Robot Operating System

Fundamentals

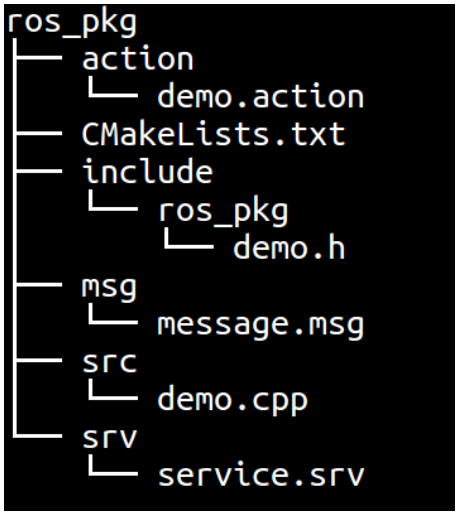
Important Concepts in ROS

- **Package**
- **Node**
- **Topic**
- **Message**
- **Service**
- **Master node**
- **Publisher-Subscriber**

Package

What is a ROS Package?

A ROS package is a basic unit in ROS system. It contains one or more nodes and provides ROS messages or services. Each package is defined via an XML file which describes package and its dependencies.



Create Your Package

1)
```
Ubuntu:~$ cd catkin_ws/src/
```

2)
```
Ubuntu:~/catkin_ws/src$ catkin_create_pkg demo_pckg rospy std_msgs
```

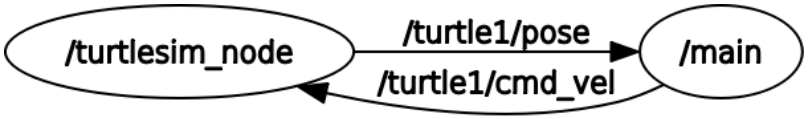
3)
```
Ubuntu:~/catkin_ws/src$ tree demo_pckg/
demo_pckg/
├── CMakeLists.txt
├── package.xml
└── src

1 directory, 2 files
```

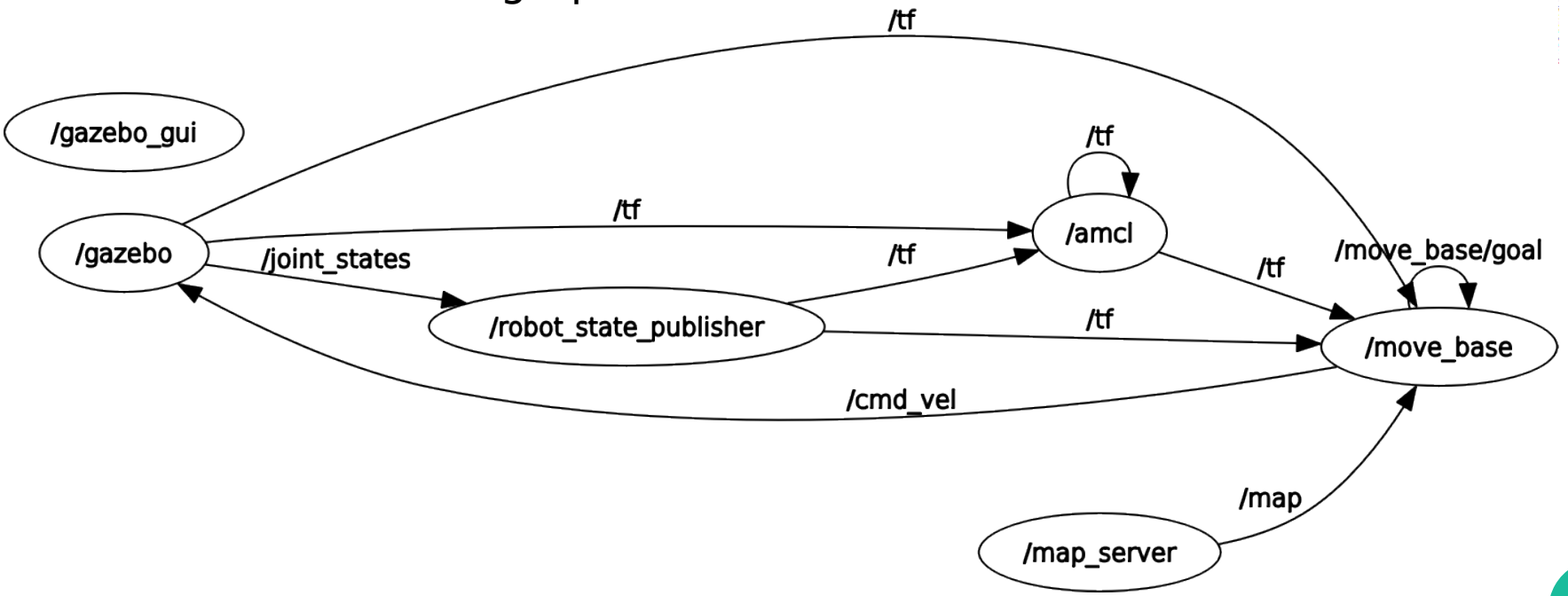
4)
```
Ubuntu:~/catkin_ws/src$ cd ..
```

