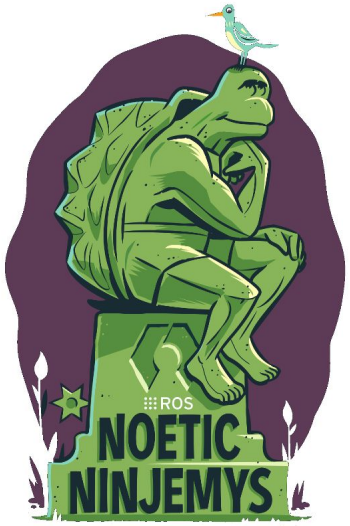
آشنایی با ROS

جلسه‌ی دوم : پیکربندی و آشنایی با مفاهیم



balazadeh.elnaz@gmail.com

تاریخچه ی مختصری از ROS

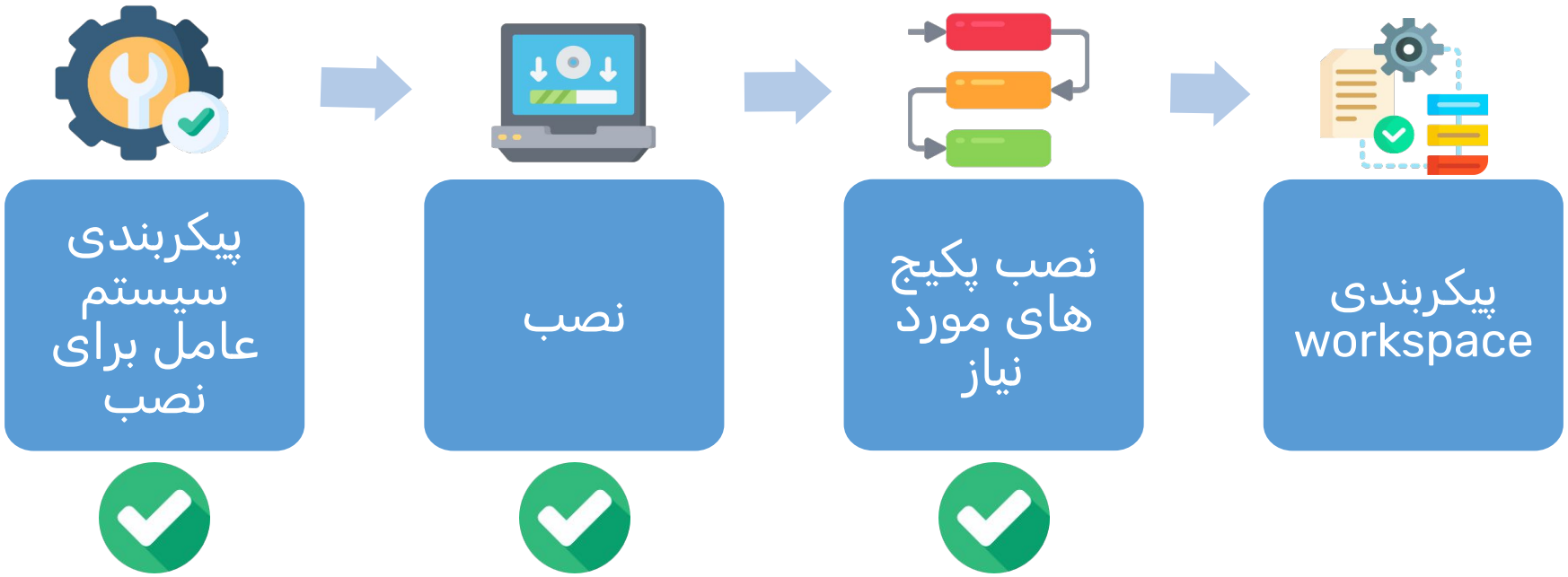
ROS

- آشنایی با مفاهیم
- پیاده سازی و کار با مفاهیم

بخش اول

بخش دوم

پیکربندی Workspace

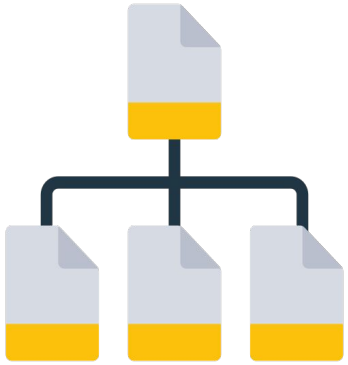


پیکربندی
workspace

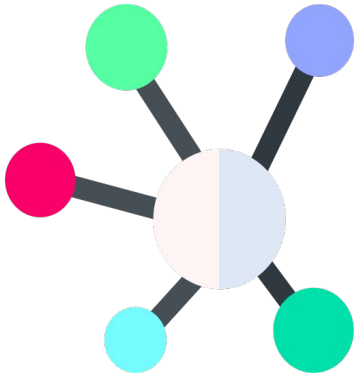
https://wiki.ros.org/catkin/Tutorials/create_a_workspace

