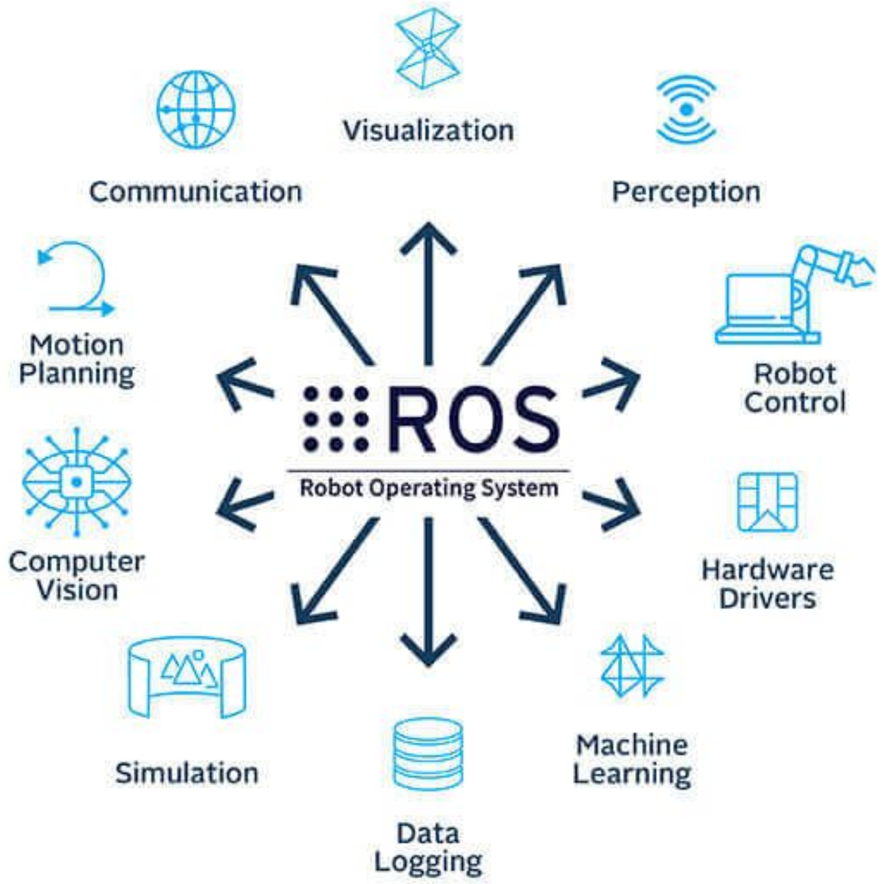
Robot Operating System

Ing. Domsa Victor
Sl.dr.ing. Levente Tamas

What is ROS?

Not an Operating System…



What is ROS?

- A collection of packaging, software building tools
- Development tools for system runtime and data analysis
- Open-source under permissive BSD licenses (ros core libraries)
- A language-independent architecture (c++, python, lisp, java, and more)
- A scalable platform (ARM CPUS to Xeon Clusters)

How it came to be?

STANFORD UNIVERSITY
Personal Robotics Program

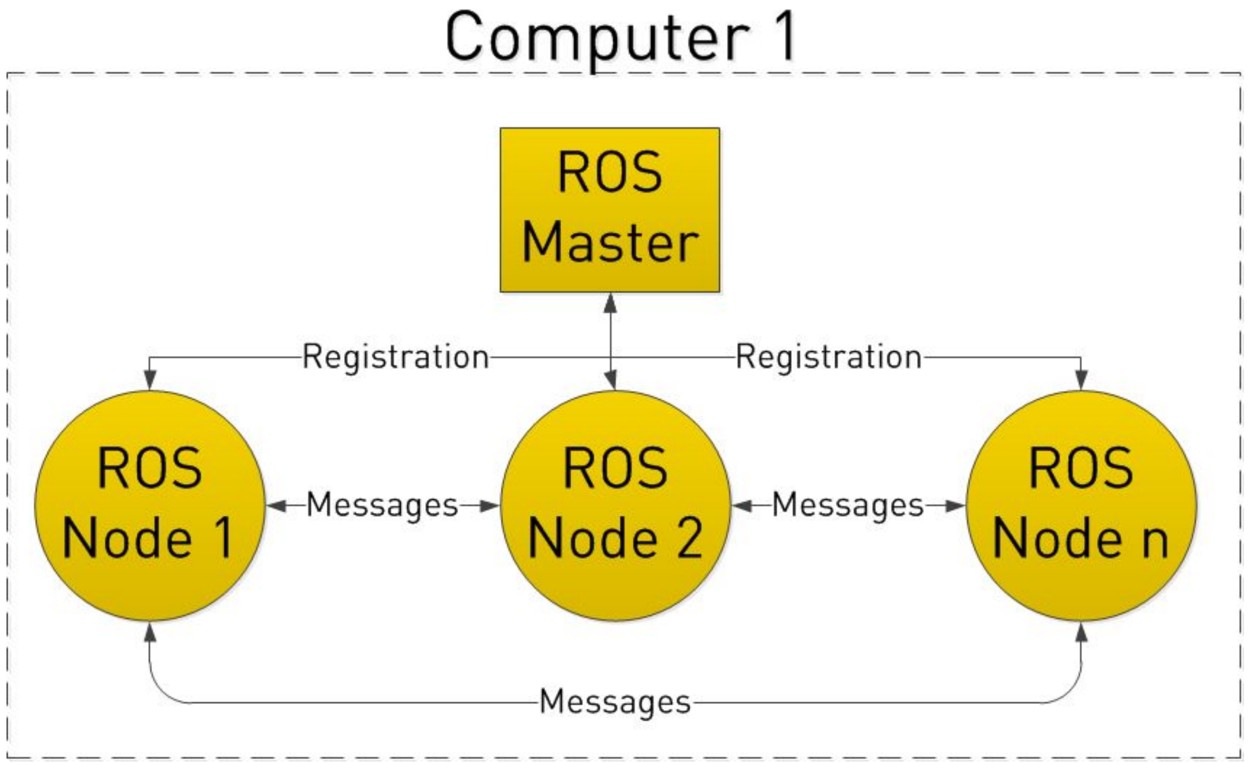
How does it work?

It has a core:

- ROS Master
  - centralized XML-RPC server
- Parameter Server
  - stores configuration params/data
- Rosout
  - network-based stdout for messages

Images from official *clearpathrobotics* ROS tutorial

ROS Master and Nodes



Nodes

- processes running inside ROS
- can be written in different programming languages
- source *and/or* sink for data sent over the ROS network (talker *and/or* subscriber)
- expect to receive specific data types (messages)
- send specific data types (messages)
- a talker specifies ***to*** which ‘topic’ it wants to send data
- a subscriber specifies ***from*** which ‘topic’ it wants to receive data