5)
```
Ubuntu:~/catkin_ws$ catkin_make
```

Node

- Each node is a process that performs computation.
- Node is like a software module.
- Node is a single purpose and executable program.
- Nodes are assembled into a graph.

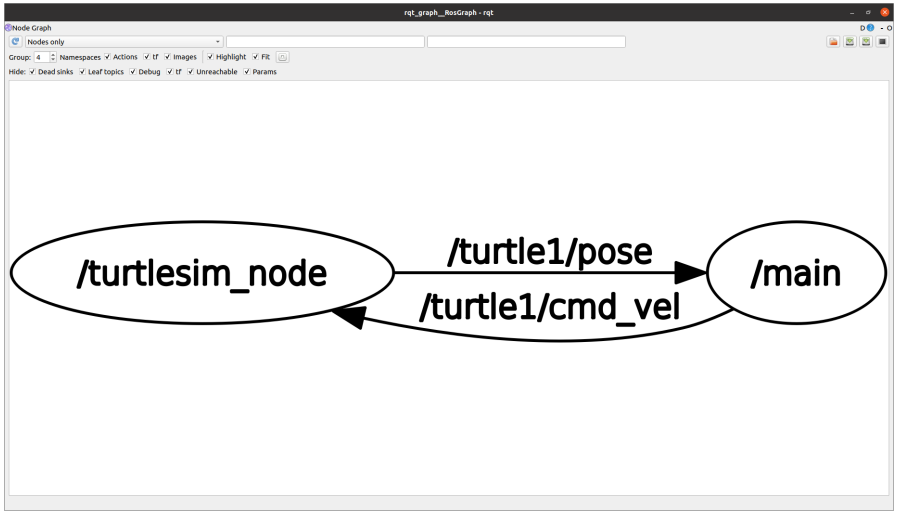




Node

Visualize Nodes

```
Ubuntu:~$ rqt_graph
```



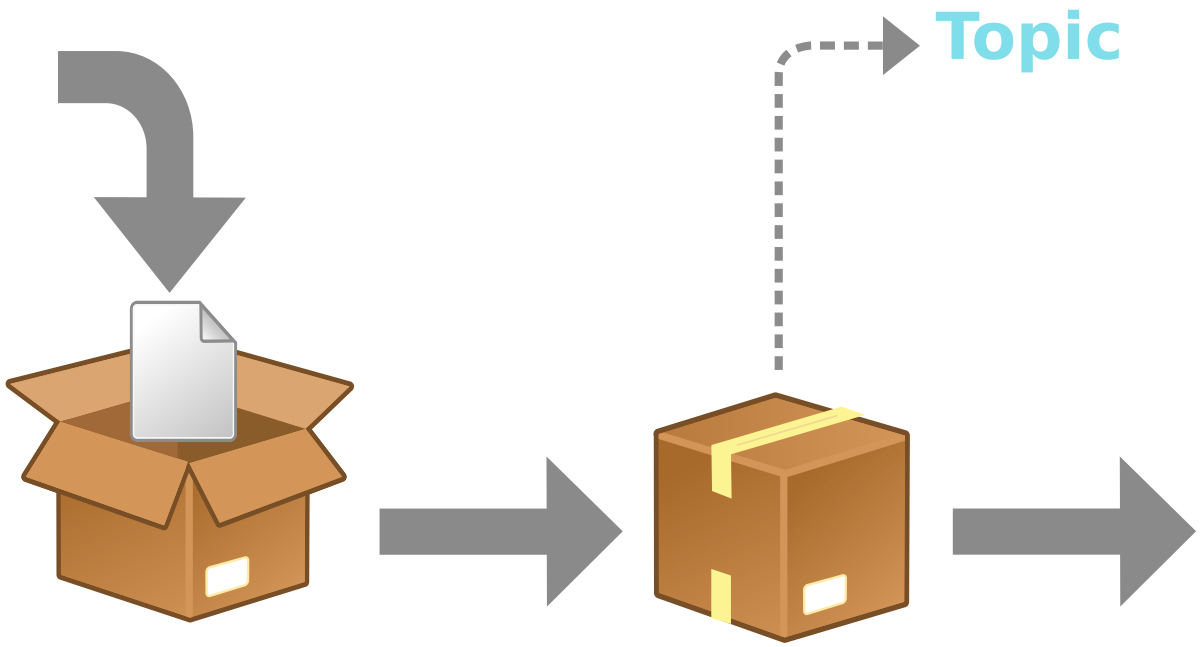
View Nodes list

```
Ubuntu:~$ rosnode list
```

```
/main
/rosout
/turtlesim_node
```

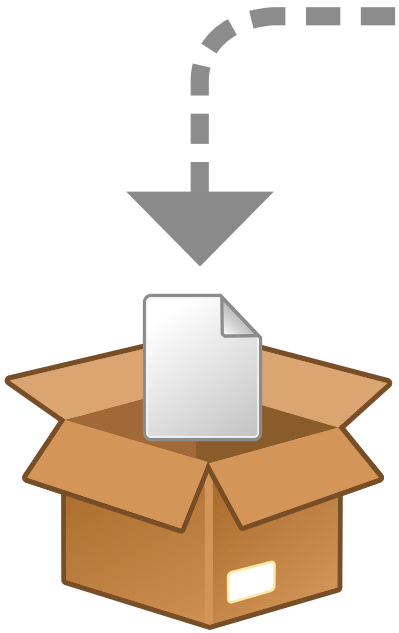
Topic

Each message in ROS in transported using named a *BOX* called Topic.



Message

