.org/questions/>

| | | |
|-----|--|-------------------------|
| 1. |  United States | 100,711 (20.08%) |
| 2. |  China | 90,120 (17.97%) |
| 3. |  Japan | 45,834 (9.14%) |
| 4. |  Germany | 39,590 (7.89%) |
| 5. |  India | 20,632 (4.11%) |
| 6. |  South Korea | 16,683 (3.33%) |
| 7. |  United Kingdom | 12,784 (2.55%) |
| 8. |  Taiwan | 11,809 (2.35%) |
| 9. |  Canada | 11,685 (2.33%) |
| 10. |  France | 11,651 (2.32%) |

| | | |
|-----|--|----------------|
| 10. |  EUROPE | 11,921 (2.00%) |
| 9. |  CHINA | 11,982 (2.00%) |



Source: Google Analytics
Site: wiki.ros.org in July 2017

Supported Host Operating System

- Full Support
 - Ubuntu/Linux
 - Debian



- Experimental
 - OSX
 - Gentoo
 - Windows





Who created ROS?

Discussing the brief history of ROS

Who started ROS?

- 2007 : ROS project started under the name *switchyard* by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot STAIR (STanford AI Robot).
- Original Author : Morgan Quigley
- Main publication: ROS: an open-source Robot Operating System

ROS: an open-source Robot Operating System

Morgan Quigley*, Brian Gerkey†, Ken Conley†, Josh Faust†, Tully Foote†,
Jeremy Leibs‡, Eric Berger†, Rob Wheeler†, Andrew Ng*

*Computer Science Department, Stanford University, Stanford, CA

†Willow Garage, Menlo Park, CA

‡Computer Science Department, University of Southern California

Who started ROS?

- Morgan Quigley
- Currently working at Open Robotics
(<https://www.openrobotics.org/>)
- <http://people.osrfoundation.org/morgan/>
- PhD under Prof: Andrew Ng
 - Stanford Artificial Intelligence Lab



Who created ROS?

- 2008 – 2013 : Willow Garage, a robotics research institute/incubator, USA
- Started ROS project
- Significant development of ROS happened here.
- <https://www.willowgarage.com/>



Who created ROS?

- PR2: The robot fully running using ROS
- Turtlebot: Education & Research robot using ROS



Who created ROS?

- 2013 – Now : Open Source Robotics Foundation or Open Robotics
- <https://osrfoundation.org>
- <https://www.openrobotics.org/>



Open Source
Robotics Foundation



Who created ROS?

- Create and support Open software and hardware
- Research, Education and Product development
- Main products: ROS & Gazebo simulator



Open Source
Robotics Foundation

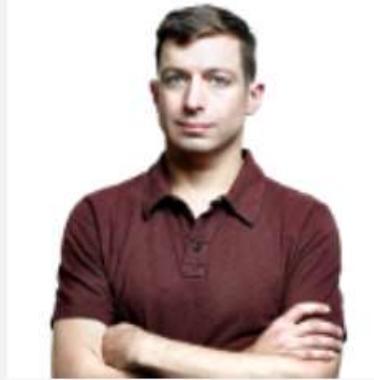


Team: Open Robotics



Morgan Quigley

Chief Architect,
Founder



Brian Gerkey

Chief Executive
Officer, Founder



Nathan Koenig

Chief Technology
Officer, Founder



Tully Foote

ROS Platform
Manager

<https://www.openrobotics.org/team/>



Why we use ROS for Robotics Programming?

Discussing the main features of ROS and the reason for using
ROS in robotics research and companies