Communication

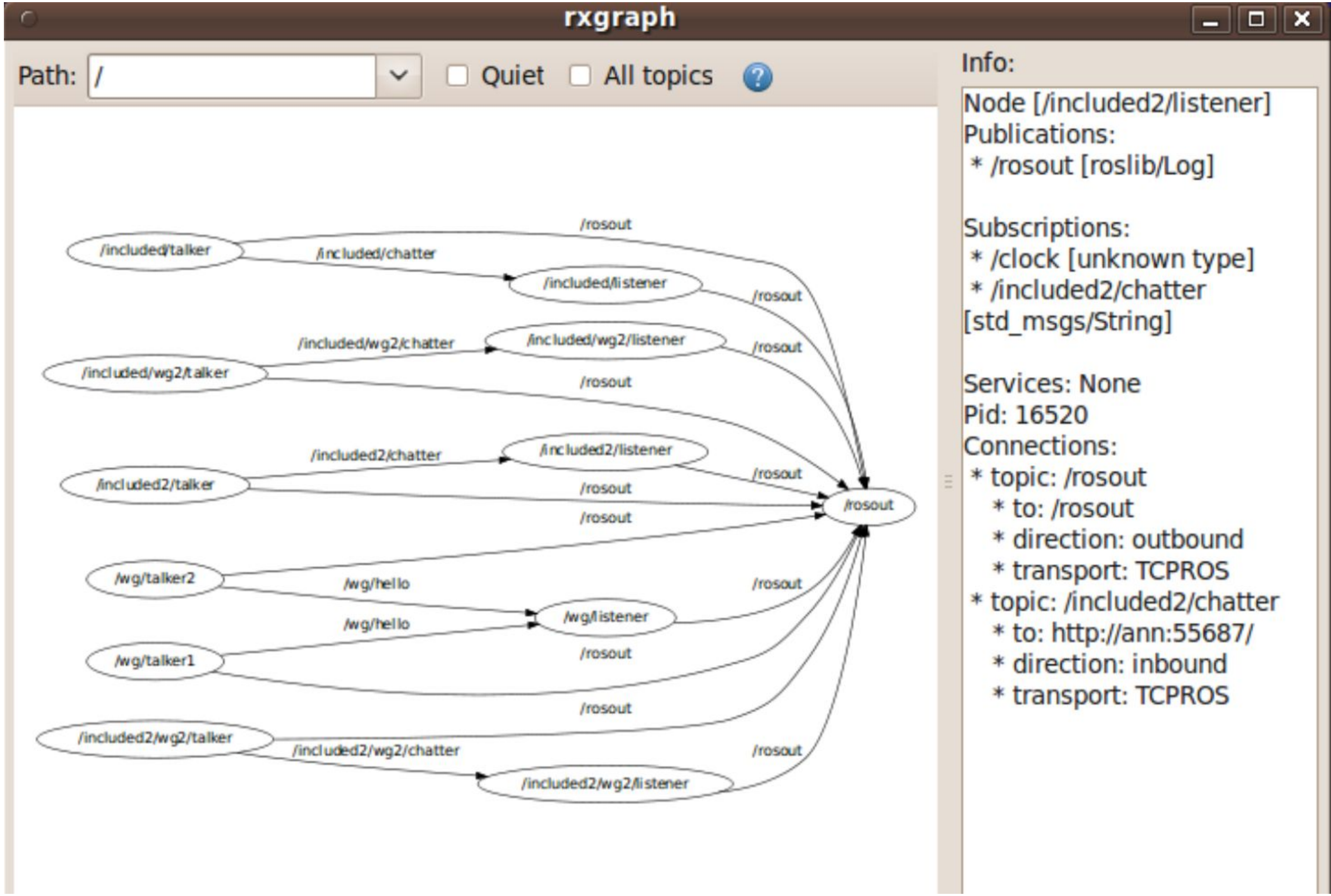
Topics

- Asynchronous many-to-many communication
- Can have one or more publishers
- Can have one or more subscribers
- Strongly-typed (ROS .msg spec)
- TCP/IP or UDP-based Transport

Services

- Synchronous function-call-like communication
- Can have only one server
- Can have one or more clients
- Strongly-typed (ROS .srv spec)
- TCP/IP or UDP Transport

How to see all the nodes?



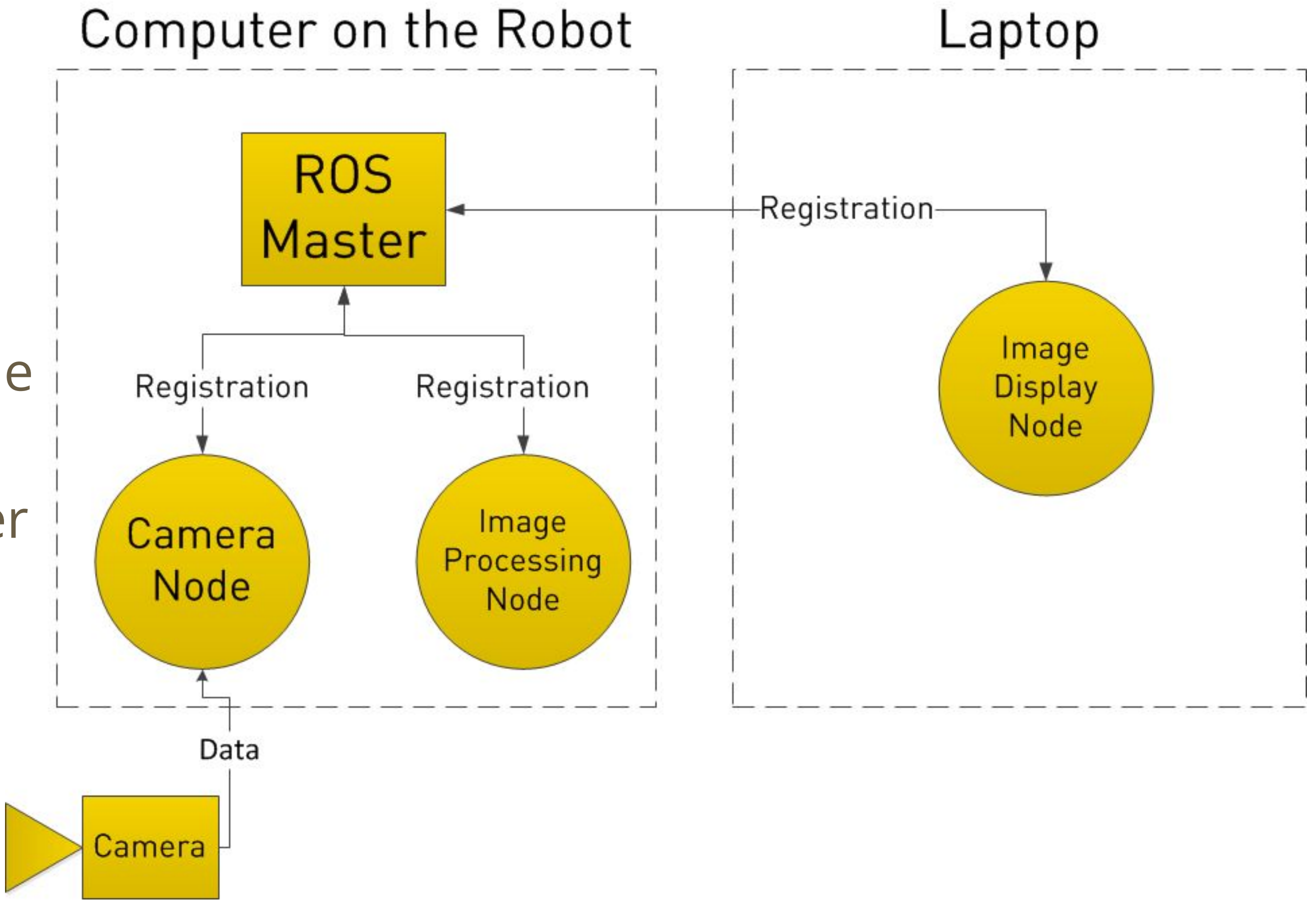
Node [/included2/listener]
Publications:
 * /rosout [roslib/Log]

Subscriptions:
 * /clock [unknown type]
 * /included2/chatter
[std_msgs/String]

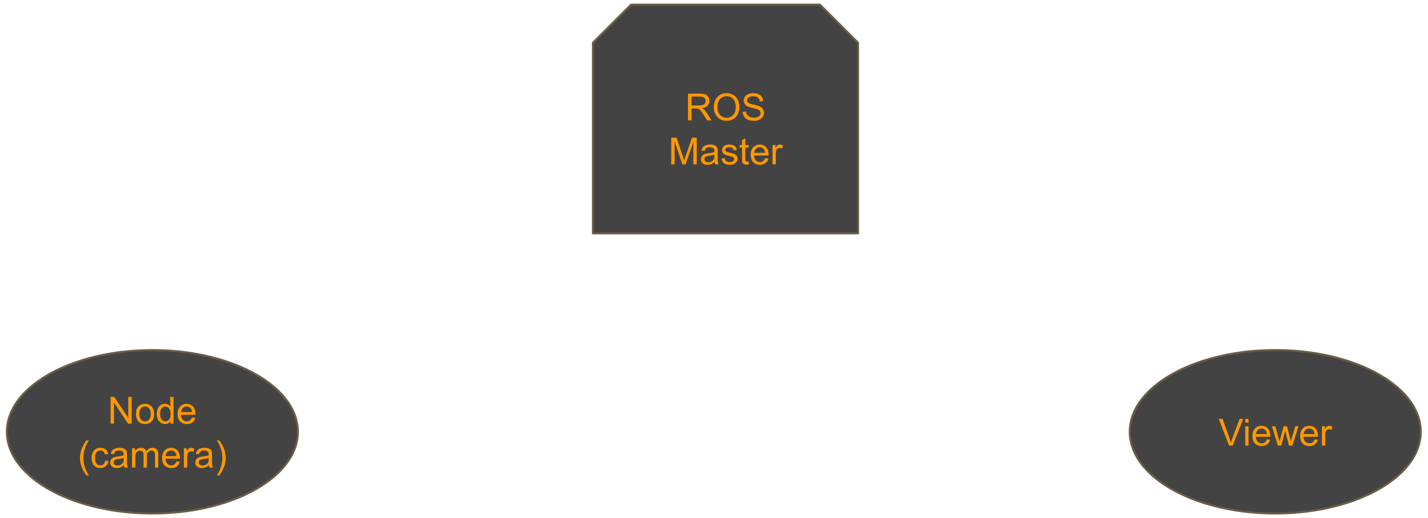
Services: None
Pid: 16520
Connections:
 * topic: /rosout
 * to: /rosout
 * direction: outbound
 * transport: TCPROS
 * topic: /included2/chatter
 * to: http://ann:55687/
 * direction: inbound
 * transport: TCPROS

Multi-Platform?