Nodes communicate with each other using messages. Messages are simply a data structure containing the typed field, which can hold a set of data, and that can be sent to another node. There are standard primitive types (integer, floating point, Boolean, and so on) and these are supported by ROS messages. We can also build our own message types using these standard types.



Message

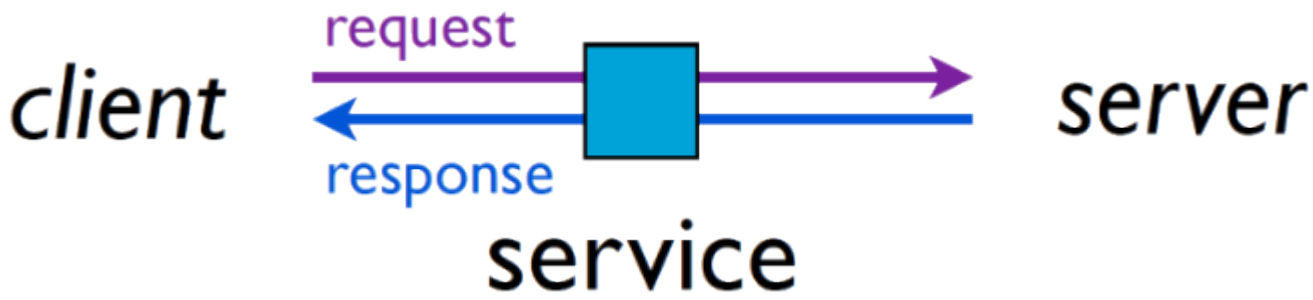
int8
uint8
int16
uint16
int32
uint32
int64
uint64
float32
float64
string
time

```
int32 number
string name
float32 speed
```

Service

The ROS services are a type request/response communication between ROS nodes.

