ROS SMP - Week 1

IMPORTANCE OF ROS

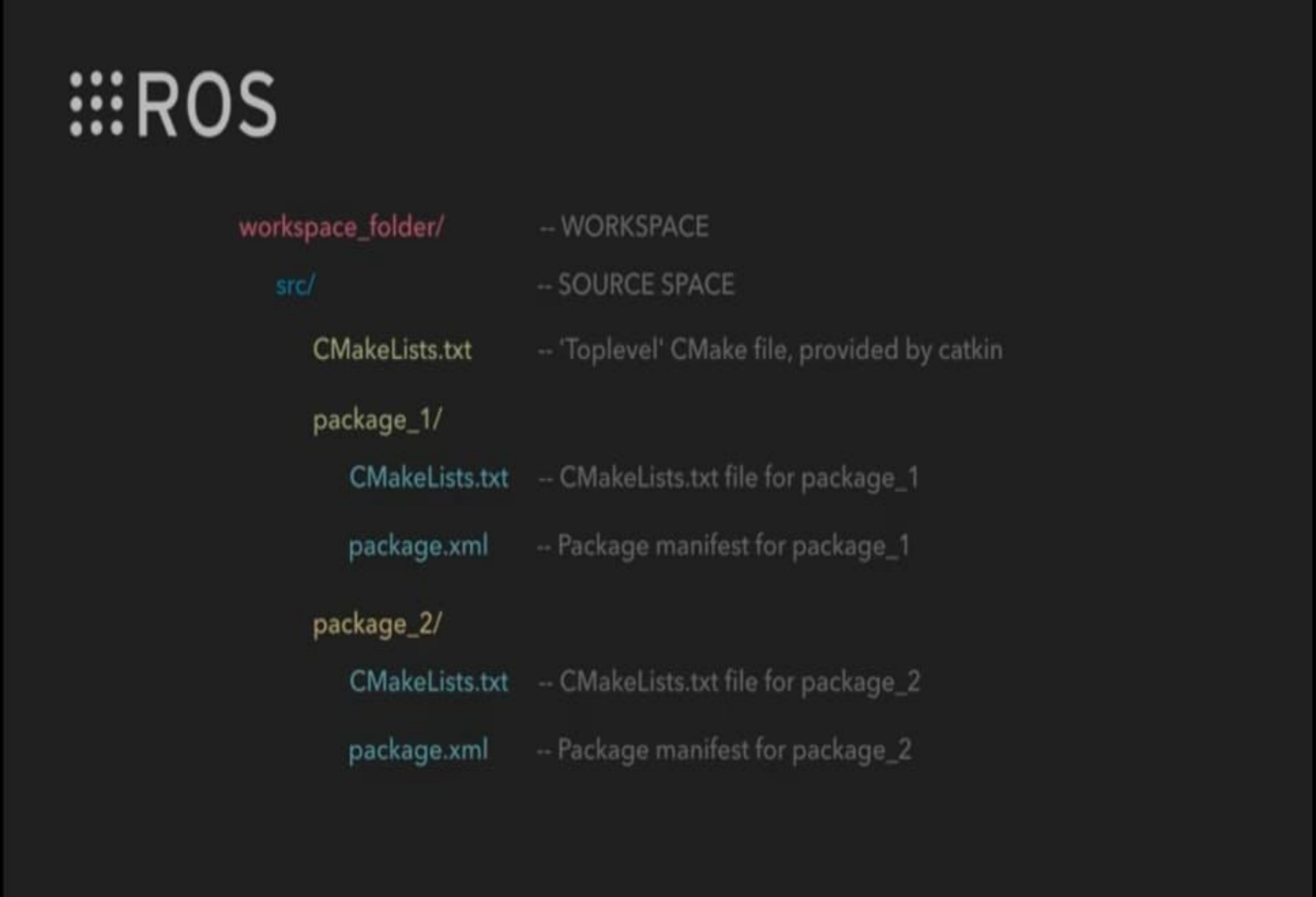
- ROS = Robotic Operating System.
- One of the most important and interesting topics in field of robotics.
- Helps build robots in an “easier” way.
- Not a traditional OS.

INDEX

- ROS Packages
- ROS Nodes
- rosnode command
- ROS Topics
- rostopic command
- Launch file
- roslaunch command
- rqt_graph.