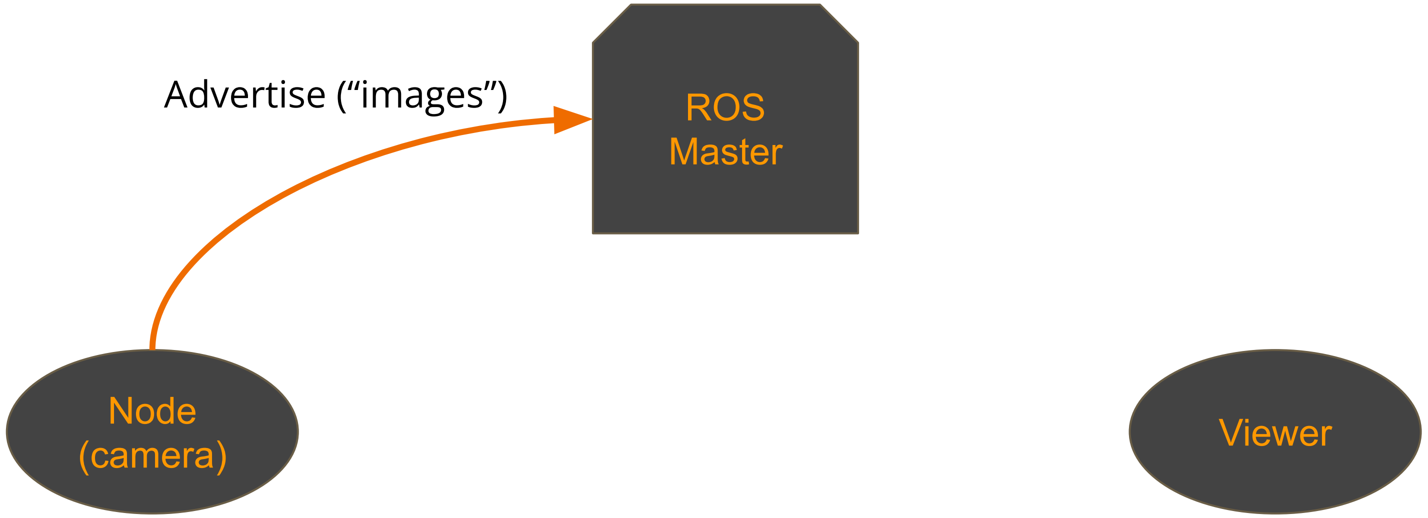
- It can run on multiple PCs/robots
- Only one ROS Master