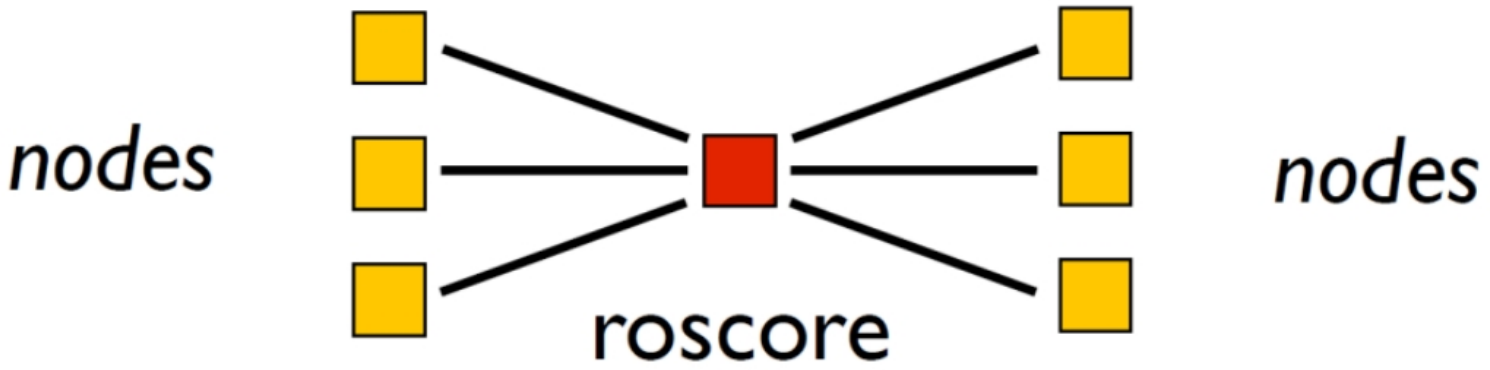
One node will send a request and wait until it gets a response from the other.




```
#Request message type
string str
---
#Response message type
string str
```

Master node

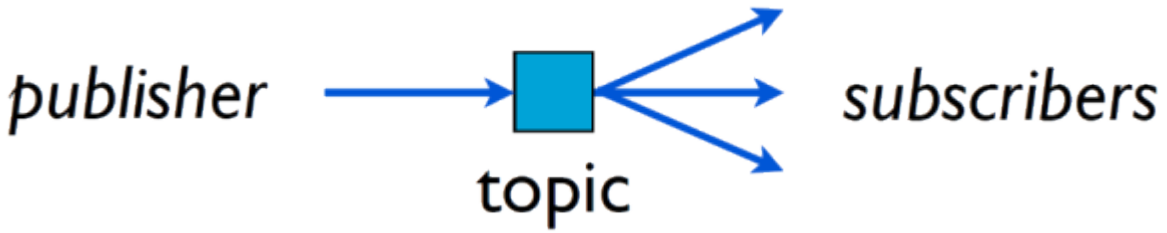
Roscore is the master that manages the connection information for communication among nodes, and is an essential element that must be the first to be launched to use ROS. The ROS master is launched by the ‘roscore’ command. The master registers node information such as names, topics and service names, message types and when there is a request this information is passed to other nodes. After the launch of ‘roscore’, ‘rosout’ is executed as well, which is used to record ROS standard output logs.



Publisher-Subscriber

Publisher: When a node sends a message through a topic, then we can say the node is publishing a topic.

Subscriber: When a node receives a message through a topic, then we can say that the node is subscribing to a topic



References

- Quigley, Morgan & Conley, Ken & Gerkey, Brian & Faust, Josh & Foote, Tully & Leibs, Jeremy & Wheeler, Rob & Ng, Andrew. (2009). ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software. 3.
- Lentin Joseph and Jonathan Cacace. 2018. *Mastering ROS for Robotics Programming - Second Edition: Design, build, and simulate complex robots using the Robot Operating System* (2nd. ed.). Packt Publishing.