ROS Packages

An auto generated directory of files that ROS needs to run



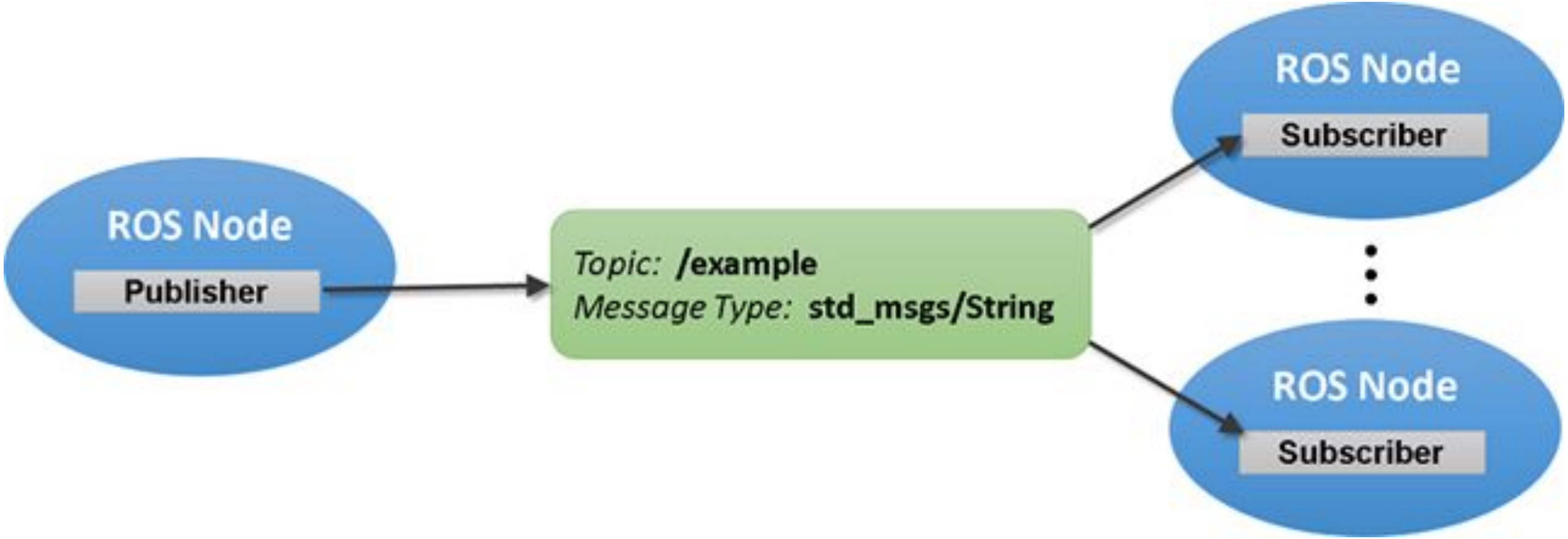
ROS

```
workspace_folder/        -- WORKSPACE
  src/                   -- SOURCE SPACE
    CMakeLists.txt       -- 'Toplevel' CMake file, provided by catkin
    package_1/
      CMakeLists.txt     -- CMakeLists.txt file for package_1
      package.xml        -- Package manifest for package_1
    package_2/
      CMakeLists.txt     -- CMakeLists.txt file for package_2
      package.xml        -- Package manifest for package_2
```

ROS NODES

Basics

- ROS nodes are the building blocks of any ROS application
- A ROS node is a piece of software /executable that uses ROS to communicate with other ROS nodes
- For example, if you have a wall-following robot then one ROS Node could get distance sensor values and another node can control the motors of the robot. So, these two nodes will communicate with each other in order to move the robot.
- Single ROS package can have any no of nodes.
- ROS nodes can be launched using different programming languages but popularly used ones are python and C++.
- Communication can happen in 3 ways
  - Ros topics
  - Ros services
  - Ros actions



ROS Node
Publisher
Topic: /example
Message Type: std_msgs/String
ROS Node
Subscriber
ROS Node
Subscriber

Launching a core node (ROSCORE)

- It is master node you must have this running in order for ros nodes to communicate.it can be launched using ***roscore*** command
- It stores all the topics and service registration information for ros nodes
- You can say, communication is established between nodes by the ROS Master. So, without ROS Master running ROS Nodes can not communicate with each other.
- roscore will start up:
  - a ROS Master
  - a ROS parameter server
  - a rosout logging node

How to run a ros node?

Rosrun allows to run an executable in a package without knowing its location

```
$ rosrun <package_name>  <executable_file_name>
```