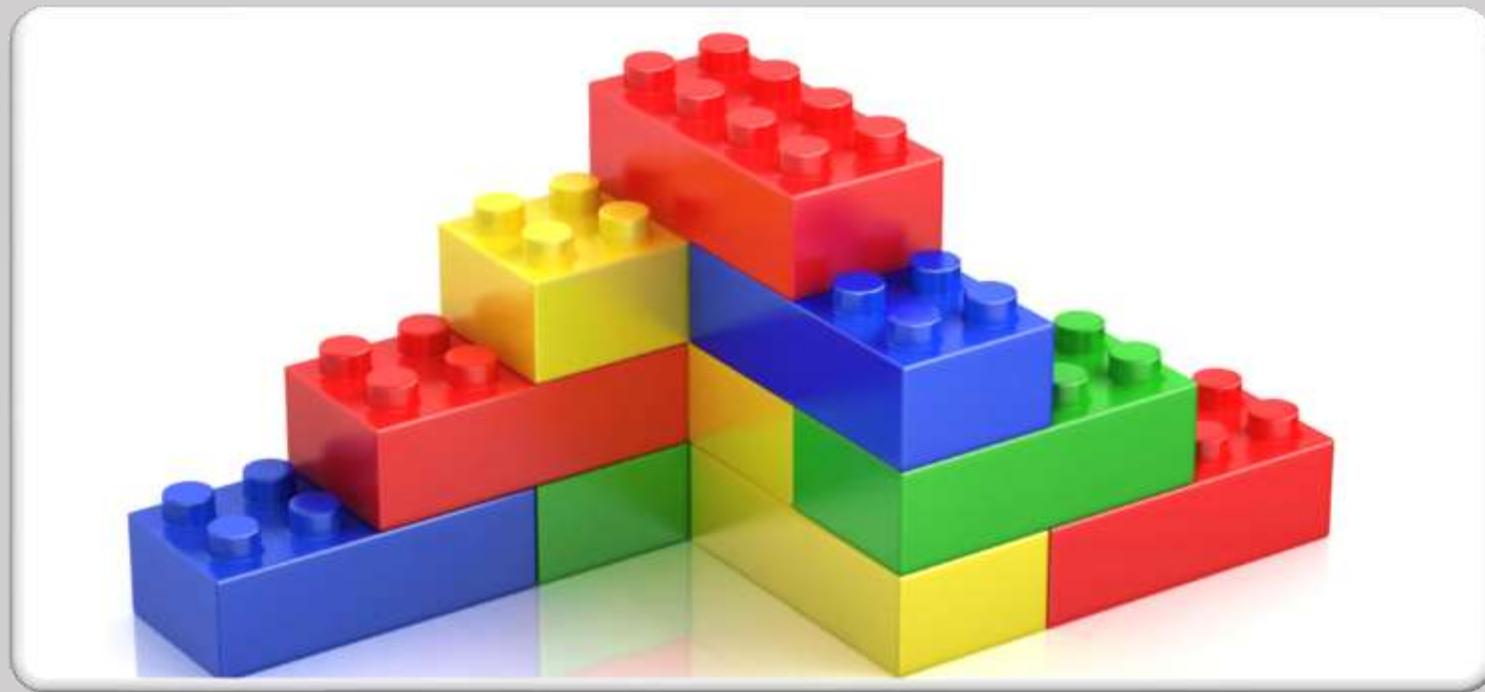
Why we use ROS

- Code reuse in Robotics research and development



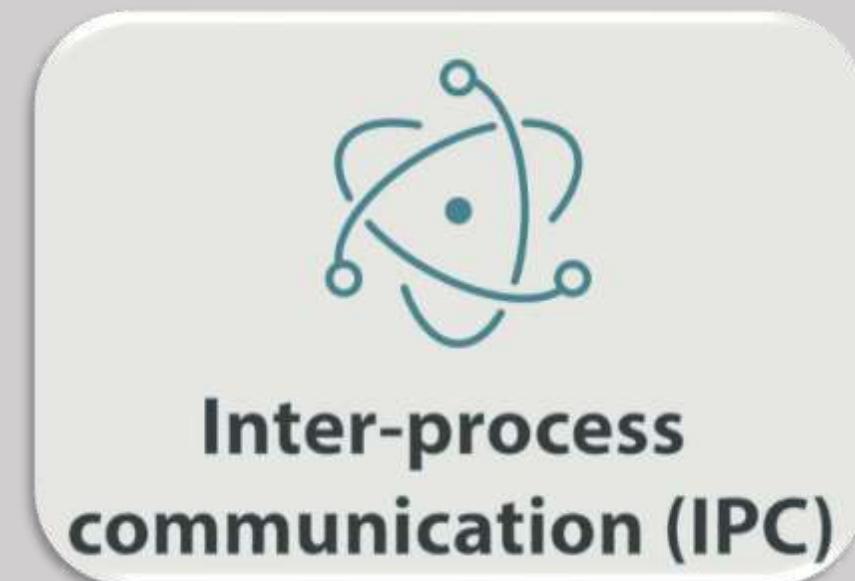
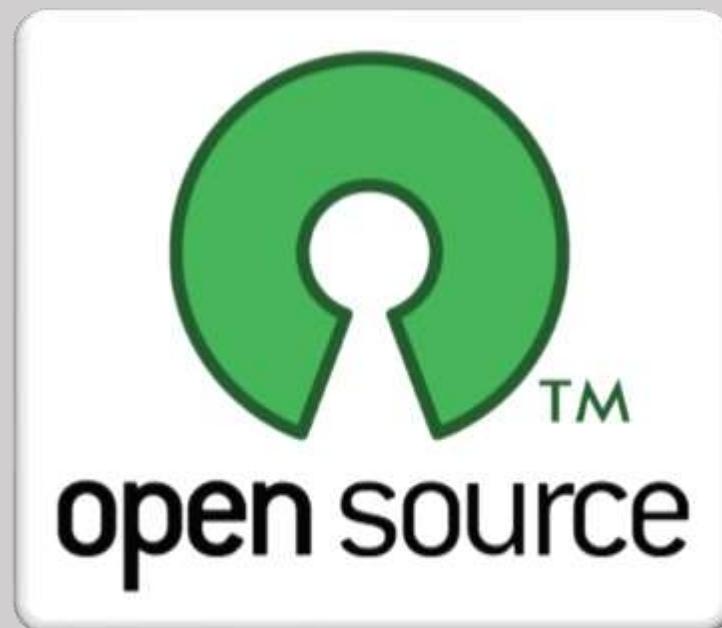
Why we use ROS

- Modularity of code: [Nodes->Packages->Meta Packages-> Repositories]