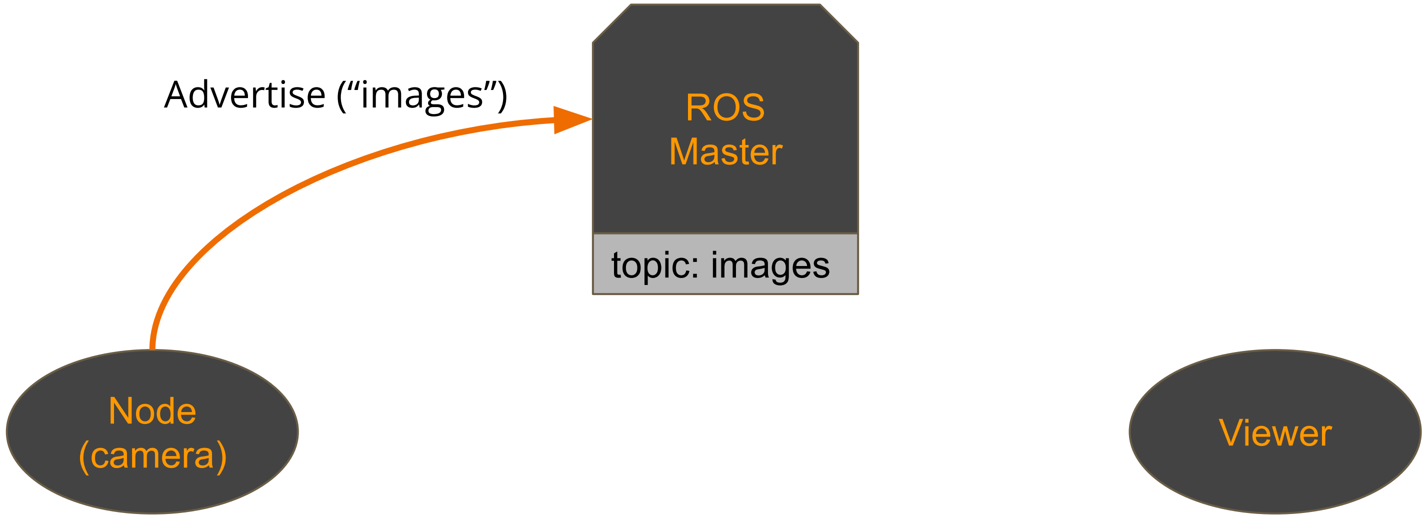
Asynchronous Distributed Communication



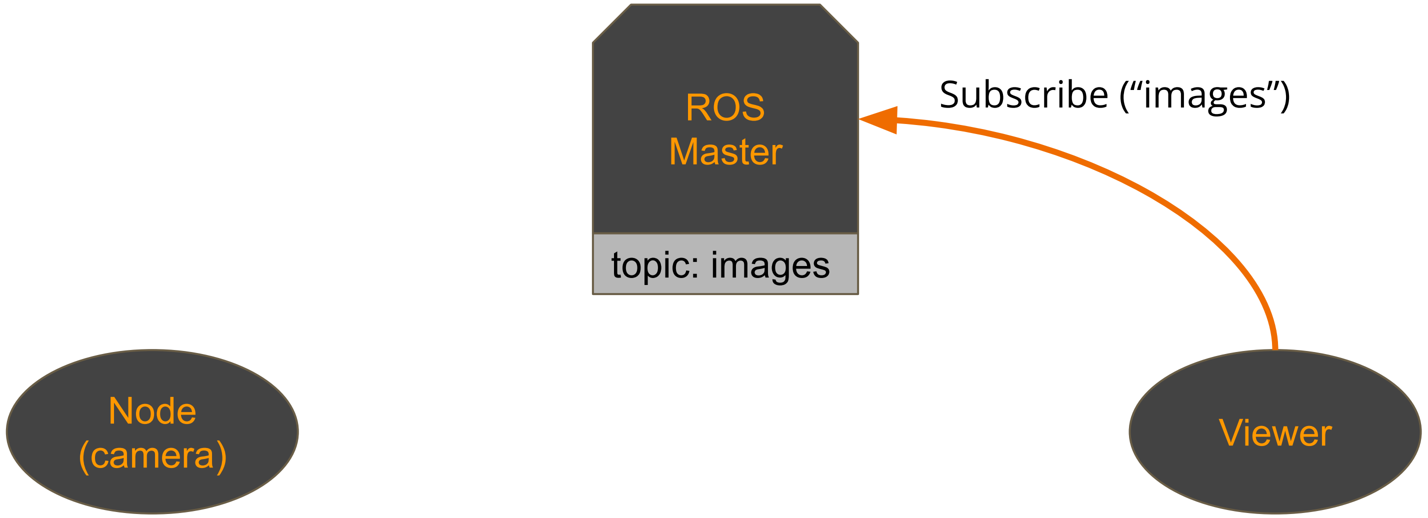
Asynchronous Distributed Communication



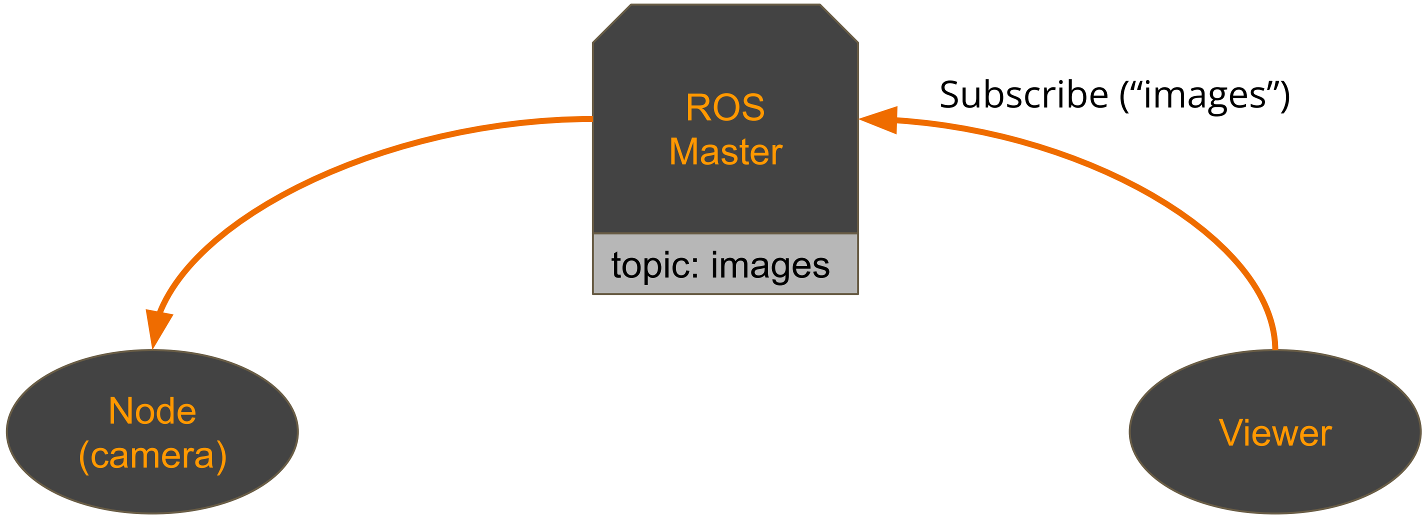
Asynchronous Distributed Communication



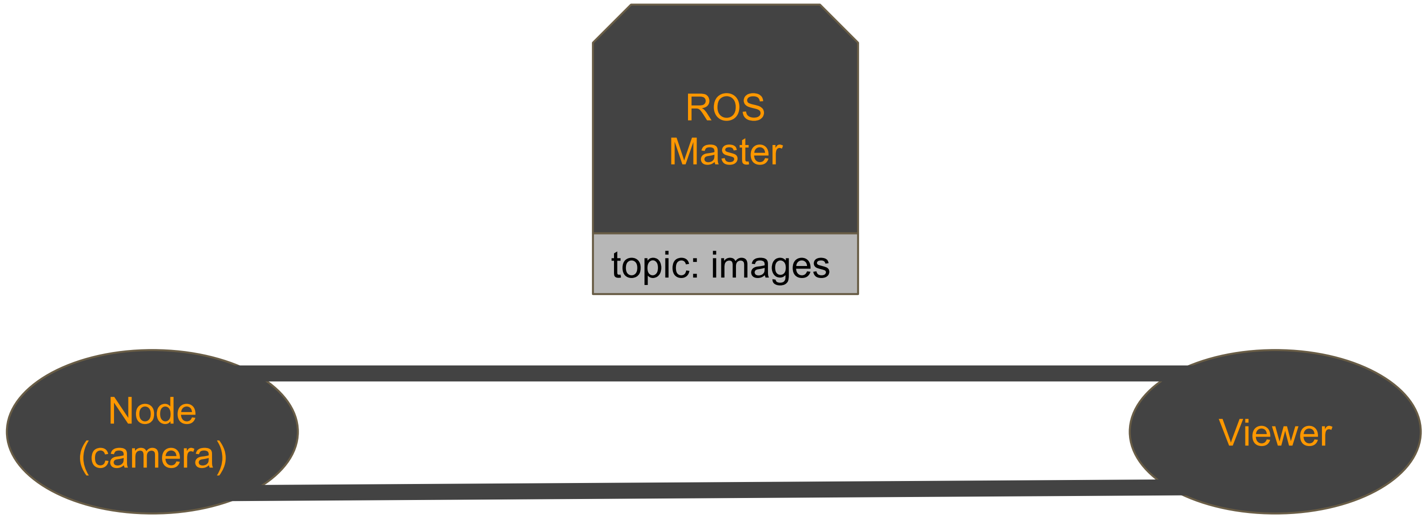
Asynchronous Distributed Communication



Asynchronous Distributed Communication

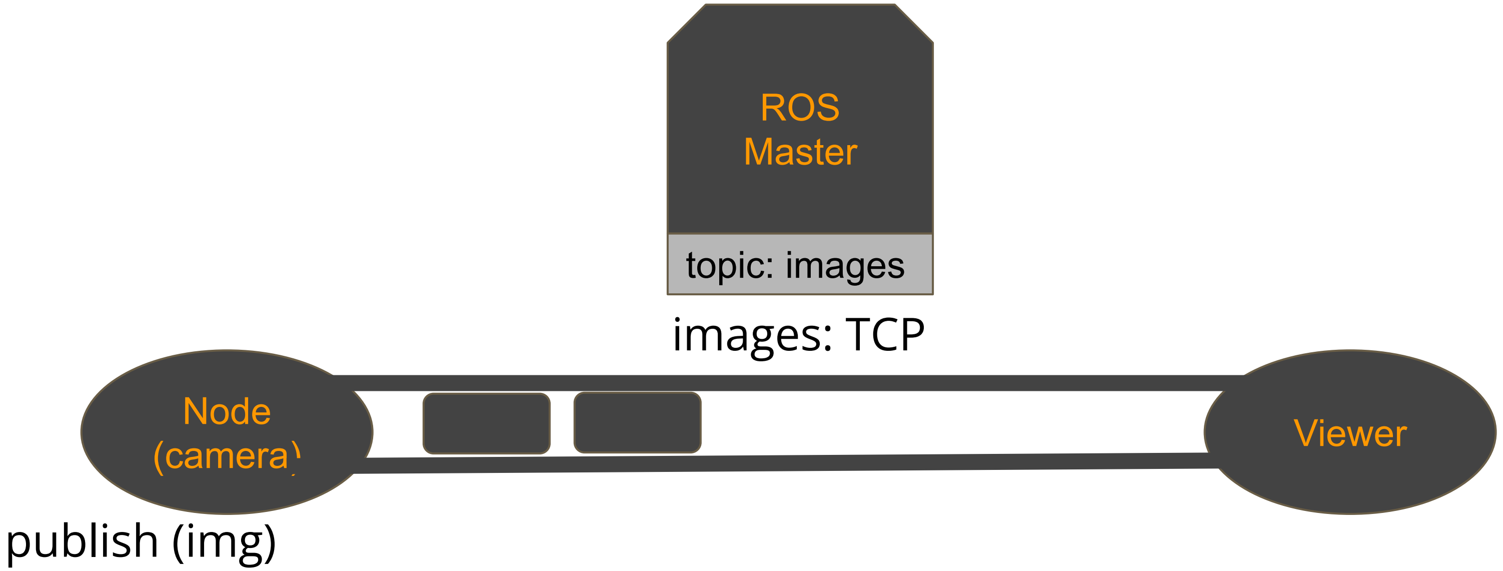


Asynchronous Distributed Communication

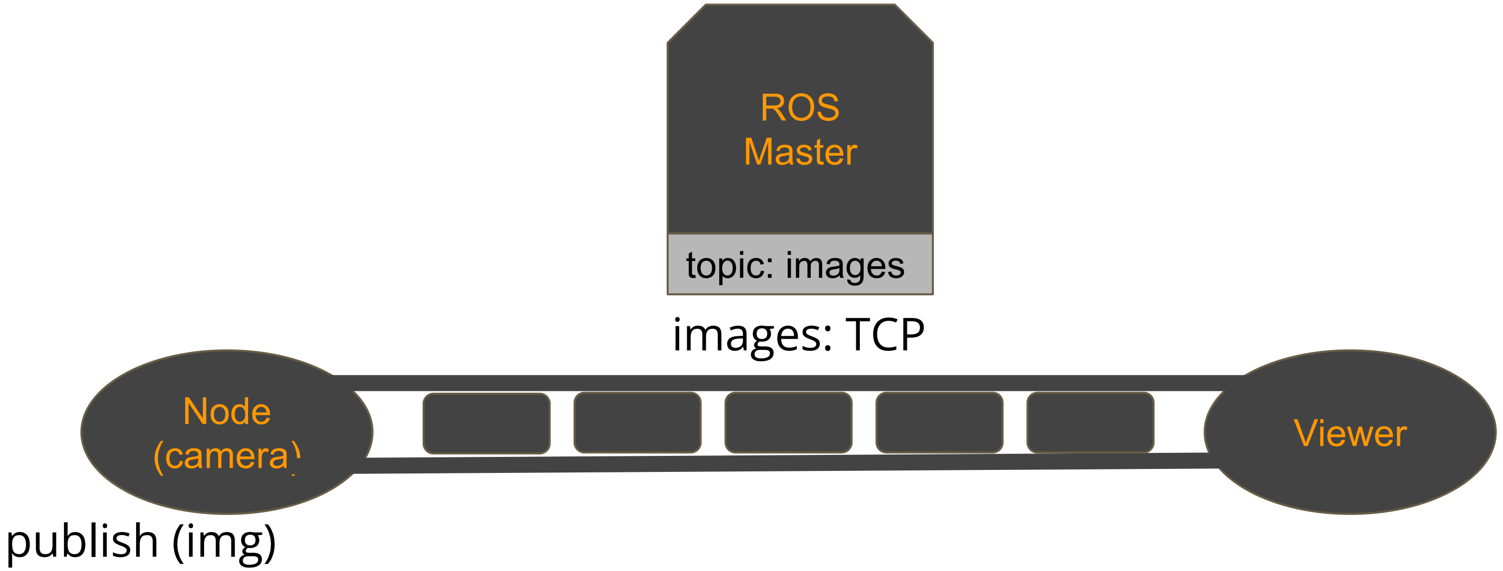


images: TCP

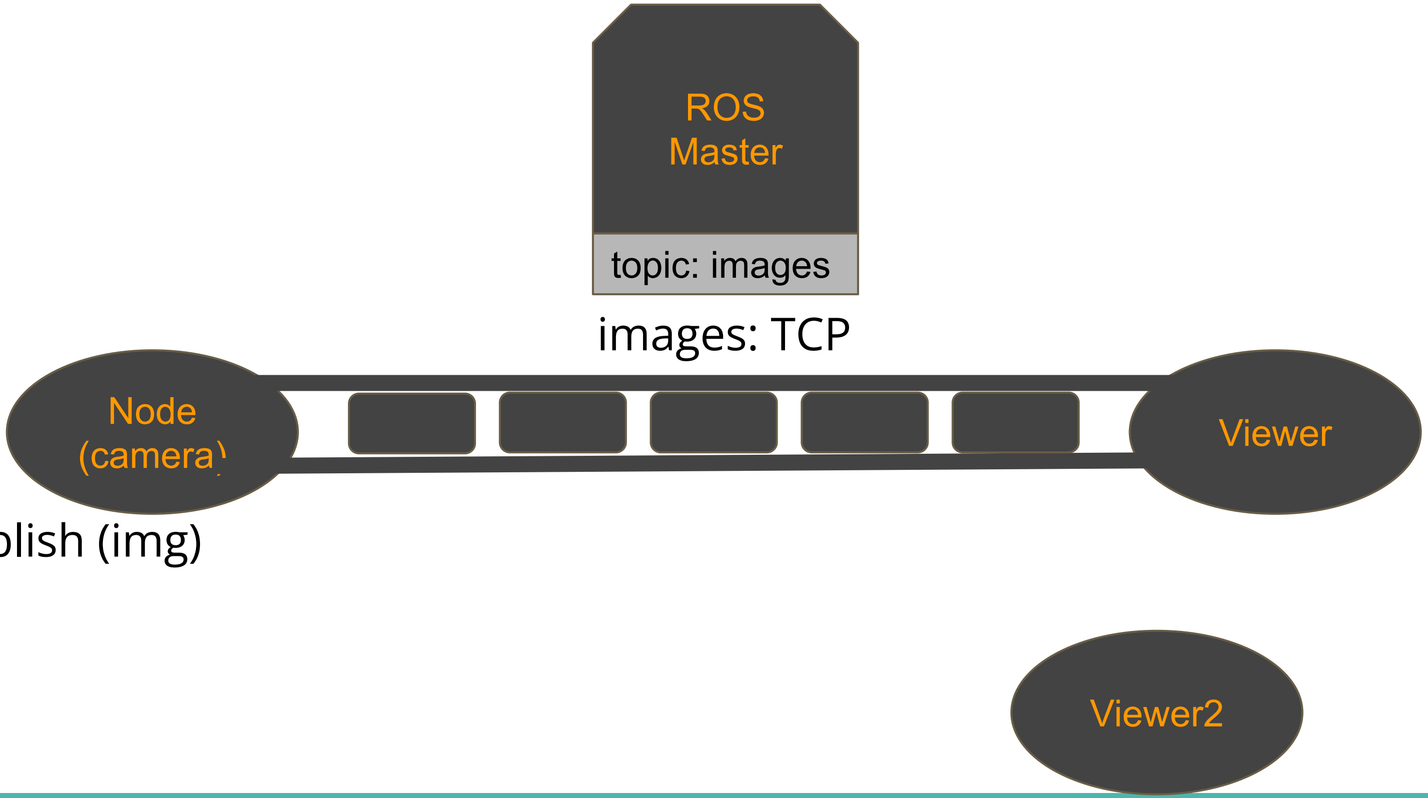
Asynchronous Distributed Communication