سطوح concept های ROS

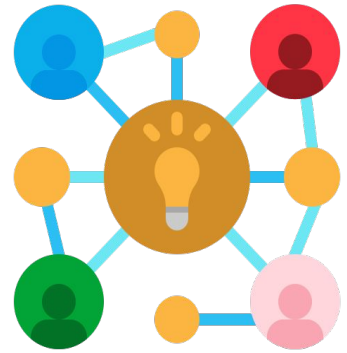
ROS File system



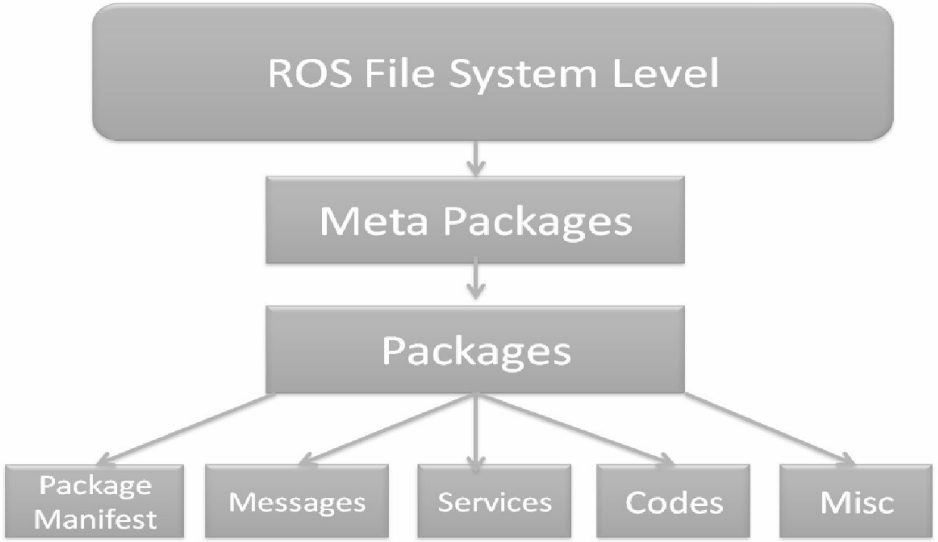
ROS Computation Graph



ROS Community

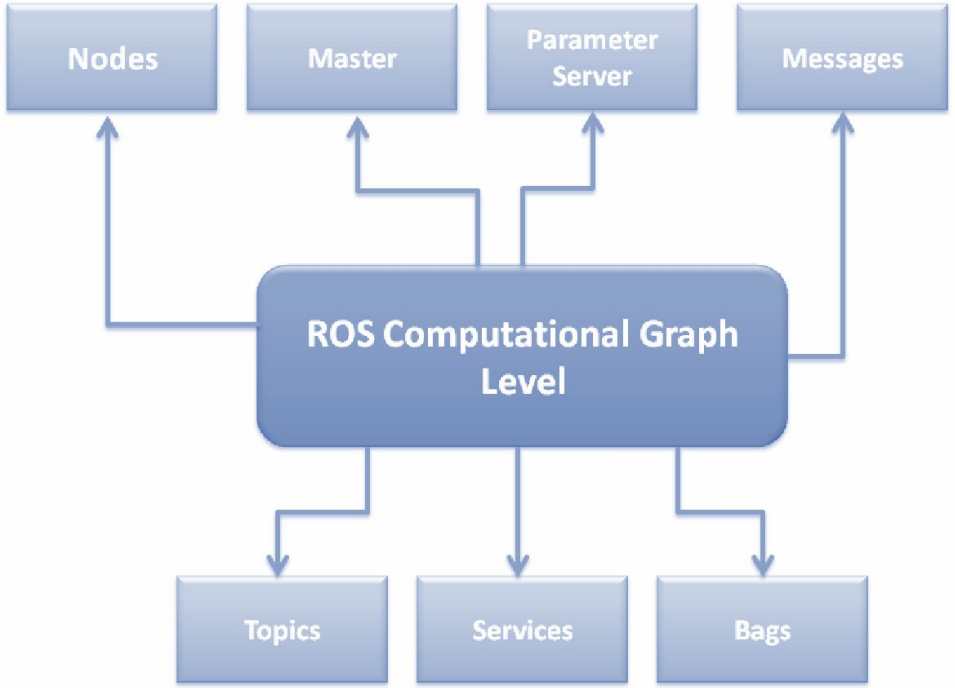


