R O S

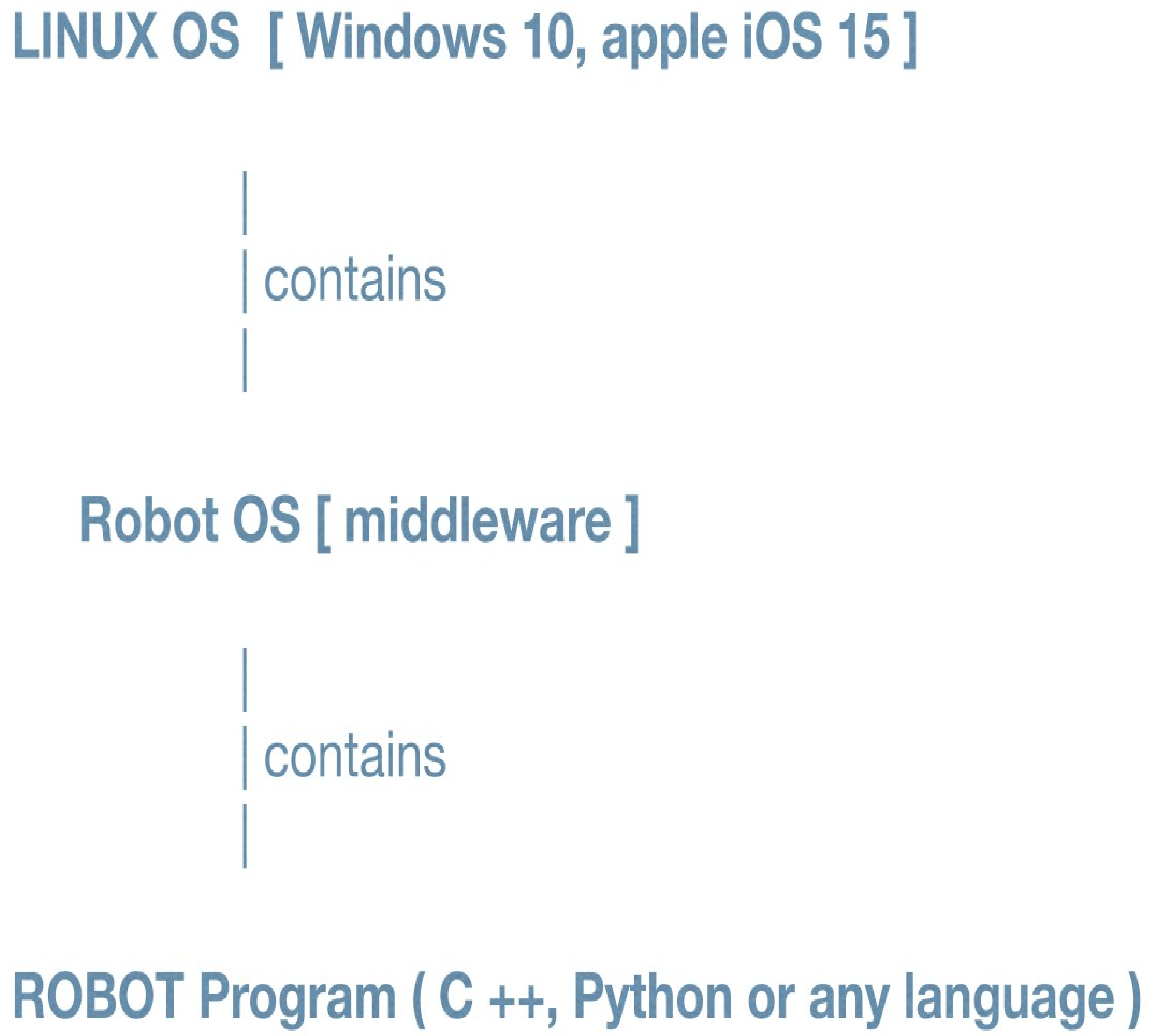
Robot Operating System Architecture

SOFTWARE Architecture

Like a building Architecture, we have software architecture

Computer OS - Linux OS
Middleware - Robot OS
Robot Programs— C++, Python

So, we need Linux terminal basic commands and ROS commands before we develop our program for Robot in ROS !!!



LINUX OS [Windows 10, apple iOS 15]
contains
Robot OS [middleware]
contains
ROBOT Program (C ++, Python or any language)

ROS PACKAGES !

ROS uses “ Software Packages “ to organise its Program .

“Packages” are downloadable ! from open source GITHUB

———> Standard three clause BSD license

Packages has three main components

1. Source file/Scripts 3. CMake list 4. Package xml

Src file/Scripts

Src file is nothing but a Source file . Basically, It contains all the C++ and python nodes (programs) for execution .

Note : Developers develop new nodes and put them here for execution . We can use rosrun or roslaunch command in Linux terminal for execution !!!