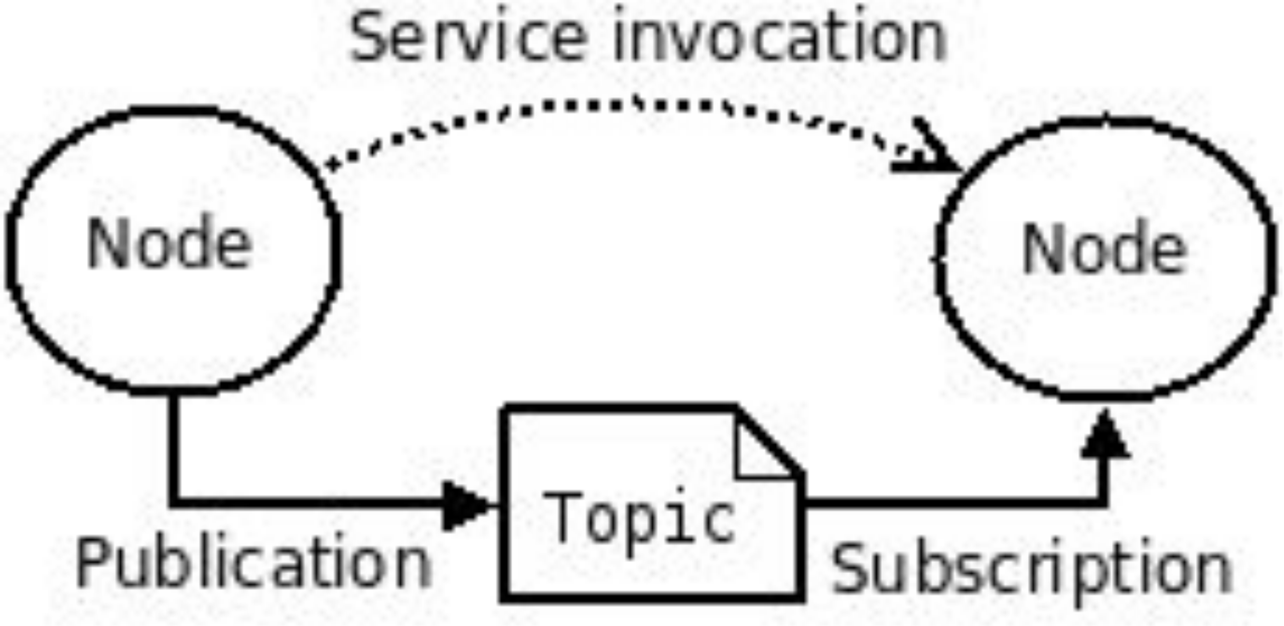
rosrun with turtlesim_node

```
$ rosrun turtlesim turtlesim_node
$ rosrun turtlesim turtle_teleop_key
```

ROS TOPICS

Basics

- A node sends out a message by publishing it to a given topic
- Another node requiring a certain type of data must subscribe to the appropriate topic
- Single node can publish and/or subscribe to multiple topics
- Publish/subscribe model is flexible (many to many , one-way transport)
- Publisher and Subscriber Nodes will exchange ROS Messages over a ROS Topic


Service invocation
Node
Node
Topic
Publication
Subscription

ROS TOPIC COMMANDS

$ **rostopic** list

Returns a list of topics currently subscribed to and published

$ **rostopic** type [topic_name]

shows the type of the message

$ **rosmsg** show [message_type]

eg. rosmsg show geometry_msgs/Twist --> shows the data structure of the message

$ **rostopic** info [topic_name]

$ **rostopic** pub [topic] [msg-type] [args]

ROS message

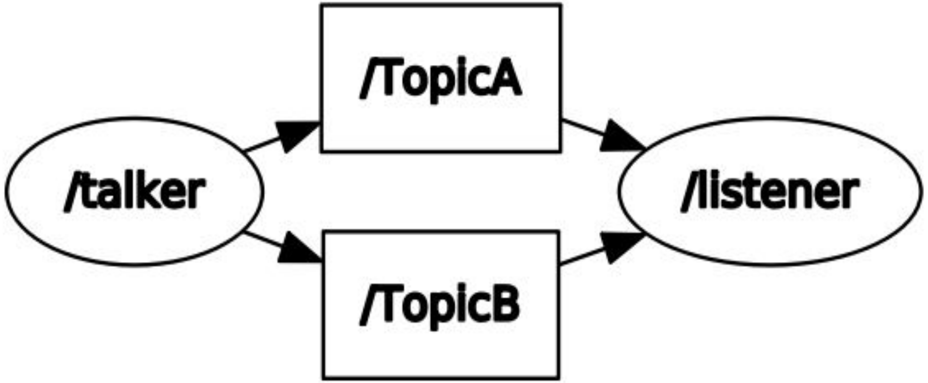
- Messages are just simple text files with a field type and field name per line.
- They are stored in the msg directory of your package.

Create a custom message file(*.msg) in msg directory and add the following to it

  - <msg_type> identifier_name   (format for a typical message file)
  - Add the dependencies in package.xml file
  - Edit the CMakelist.txt file

RQT Graph

- Gives a graphical interpretation of all the nodes , topics, services, actions
- Handy tool while debugging.
- $ rqt_graph   (terminal command- gives the graph)



Launch file

- Launch file can launch multiple ROS nodes, shell scripts and can fetch parameters from the parameter server with a single command in a single terminal tab.
- The idea is to mention all the nodes that you want to run, all the config file that you want to load etc in a single file which you can run using ***roslaunch*** command.

Usage : roslaunch <package_name> file.launch

Rosrun overview

Launching nodes manually

- Run roscore (on terminal 1)
- Run your custom nodes on different terminals. (one node on each tab)

(assume how many terminals to launch to have a industrial robot running , thanks to launch file :)

Launching nodes using launch file

- Run roslaunch with necessary parameters on a single terminal tab

THANK YOU ;)