Concept ها در سطح File system



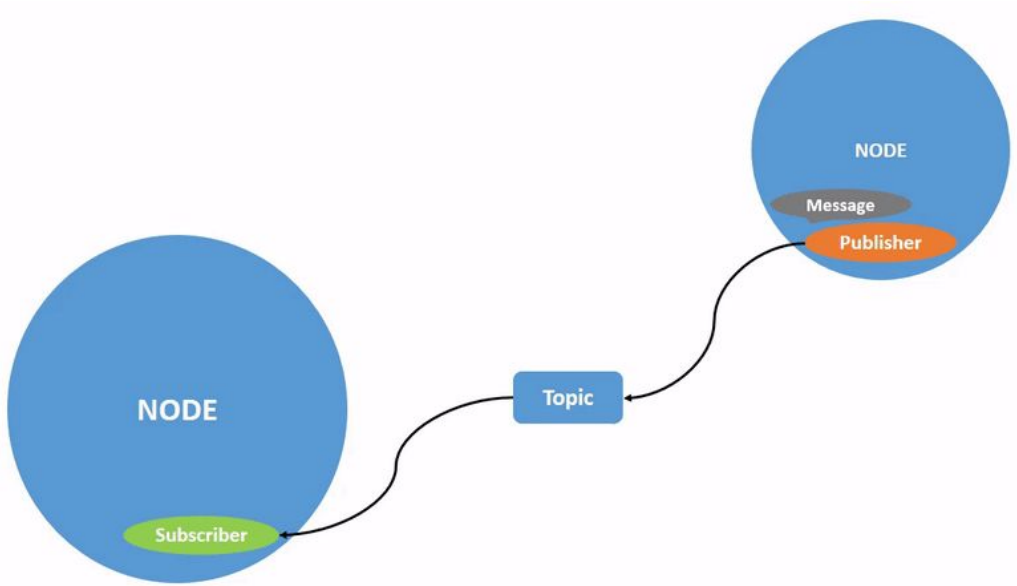
Joseph, L. and Cacace, J., 2018. *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd.