Why we use ROS

- Free & Open Source framework to implement inter process communication



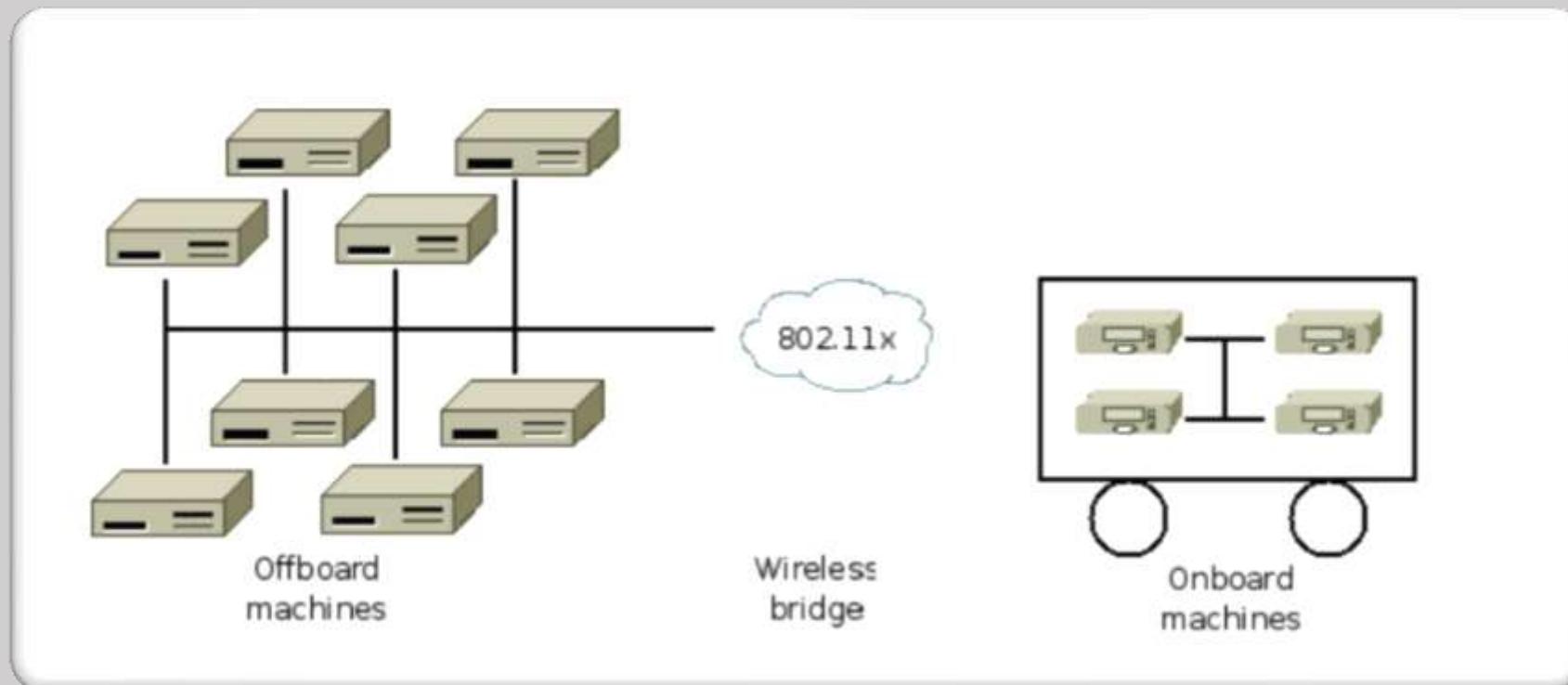
Why we use ROS

- Great community support



Why we use ROS

- Scalable framework, Distributed communication architecture



Why we use ROS

- Popular programming language support for development: C++, Python
- Can create Agnostic libraries
- Easy test interface
- Various tools to visualize and debug robot data



What is ROS Distribution

Discussing important concepts of ROS distributions

What is a ROS Distribution?

- Versioned set of ROS programs
- Similar to Linux Distribution: Ubuntu, Fedora, Kali Linux
- Relatively Stable set of ROS programs/packages
- Easy to maintain



List of ROS Distributions

- 2010 – ROS Box Turtle : First ROS distribution
- Ubuntu support: 8.04,9.04,9.10 & 10.04
- Status: EOL (End of Life)



Box Turtle

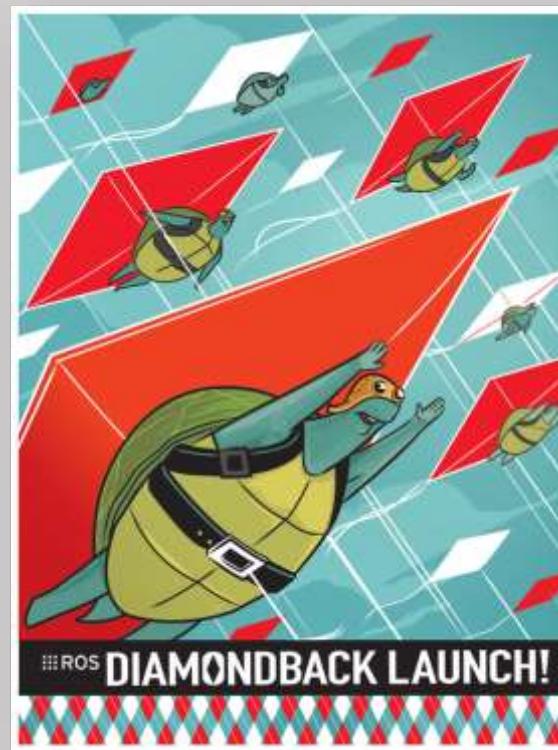
List of ROS Distributions

- 2010 – ROS C Turtle : Second ROS distribution
- Ubuntu support: 9.04,9.10,10.04 & 10.10
- Status: EOL



List of ROS Distributions

- 2011 – ROS Diamondback : Third ROS distribution
- Ubuntu support: 10.04,10.10 & 11.04
- Status: EOL



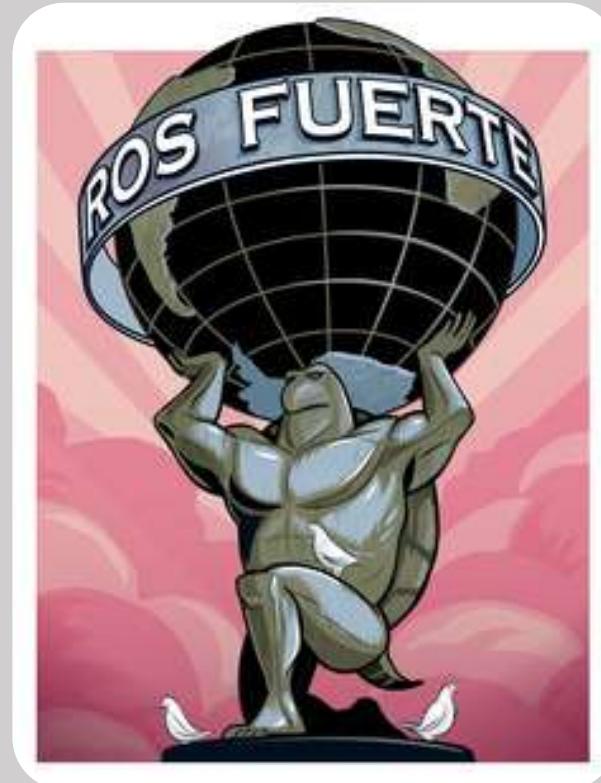
List of ROS Distributions

- 2011 – ROS Electric Emys: Fourth ROS distribution
- Ubuntu support: 10.04, 10.10, 11.04 & 11.10
- Status: EOL



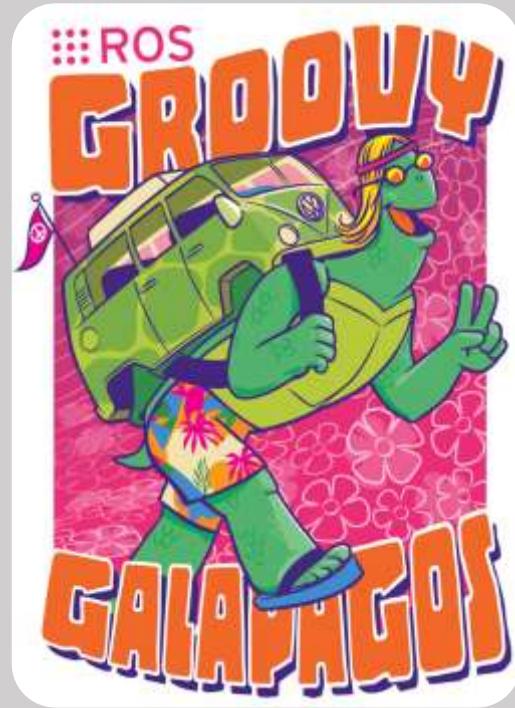
List of ROS Distributions

- 2012 – ROS Fuerte: Fifth ROS distribution
- Ubuntu support: 10.04, 10.10 & 12.04
- Status: EOL