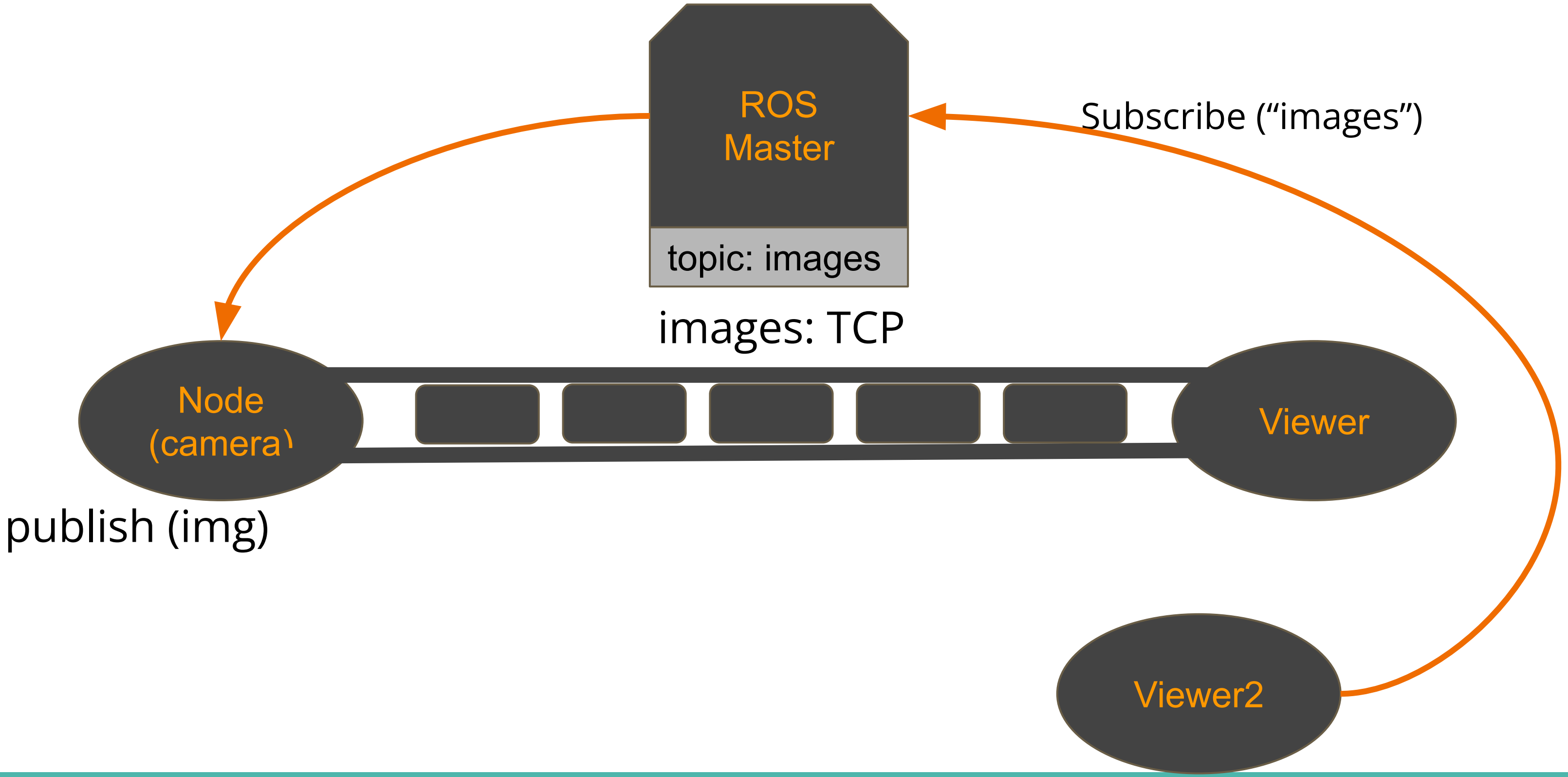
Asynchronous Distributed Communication



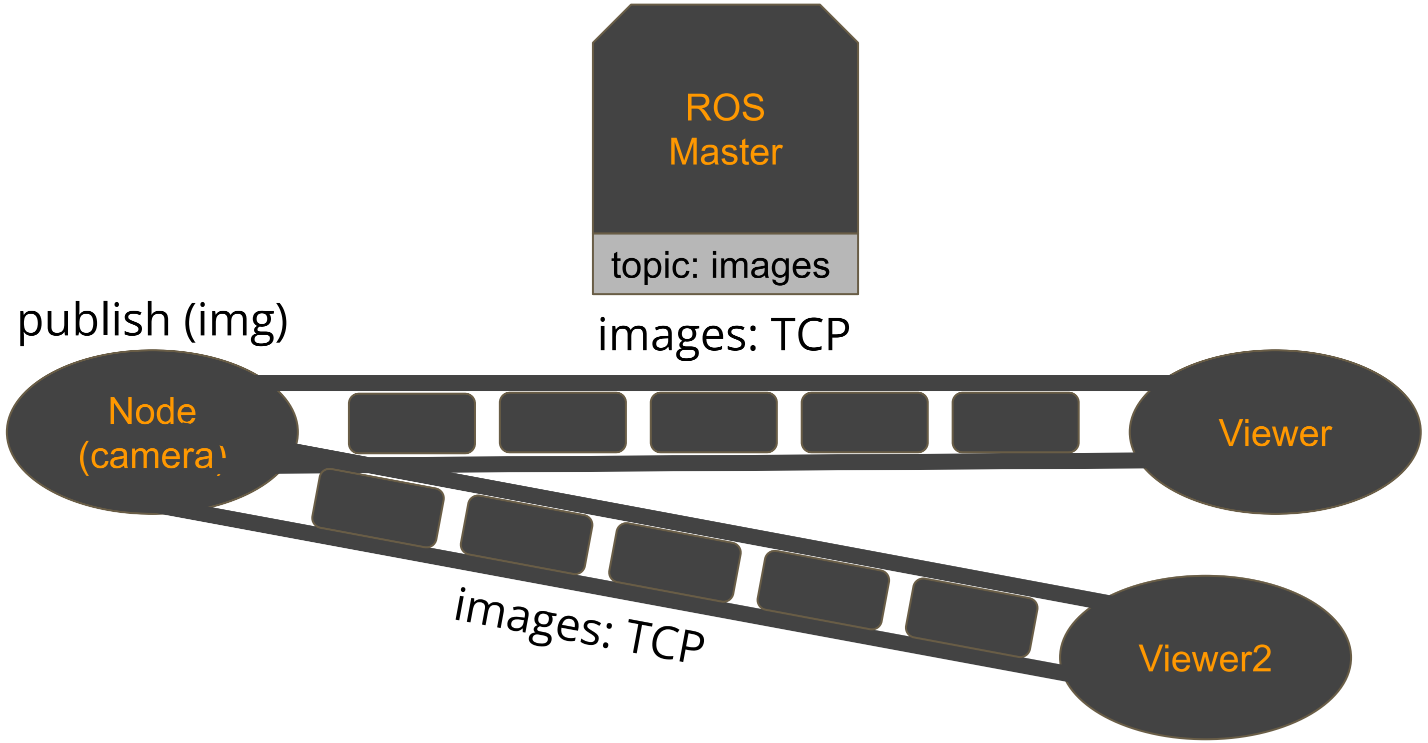
Asynchronous Distributed Communication



Asynchronous Distributed Communication



Asynchronous Distributed Communication



images: TCP

Creating and running ROS Nodes

Launch files

- XML file
- contains parameters and configurations
- used by the 'roslaunch' tool

Creating and running ROS Nodes

Launch files enable users to:

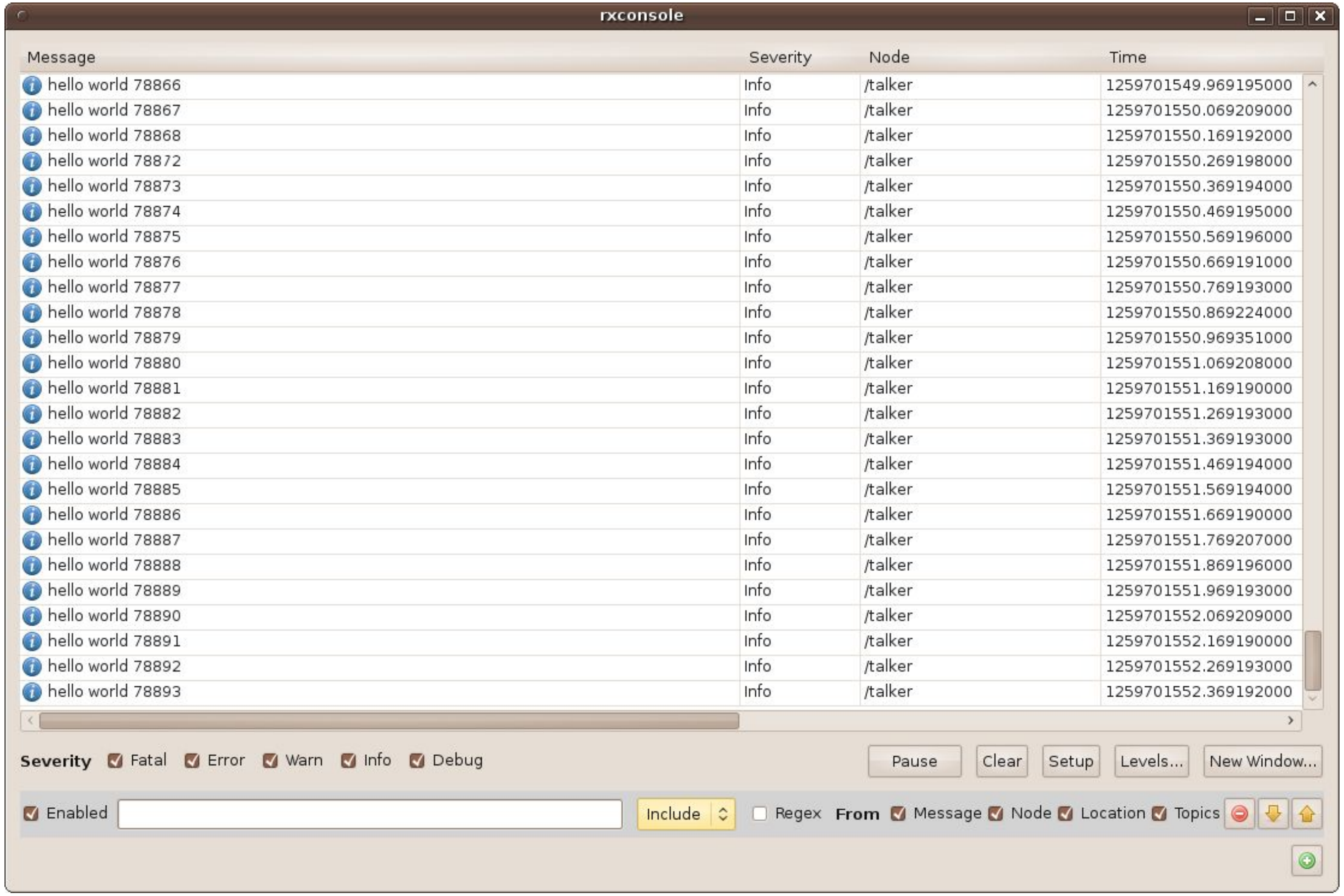
- Associate a set of parameters and nodes with a single le
- Automatically re-spawn nodes if they crash