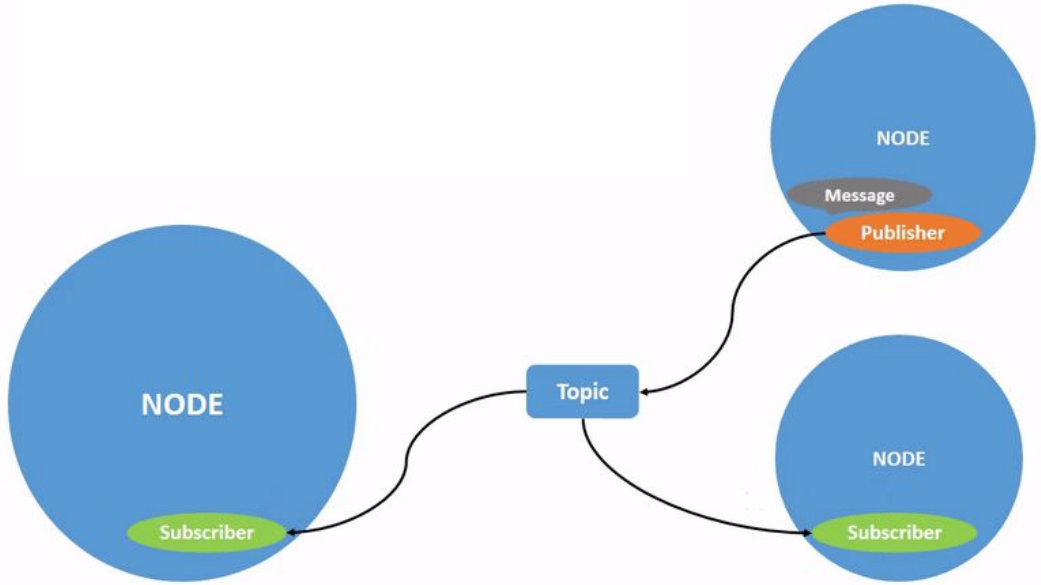
Concept ها در سطح Computation Graph



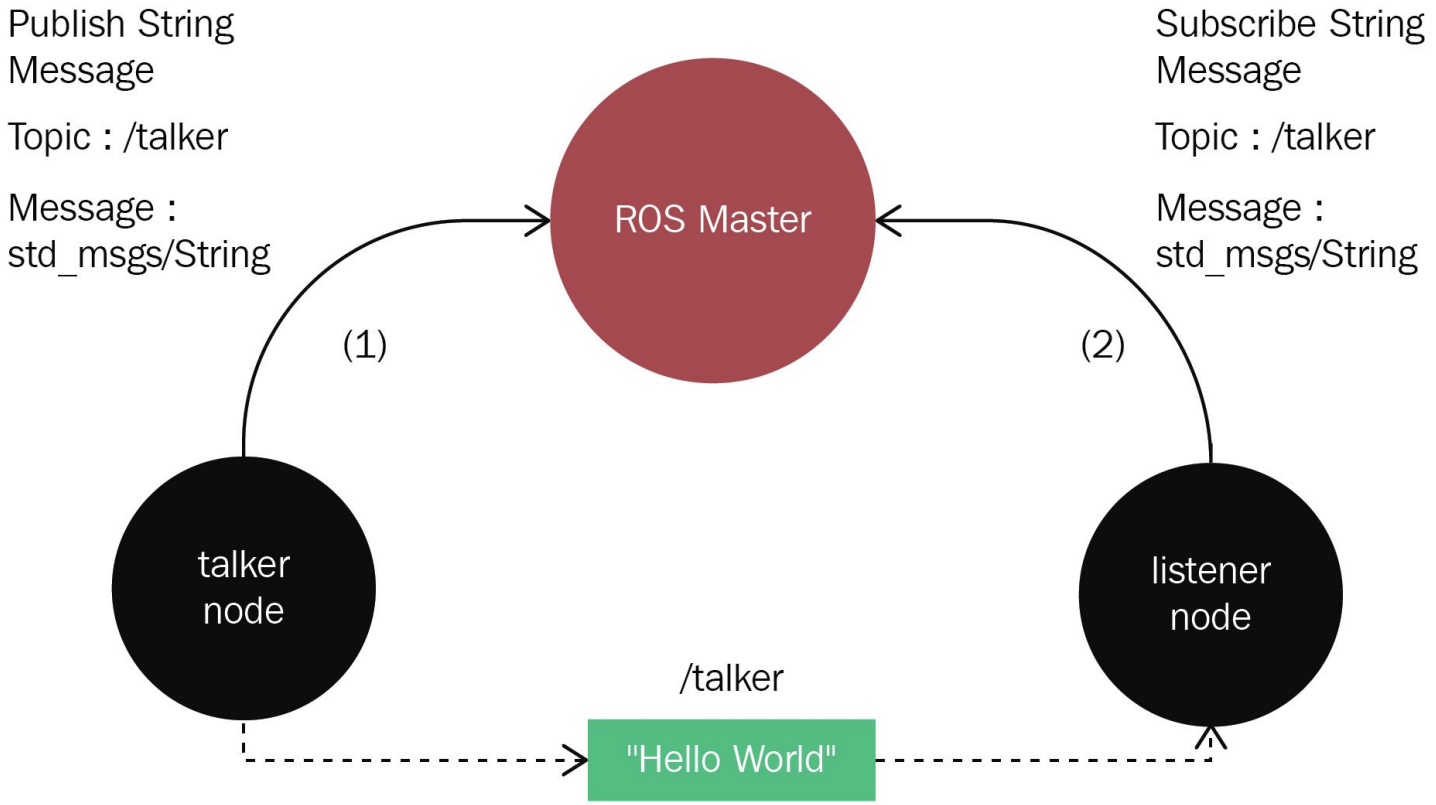
Joseph, L. and Cacace, J., 2018. *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd.