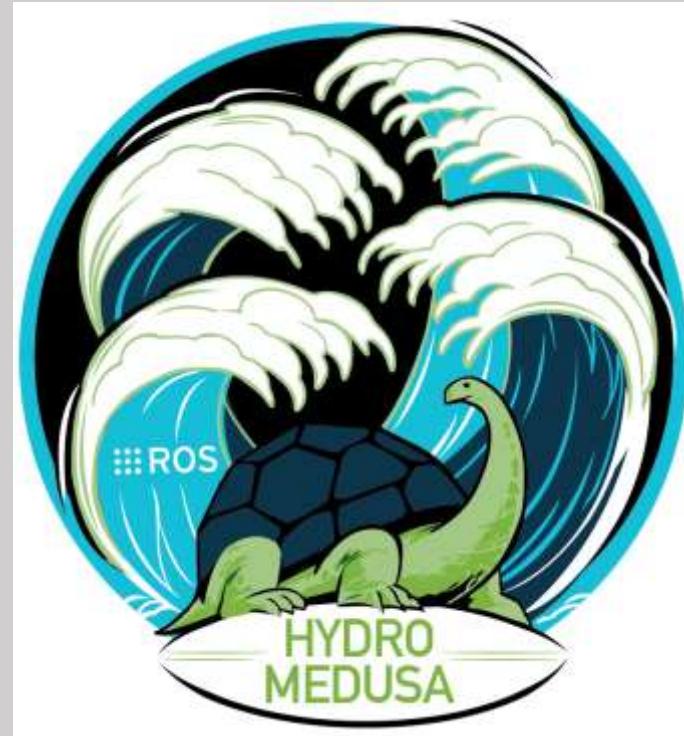
List of ROS Distributions

- 2013 – ROS Groovy Galapagos: Sixth ROS distribution
- Ubuntu support: 11.10, 12.04 & 12.10
- Status: EOL, July 2014



List of ROS Distributions

- 2013 – ROS Hydro Medusa: Seventh ROS distribution
- Ubuntu support: 12.04, 12.10 & 13.04
- Status: EOL, May 2015



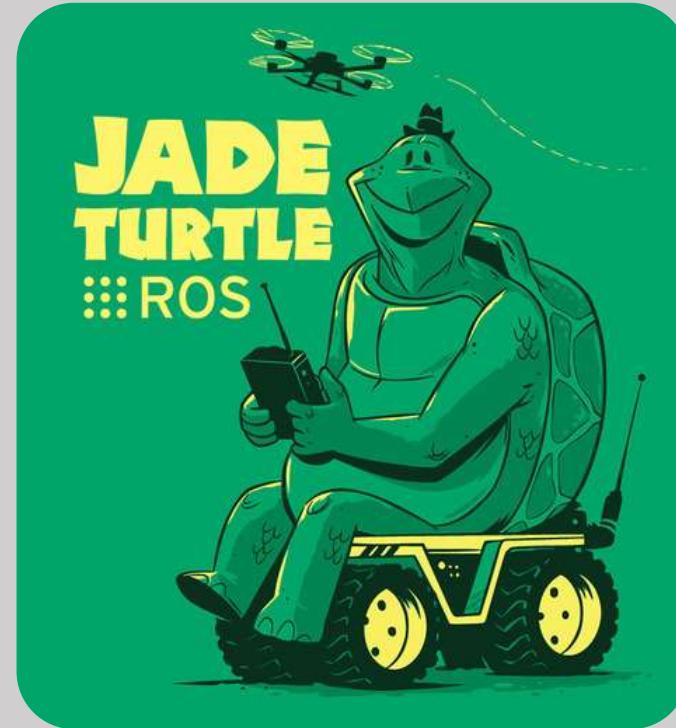
List of ROS Distributions

- 2014 – ROS Indigo Igloo: Eighth ROS distribution
- Ubuntu support: 13.10 & 14.04
- Status: April, 2019



List of ROS Distributions

- 2015 – ROS Jade Turtle: Ninth ROS distribution
- Ubuntu support: 14.04, 14.10, & 15.04
- Status: May, 2017



List of ROS Distributions

- 2016 – ROS Kinetic Kame: Tenth ROS distribution
- Ubuntu support: 15.10 & 16.04
- Status: April, 2021 , Recommended



List of ROS Distributions

- 2017 – ROS Lunar Loggerhead: Eleventh ROS distribution
- Ubuntu support: 16.04, 16.10 & 17.04
- Status: May, 2019 (EOL)



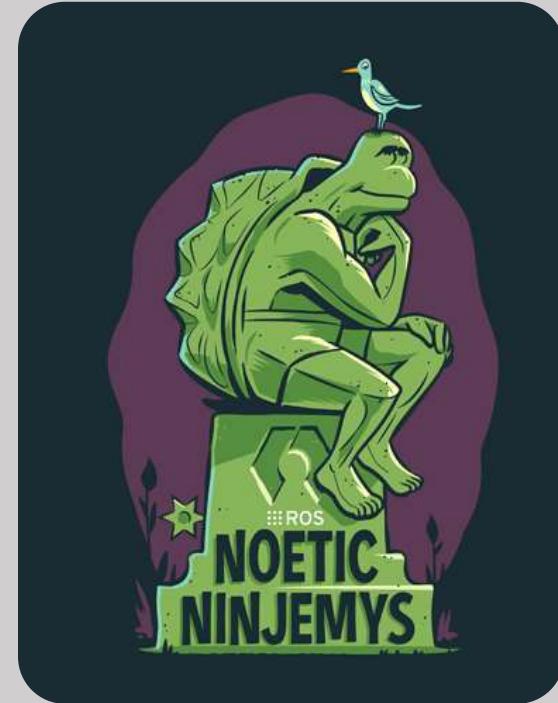
List of ROS Distributions

- 2018 – ROS Melodic Morenia: Twelfth ROS distribution
- Ubuntu support: 17.10 & 18.04
- Status: May, 2023 (EOL)



List of ROS Distributions

- 2018 – ROS Noetic Ninjemys: Thirteenth ROS distribution
- Ubuntu support: 20.04
- Status: May, 2025 (EOL), Latest
- List of ROS1 distributions
- <http://wiki.ros.org/Distributions>



List of ROS 2 Distributions

- 2020 – ROS 2 Foxy Fitzroy
 - Ubuntu support: 20.04
 - Status: May, 2023 (EOL), Latest
-
- List of ROS 2 distributions
 - <https://index.ros.org/doc/ros2/Releases/>





Which are the robots supporting ROS?