CMake List (COMPILATION)

Set of Rules or Instructions for making compilation

$add executables {package&node_name}

Dependency [target libraries(Cmake_library)]

XML file

Package Info (author name- email - licenses) and **dependencies**

DESCRIPTION OF THE PACAKAGE

WHEN YOU RELEASE YOUR TO ROS COMMUNITY, MAKE SURE ABOUT XML FILES...

Nodes - c++ or python code in Ros

How nodes communicate??? Topics and Services

Node is basically an end program stored in Source file . It can be either publisher or subscriber ....

topics

NODE 1 <-------> NODE 2

Robot Program

```
int final_position ( )
{.
 x = x+y ;
return x ;
}
```

Publish x in topic

Robot Program

```
int robot_action ( )
{
 if (x = 50)
   { turn Right }
 }
```

Subscribe to topic to get X value

topics

NODE 1 < — — — — — - > NODE 2

Robot Program	Robot Program

```
int final_position ( )
{.
 x = x+y ;
return x ;
}
```

```
int robot_action ( )
{
 if (x = 50)
  { turn Right }
 }
```

Publish x in topic

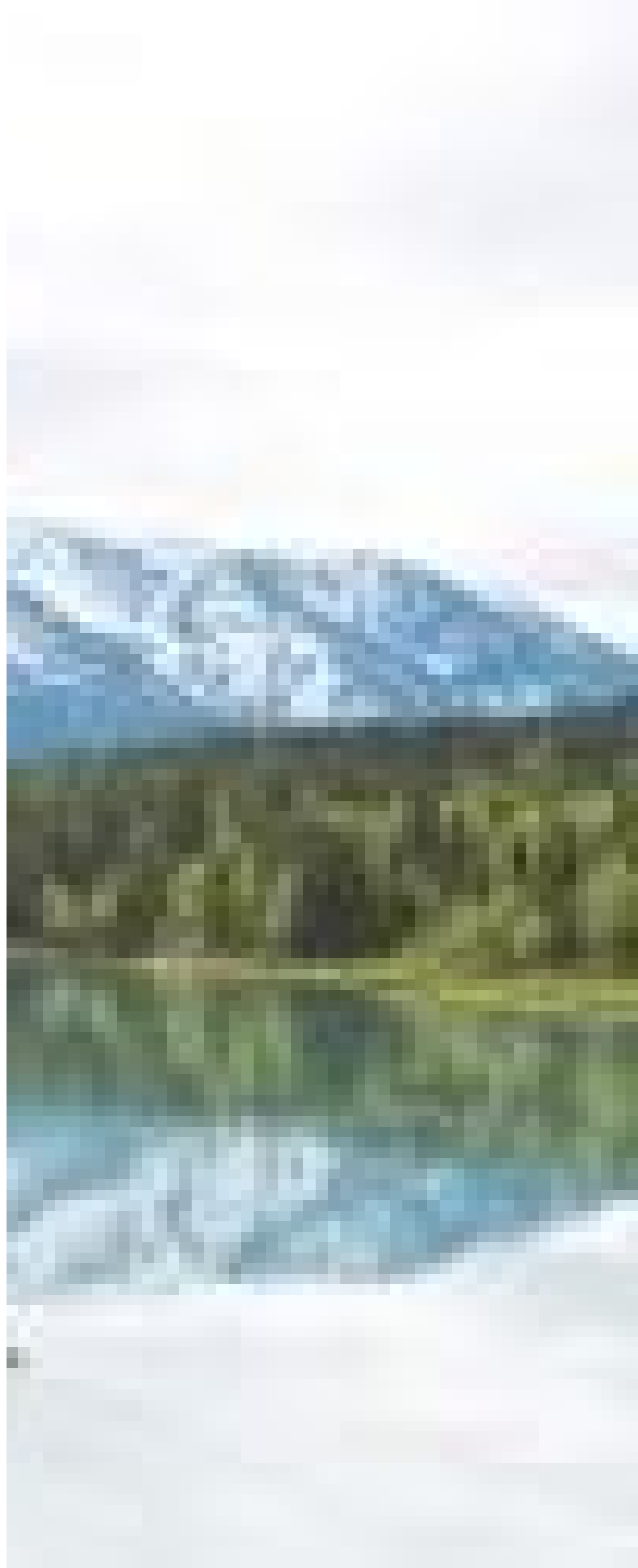
Subscribe to topic
to get X value

topics ______________

NODE 1 ___ messages ________ NODE 2

topics ______________

NODE 1 ___ x=20 or any value ______ NODE 2
Publisher Subscriber



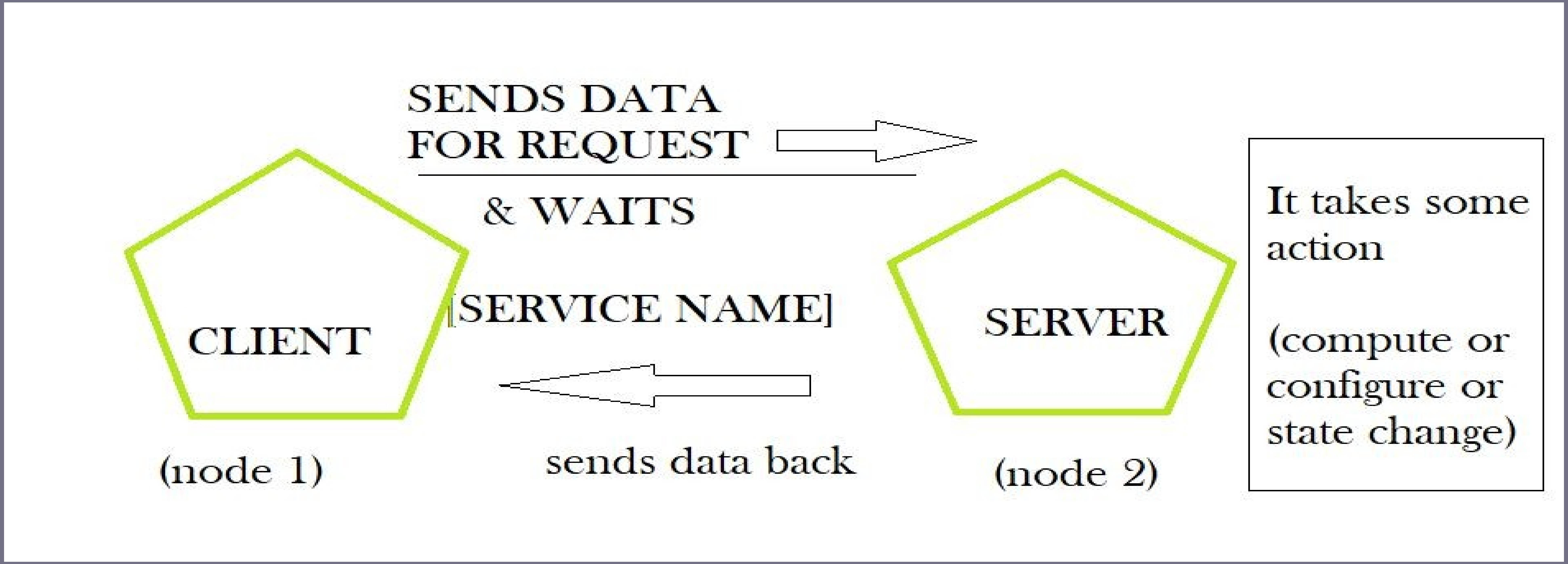
NODES

Nodes			Topics
Publisher Node	Many nodes publish to same topic	Publish to many topics at a time	It’s a channel
Subscriber Node	Many nodes subscribe to same topic		It allows messages to pass through
Publisher and Subscriber			Messages are mostly function parameters

Service Communication

SYNCHRONOUS COMMUNICATION

IT IS BASED ON REQUEST AND RESPONSE COMMUNICATION ...



SOME LINUX TERMINAL COMMANDS.

Making directory or file — $mkdir directory_name ; $ mkfile file_name

Removing file - $rm file_name

Removing Directory (folder) - $rm -rf directory_name