- Change node names, namespaces, topics, and other resource names without recompiling
- Easily distribute nodes across multiple machines

ROS Graph Introspection

ROS provides several tools for analyzing the data owing over ROS communication resources:

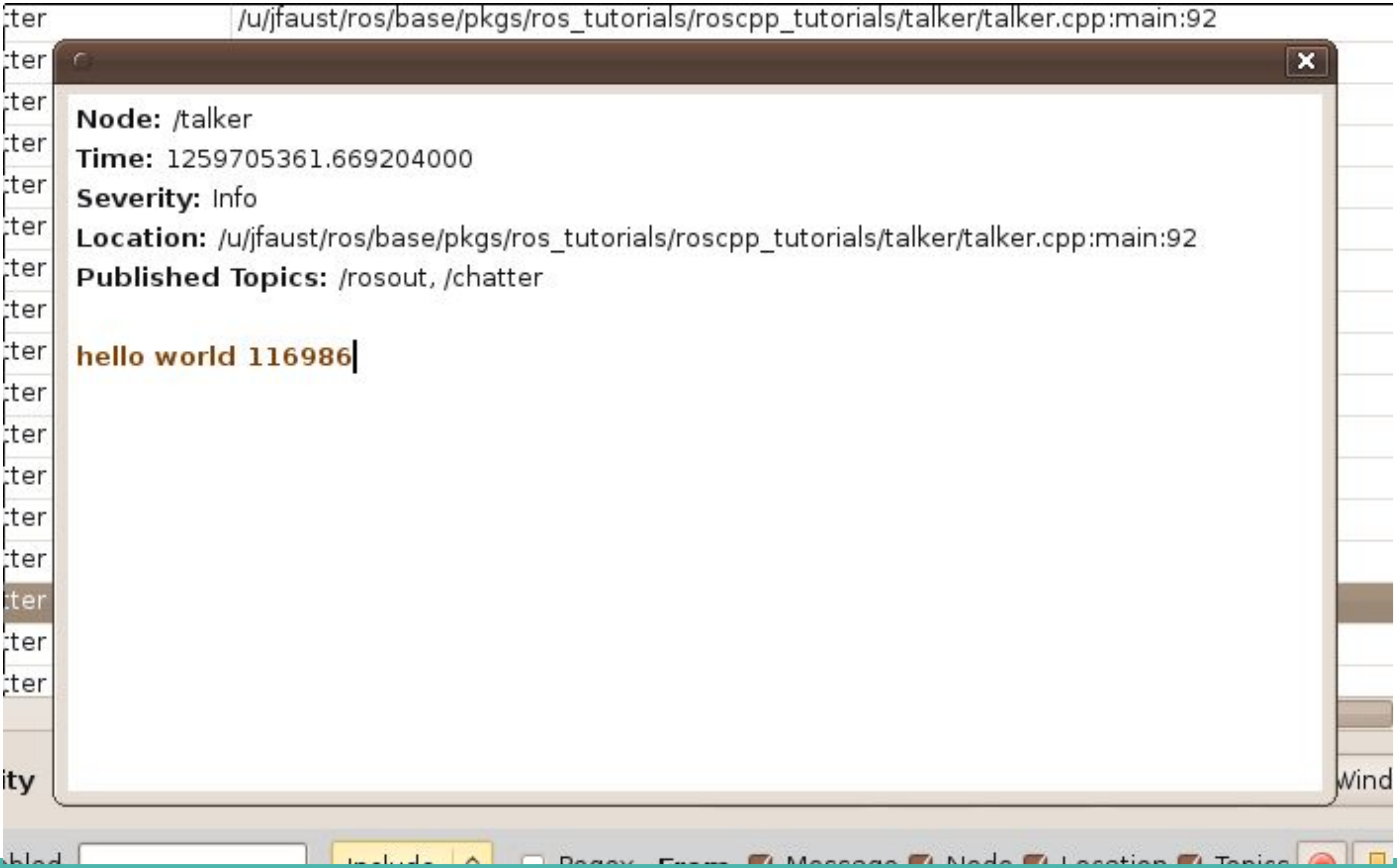
- **rosnode**
  - Gives a user information about a node: publications, subscriptions, etc
- **rostopic**
  - Gives data rate, actual data, publishers, subscribes
- **rosservice**
  - Enables a user to call a ROS Service from the command line
- **roswtf** (wire trouble finder)
  - Diagnoses problems with a ROS network