Discussing list of robots working using ROS framework

ROS supported Robots

- The Robots which are programmed using ROS
- Complete or partial ROS interface
- Research and commercial robots
- Open-Source reusable code and open hardware design
- List of ROS supported robots: <https://robots.ros.org/>



ROS Supported Robots

- PR2(Personal Robot)
- <https://www.willowgarage.com/pages/pr2/overview>
- Processor : 2 X Quad Core i7 Xeon
- RAM : 24 GB
- Hard disk : 2 TB
- 2 x 7DOF arm
- Sensors : Laser scanners, Kinect



ROS Supported Robots

- Turtlebot 2 : Mobile Robot
 - <http://www.turtlebot.com/>
-
- Roomba Base
 - Asus Xtion Pro Live
 - Netbook loaded with ROS
 - Educational and Research



ROS Supported Robots



Baxter
Collaborative Robot
Rethink Robotics



REEM-C
Full size biped humanoid robot
PAL Robotics



Pepper
Semi-humanoid robot
Softbank Robotics



Tiago
Service Robot
PAL Robotics

ROS Supported Robots



Robonaut - 2
Robotic Astronaut
NASA



REEM
Full size humanoid robot
PAL Robotics



Fetch
Mobile Manipulation platform
Fetch Robotics



TALOS
Biped Robot
PAL Robotics

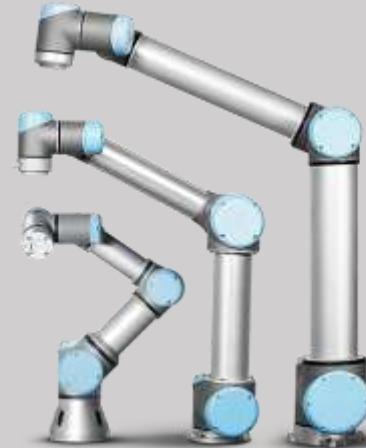
ROS Supported Robots



KINOVA-JACO
Robotic Arm
Kinova Robotics



ABB -Industrial Arm
Industrial Arm
ABB



UR3, UR5 & UR 10
Collaborative Robot Arm
Universal Robots

ROS Supported Robots



Pioneer 3DX
Differential drive robot
Omron Adept Mobile Robots



Husky
Unmanned Ground Vehicle (UGV)
Clearpath Robotics



Freight
Mobile robotics platform
Fetch Robotics

ROS Robot Demos



The ROS logo consists of a white square containing a 4x4 grid of dark blue dots on the left, followed by the letters "ROS" in a large, bold, dark blue sans-serif font.

Which are the Sensors supported by ROS??

Important sensors supported in ROS.

ROS supported Sensors

- Sensors having a ROS interface
- Sensor data can be access from all ROS programs
- Sensor data can be visualized in Rviz
- List of ROS supported robots: <http://wiki.ros.org/Sensors>





UTM - 30LX, Laser Scanner

Velodyne LiDAR



Kinect, 3D Depth Sensor Intel RealSense, 3D Depth Sensor



Vision Sensors Supported in ROS



Velodyne LiDAR



UTM - 30LX, Laser Scanner



Kinect, 3D Depth Sensor



Intel RealSense, 3D Depth Sensor

Vision Sensors Supported in ROS



ZED Camera, 3D Depth camera



USB Cam



Leap Motion, Hand tracker device



Orbbec Astra, 3D Depth camera

Important GPS + IMU Supported in ROS



Micro Strain, IMU



Applanix,
IMU + GPS

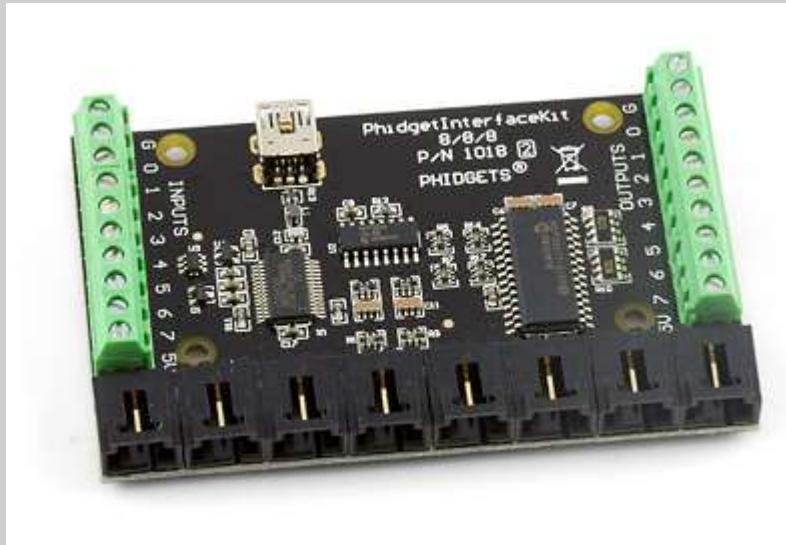


Xsens, IMU

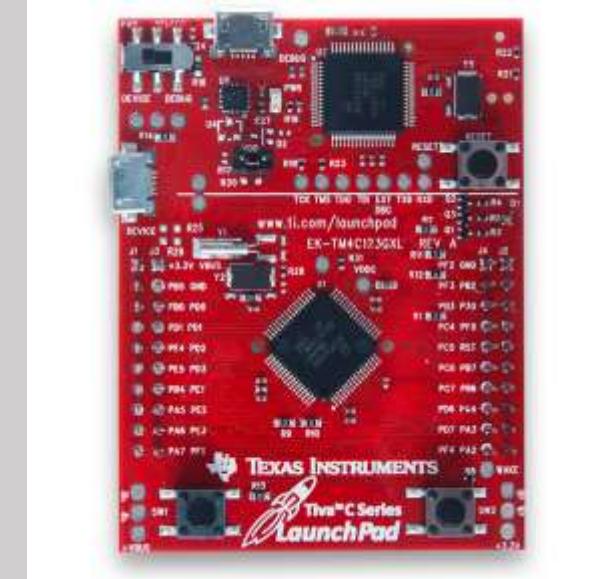
Important Sensor interface Supported in ROS