Cloning a Software Repository (software packages)

$gitclone url-copy and paste from GITHUB (this is also called copying it from Source)

sudo apt-get install package1 package 2 ., ;sudo autoremove package

Linux commands

```
Chmod 777 file_name

ls-l |grep file_name

Sudo chmod +x file_name
```

----- > giving permission

ROS COMMANDS

roscore - Runs Ros master

rosrun ros_package ros_node_name

rosrun rqt_graph rqt_graph

rostopic -h , rostopic echo topic_name ,
rostopic - v (details about the topic)

rosmsg show message_typename

rostopic type topic_name —-> you ll get msg
type …. Refer rqt graph for topic names !!

rostopic pub-1 topic_name msg_type“ arg “

Launching Single and multiple Node

$ roslaunch < package name > < launch
file name >

$ rosrun < package name > < node >

Launch file :

<launch>

<package =   Type= Node name =
Output =   >

<\launch >

Creating a Catkin Work Space/ ROS Package

I want to create a package then we need to work with a specific space

catkin_ws

Building a workspace

Step one

Go inside the source file in the Catkin space

Step two

In the workspace build the catkin $catkin_ws build

Note: even with no packages we can build the Catkin space

Building a catkin space

Step three

$source devel / setup.bash

This will add a path for a current terminal

All other terminal, use the following comments

$cd~ $ nano.bashrc

$source ~/catkin_ws/devel/setup.bash

$ctrl + x

$ y

Make sure you create a package in the source file

Step four

catkin_create_pkg < package name > rospy roscpp

Checking —— rospack list

rospack/ grep my package

roscd rospackage name

Adding Sensors and Models

Urdf models Tags are used to add a model

Publishing and Subscribing Node

Wikipedia Ros ! + YouTube series

Construct

ROS development Studio