ROS Graphical User Interface(s)



Message	Severity	Node	Time
hello world 78866	Info	/talker	1259701549.969195000
hello world 78867	Info	/talker	1259701550.069209000
hello world 78868	Info	/talker	1259701550.169192000
hello world 78872	Info	/talker	1259701550.269198000
hello world 78873	Info	/talker	1259701550.369194000
hello world 78874	Info	/talker	1259701550.469195000
hello world 78875	Info	/talker	1259701550.569196000
hello world 78876	Info	/talker	1259701550.669191000
hello world 78877	Info	/talker	1259701550.769193000
hello world 78878	Info	/talker	1259701550.869224000
hello world 78879	Info	/talker	1259701550.969351000
hello world 78880	Info	/talker	1259701551.069208000
hello world 78881	Info	/talker	1259701551.169190000
hello world 78882	Info	/talker	1259701551.269193000
hello world 78883	Info	/talker	1259701551.369193000
hello world 78884	Info	/talker	1259701551.469194000
hello world 78885	Info	/talker	1259701551.569194000
hello world 78886	Info	/talker	1259701551.669190000
hello world 78887	Info	/talker	1259701551.769207000
hello world 78888	Info	/talker	1259701551.869196000
hello world 78889	Info	/talker	1259701551.969193000
hello world 78890	Info	/talker	1259701552.069209000
hello world 78891	Info	/talker	1259701552.169190000
hello world 78892	Info	/talker	1259701552.269193000
hello world 78893	Info	/talker	1259701552.369192000

Severity ✓ Fatal ✓ Error ✓ Warn ✓ Info ✓ Debug
Pause
Clear
Setup
Levels...
New Window...
✓ Enabled
Include
☐ Regex
From ✓ Message ✓ Node ✓ Location ✓ Topics

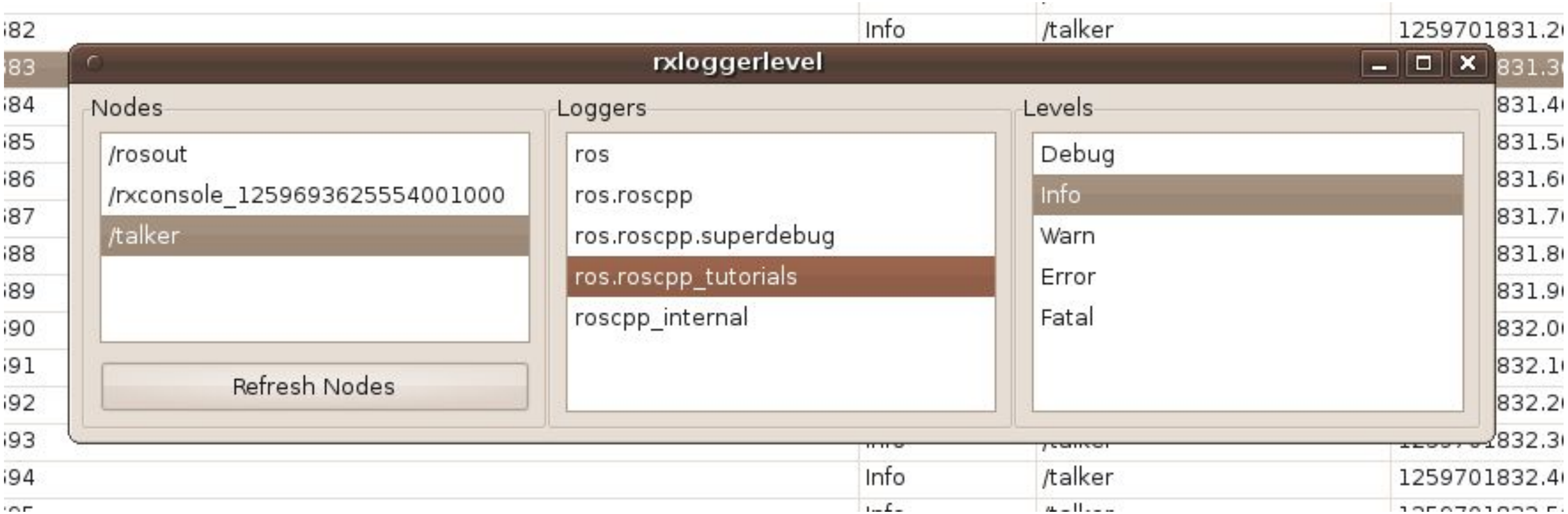


/u/jfaust/ros/base/pkgs/ros_tutorials/roscpp_tutorials/talker/talker.cpp:main:92

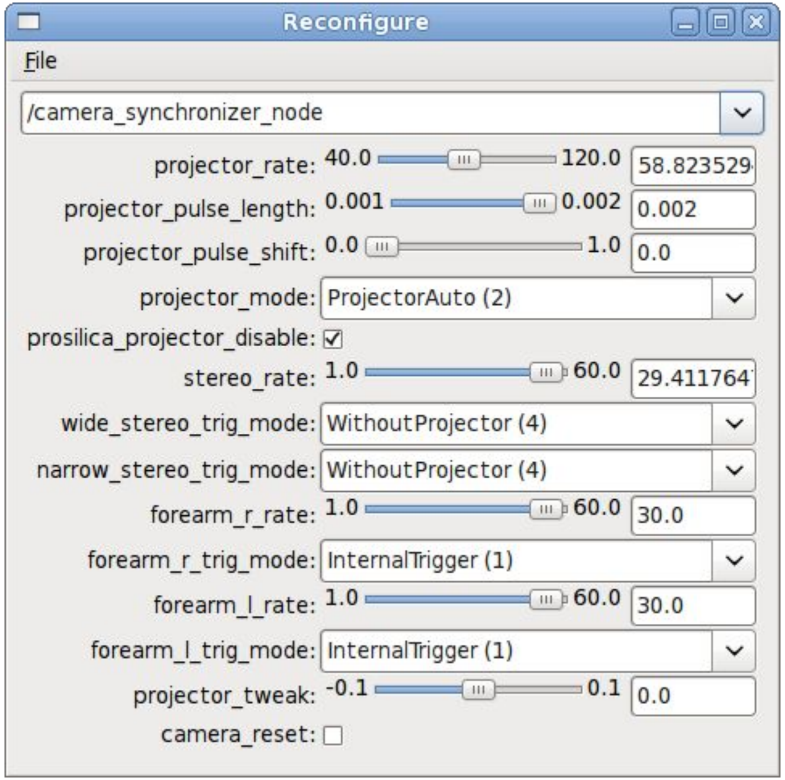
Node: /talker
Time: 1259705361.669204000
Severity: Info
Location: /u/jfaust/ros/base/pkgs/ros_tutorials/roscpp_tutorials/talker/talker.cpp:main:92
Published Topics: /rosout, /chatter

hello world 116986

ROS Graphical User Interface(s)