Arduino board
ROS Interface: `rosserial_arduino`



Phidget
ROS Interface: `phidgets_ros`

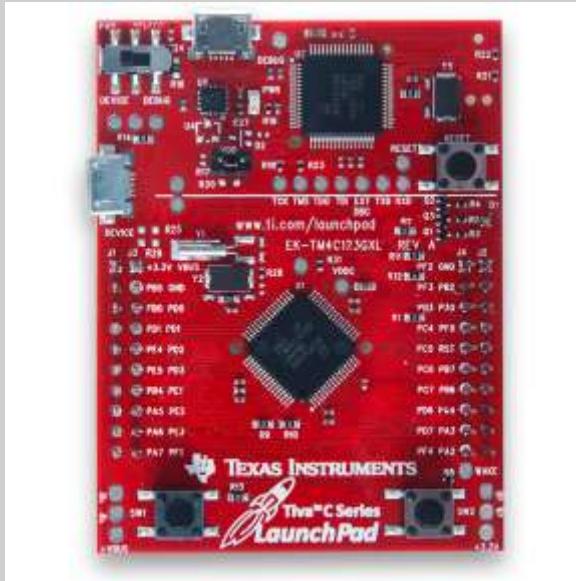


Tiva C Launchpad
ROS Interface : `rosserial_tivac`

Important Sensor interface Supported in ROS



Arduino board



Tiva C Launchpad



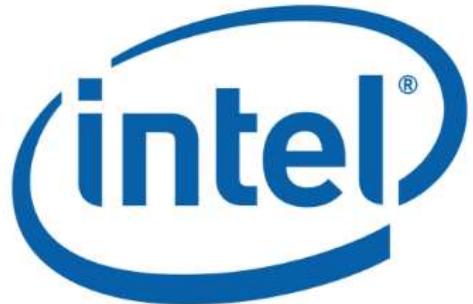
STM 32 Discovery

Library and Simulators supported in ROS



Who all using ROS?

Platinum Sponsor



Gold Sponsors



CLEAR PATH



Erle Robotics



GaiTech



LOCUS



Rapyuta Robotics

SICK

Sensor Intelligence.



sick

intelligent sensors

SICK

Who all using ROS?

Silver Sponsors

ARM



dabit
industries

Northwestern

McCORMICK SCHOOL
OF ENGINEERING

MS in Robotics

QUALCOMM®



Bronze Sponsors



Apex.AI

AVIDBOTS

bonsai

BOSCH
Invented for life



MAGAZINO
simple storage

NVIDIA.

PAL
ROBOTICS

PlusOne
Robotics

rethink
robotics.

Robotnik

Tango

VOYAGE

YUWINROBOT

Who all using ROS?

- Robotics researchers/students
- Professionals working in Robotics companies
- Robotics Hobbyist
- Robot and sensor providers

Jobs and research Opportunities

The screenshot shows the Glassdoor homepage with a search bar for "robotics engineer" and a location filter set to "Los Angeles, CA". The main content area displays a list of job postings:

- Principal Robotics Engineer** at NASA's Jet Propulsion Laboratory – Pasadena, CA (4.2★) - Est. Salary \$105k - \$162k. Posted 11 days ago.
- Technical Sales Engineer, Robotics - West Coast** at Sharp Electronics Corporation – Los Angeles, CA (3.3★) - Est. Salary \$57k - \$97k. Posted 1 day ago.
- Principal Robotics Engineer** at Hyperloop One – Los Angeles, CA (4.7★) - Est. Salary \$163k - \$230k. Posted 17 days ago.
- SENIOR ROBOTICS ENGINEER** at Hyperloop Technologies – Los Angeles, CA (4.1★) - Est. Salary \$88k - \$123k. Posted 12 days ago.

On the right side, there is a detailed view of the first job listing, "Principal Robotics Engineer" at NASA's Jet Propulsion Laboratory. It includes a "Create Job Alert" button, "Apply on Company Site" button, and a "Save" button. Below the job title, there are tabs for "Job", "Company", "Rating", and "Reviews". A section titled "PREFERRED EXPERIENCE:" lists the following requirements:

- PhD degree in robotics, computer science or electrical engineering.
- Experience with GPU programming, ROS
- Experience in writing codes for any of the following systems: sensor fusion, multi-target, tracking system, computer vision algorithms for object recognition.
- Experience with Agile or any other SW development methodologies.
- Experience with Python, shell scripts.
- Hands-on experience in working on any types of active and passive sensor data acquisition pipeline.
- Any publication records on related fields.

<https://www.glassdoor.com/>

Future Scope

- Software is the key aspect in any robot. If there is no computer program running inside a robot, its just a piece of hardware.
- ROS allows programmers to prototype and deploy a robot faster than any other framework available now.

Future Scope

- Lot of opportunities coming for Robotics developers across the globe in which ROS is an important skillset.
- Other than Robotics companies, almost all the universities working in Robotics using ROS in someway.
- ROS is becoming a standard in robotics programming

ROSCon 2019



<https://roscon.ros.org/2019/>

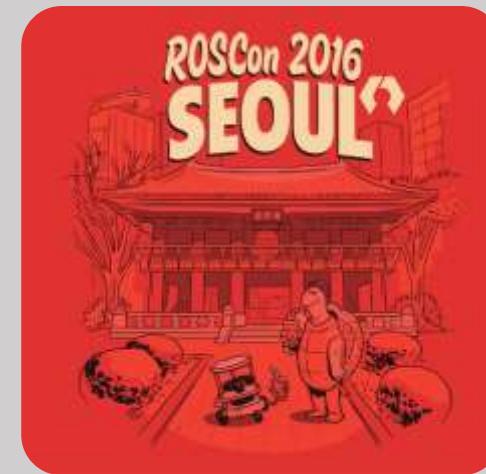
ROSCon 2019



<https://roscon.ros.org/2019/>

ROSCon 2016 & ROSCon 2015 Videos

- <https://roscon.ros.org/2016/>
- <https://roscon.ros.org/2015/>



Getting ROS support

- ROS Mailing list: ros-users@ mailing list
- ROS Wiki: <http://wiki.ros.org/>
- ROS Answers: <http://answers.ros.org/questions/>
- Issue trackers: <http://wiki.ros.org/Tickets>
- ROS Discourse Forums:
<https://discourse.ros.org/>

Getting ROS news

- <http://www.ros.org/news/>
- Latest updates of ROS release, new jobs in ROS, new packages in ROS

ROS Robot Demo: Simulation

ROS Robot Demos



ROS Robot Demos

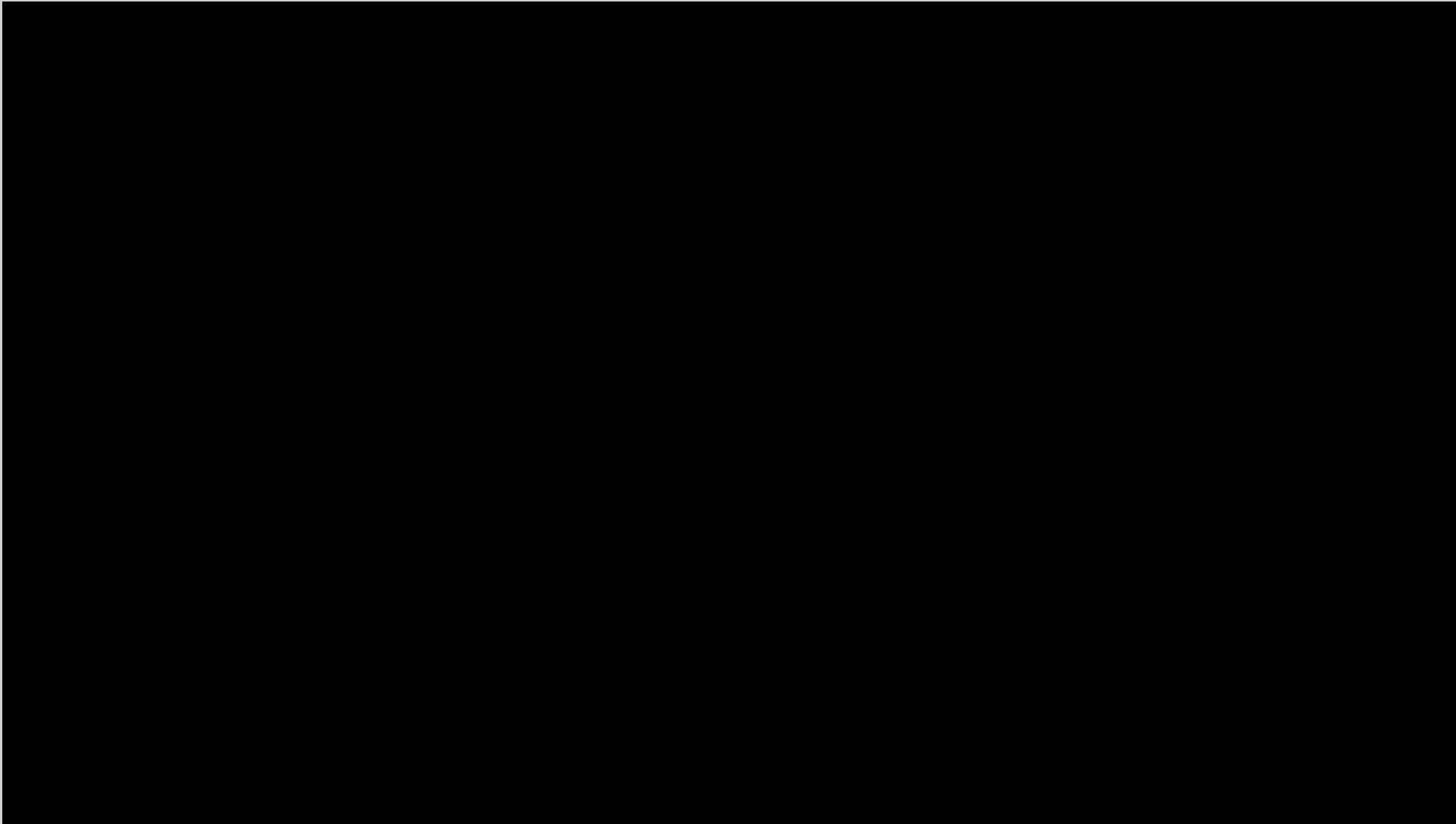
PR2 Making Popcorn at Automatica Fair 2012