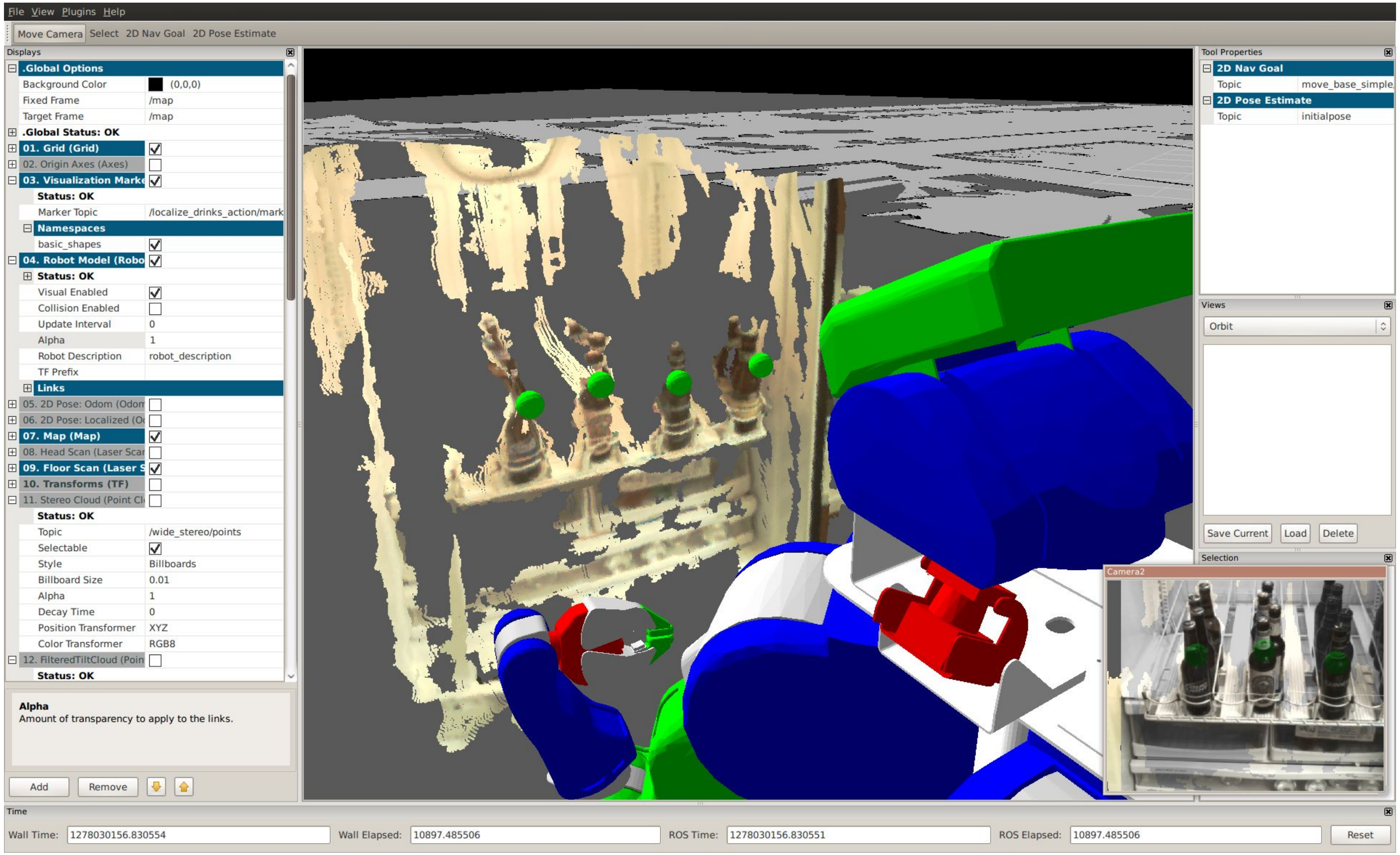
ROS Graphical User Interface(s)



/camera_synchronizer_node

projector_rate:	40.0	120.0	58.823529
projector_pulse_length:	0.001	0.002	0.002
projector_pulse_shift:	0.0	1.0	0.0
projector_mode:	ProjectorAuto (2)		
prosilica_projector_disable:	✓		
stereo_rate:	1.0	60.0	29.411764
wide_stereo_trig_mode:	WithoutProjector (4)		
narrow_stereo_trig_mode:	WithoutProjector (4)		
forearm_r_rate:	1.0	60.0	30.0
forearm_r_trig_mode:	InternalTrigger (1)		
forearm_l_rate:	1.0	60.0	30.0
forearm_l_trig_mode:	InternalTrigger (1)		
projector_tweak:	-0.1	0.1	0.0
camera_reset:	☐		

Rviz



.Global Options	
Background Color	(0,0,0)
Fixed Frame	/map
Target Frame	/map
.Global Status: OK	
01. Grid (Grid)	✓
02. Origin Axes (Axes)	☐
03. Visualization Marke	✓
Status: OK	
Marker Topic	/localize_drinks_action/mark
Namespaces	
basic_shapes	✓
04. Robot Model (Robo	✓
Status: OK	
Visual Enabled	✓
Collision Enabled	☐
Update Interval	0
Alpha	1
Robot Description	robot_description
TF Prefix	
Links	
05. 2D Pose: Odom (Odom	☐
06. 2D Pose: Localized (O	☐
07. Map (Map)	✓
08. Head Scan (Laser Scan	☐
09. Floor Scan (Laser S	✓
10. Transforms (TF)	☐
11. Stereo Cloud (Point Cl	☐
Status: OK	
Topic	/wide_stereo/points
Selectable	✓
Style	Billboards
Billboard Size	0.01
Alpha	1
Decay Time	0
Position Transformer	XYZ
Color Transformer	RGB8
12. FilteredTiltCloud (Poin	☐
Status: OK	

Alpha
Amount of transparency to apply to the links.

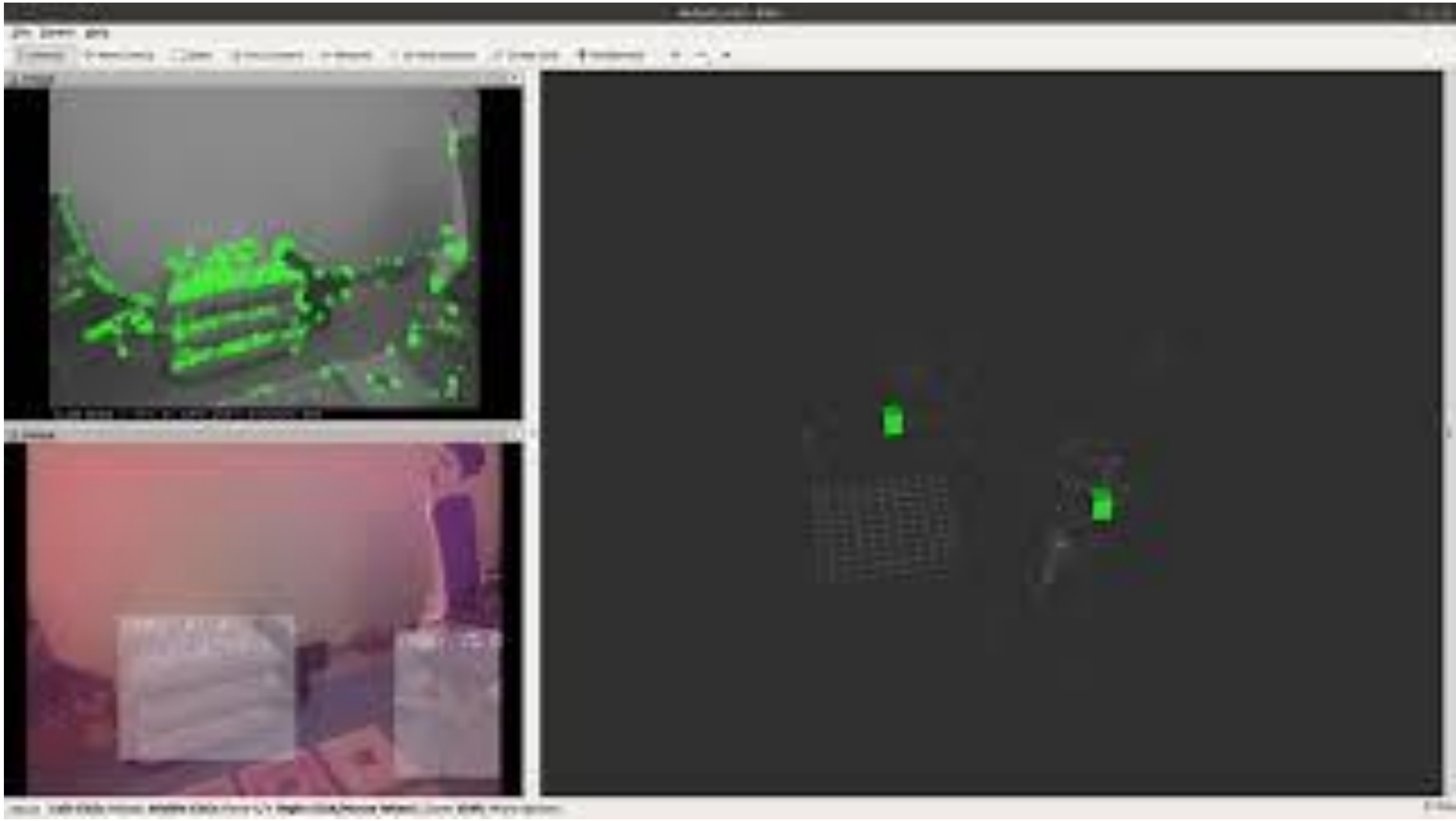
2D Nav Goal	
Topic	move_base_simple
2D Pose Estimate	
Topic	initialpose

Views: Orbit

Time

Wall Time: 1278030156.830554 Wall Elapsed: 10897.485506 ROS Time: 1278030156.830551 ROS Elapsed: 10897.485506

Rviz



ROS Meta-Filesystem

- This meta-system allows ROS (rospack, specifically) to locate any package in the designated path, be it at compile time or runtime.
- Since ROS can find any package at any time, it enables packages to be moved around in the actual system for greater codebase flexibility.
- ROS uses CMake (www.cmake.org) internally to compile and link code, and some more useful tools (e.g. catkin tools).

Packages - ROS Meta-Filesystem

The minimal representation of a ROS package is a directory in the $ROS_PACKAGE_PATH which contains:

- **manifest.xml**
  - Contains package metadata (author, license, url, etc)
  - Species system and package dependencies
  - Species language-specific export ags
- **CMakeLists.txt**
  - Contains ROS build rules (executables, libraries, custom build ags, etc)
- **Makefile**
  - Just a proxy to build this package

Thank You