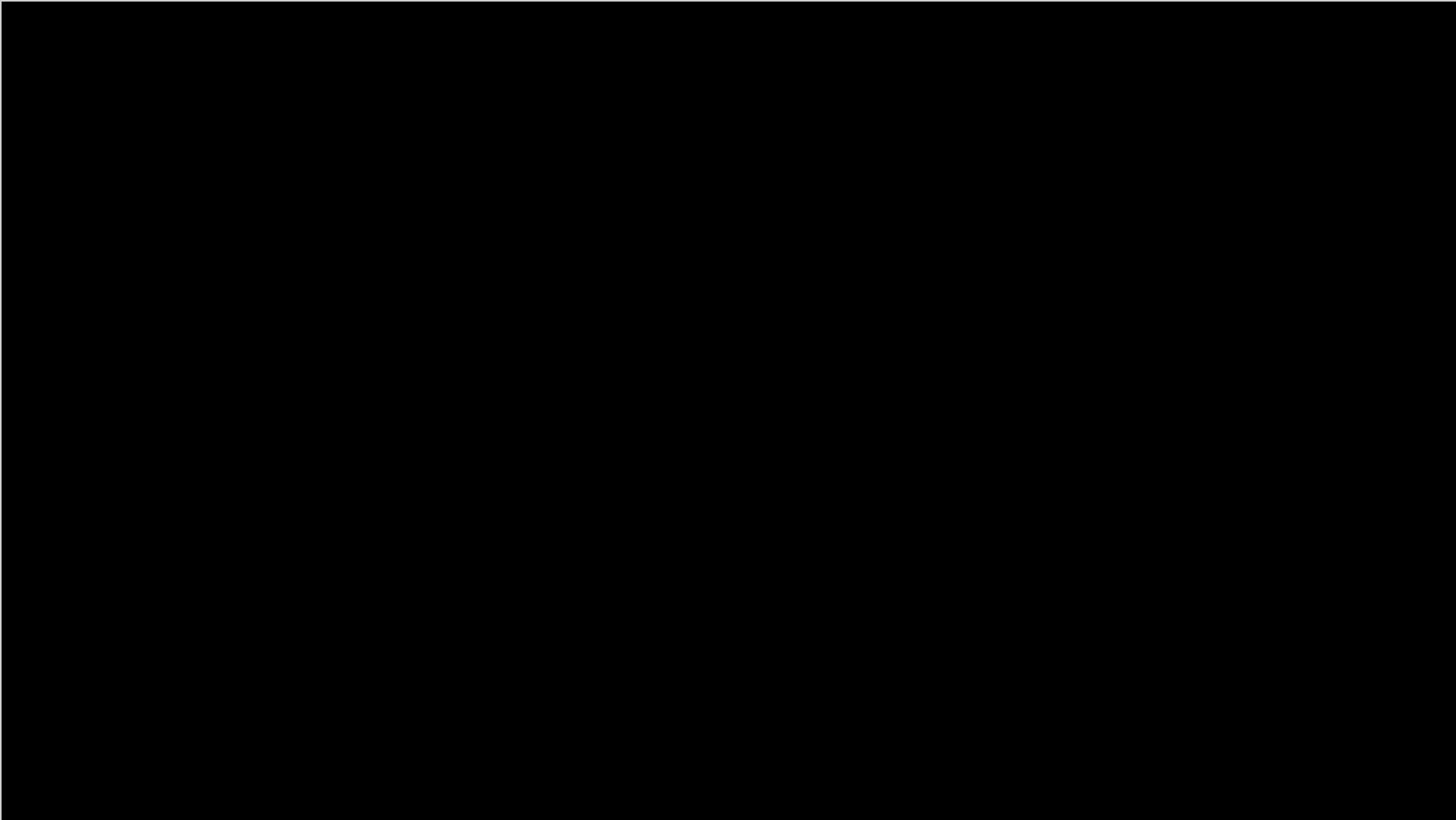
Intelligent
Autonomous
Systems



ROS Robot Demos



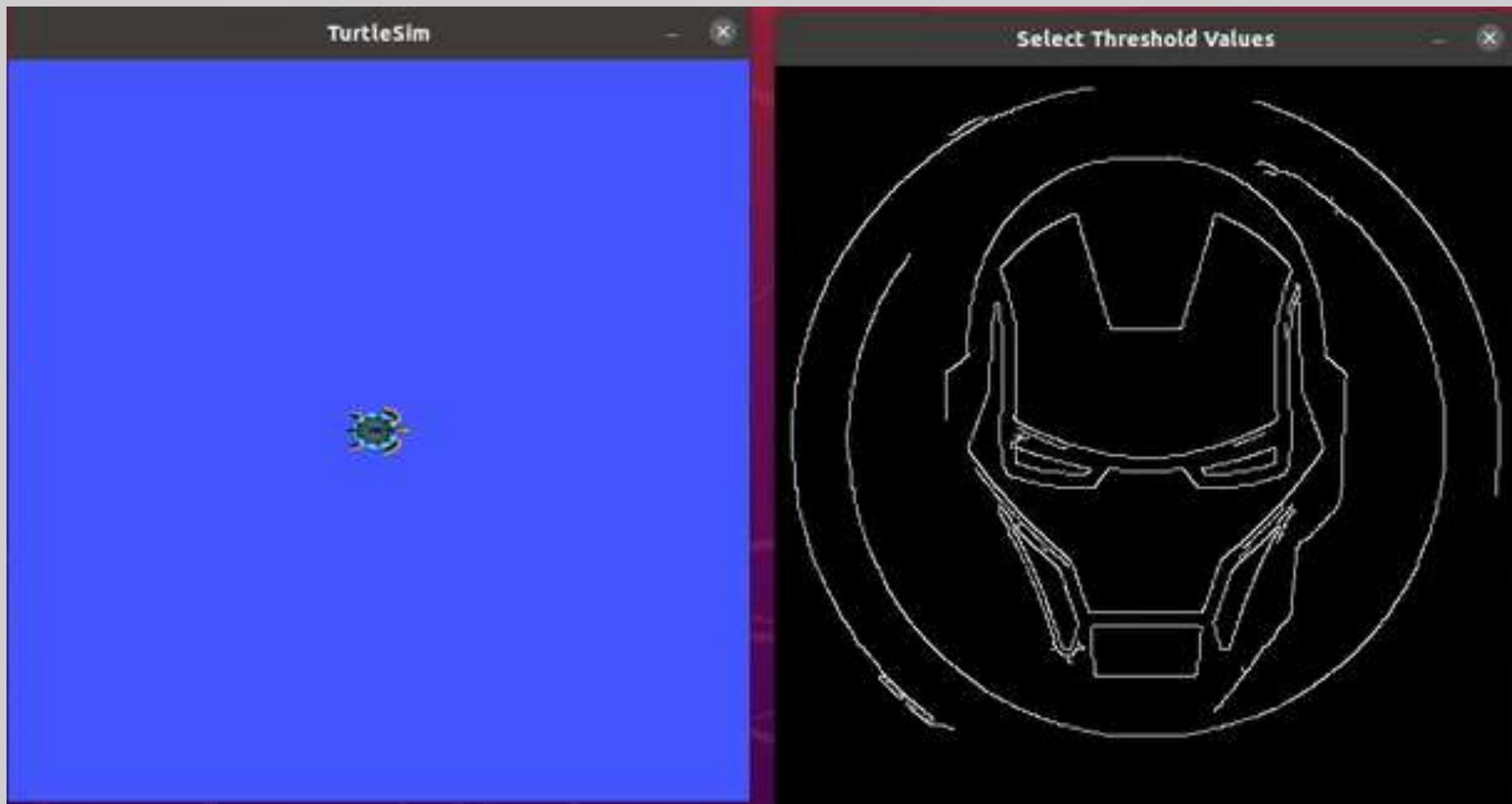
ROS Robot Demos



ROS Robot Demos



Turtlesim Demo: Draw Robot



Turtlesim Demo: Ping Pong Robot



ROS Navigation and Manipulation