Node .1

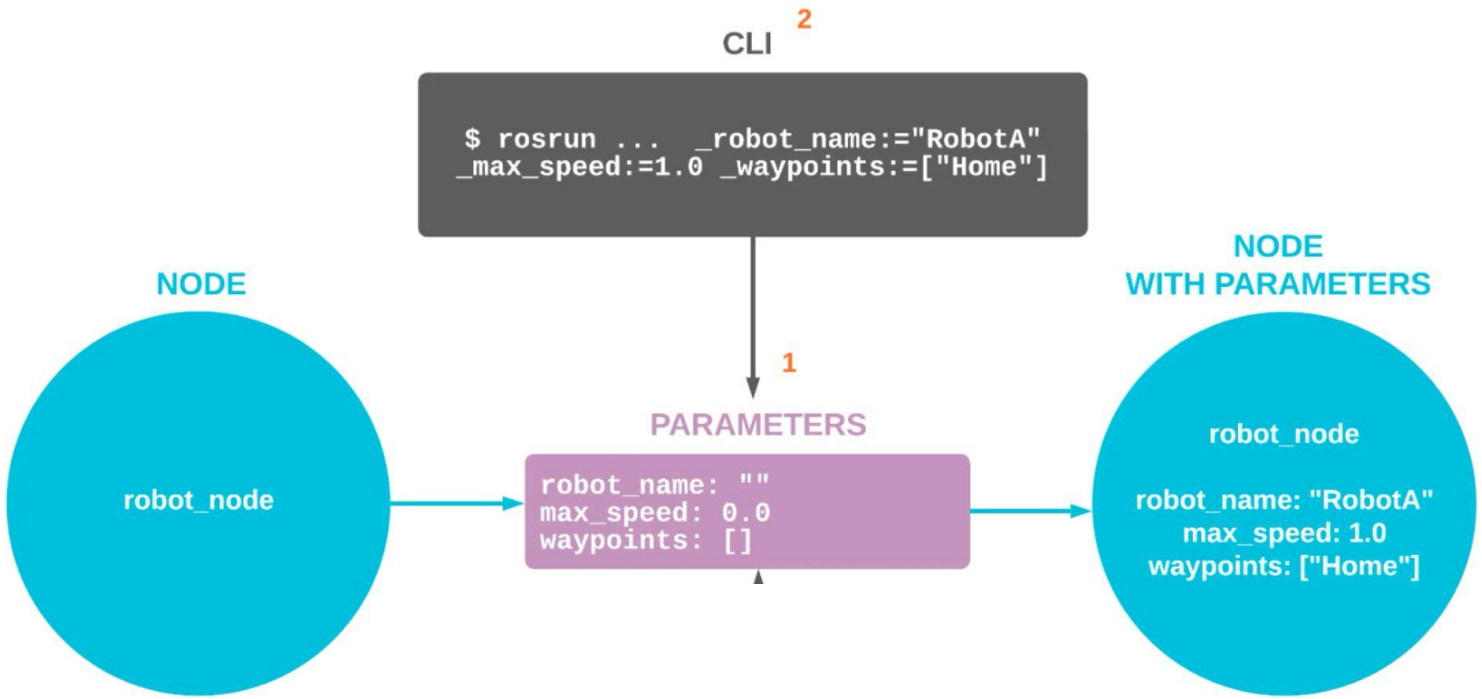

NODE
Message
Publisher
Topic
NODE
Subscriber


NODE
Message
Publisher
NODE
Topic
NODE
Subscriber

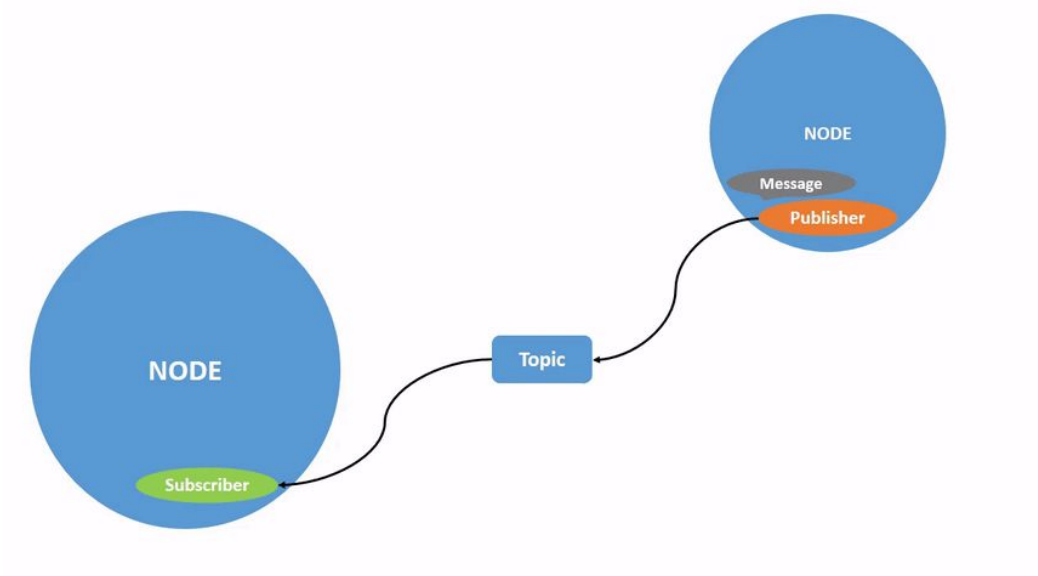
Master .2

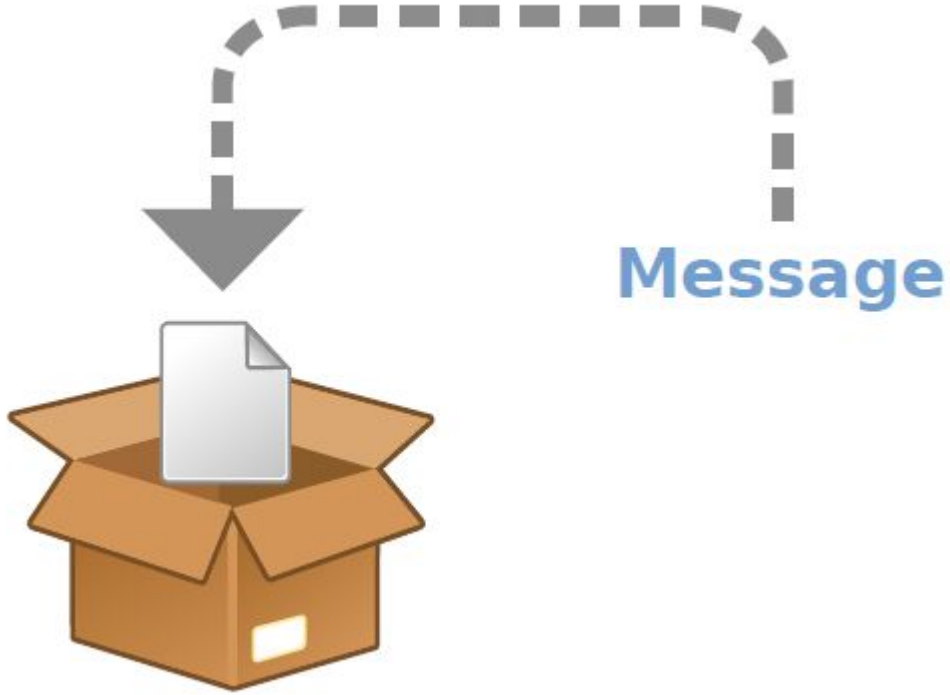

Publish String Message
Topic : /talker
Message : std_msgs/String
Subscribe String Message
Topic : /talker
Message : std_msgs/String
ROS Master
(1)
(2)
talker node
listener node
/talker
"Hello World"

Parameters .3

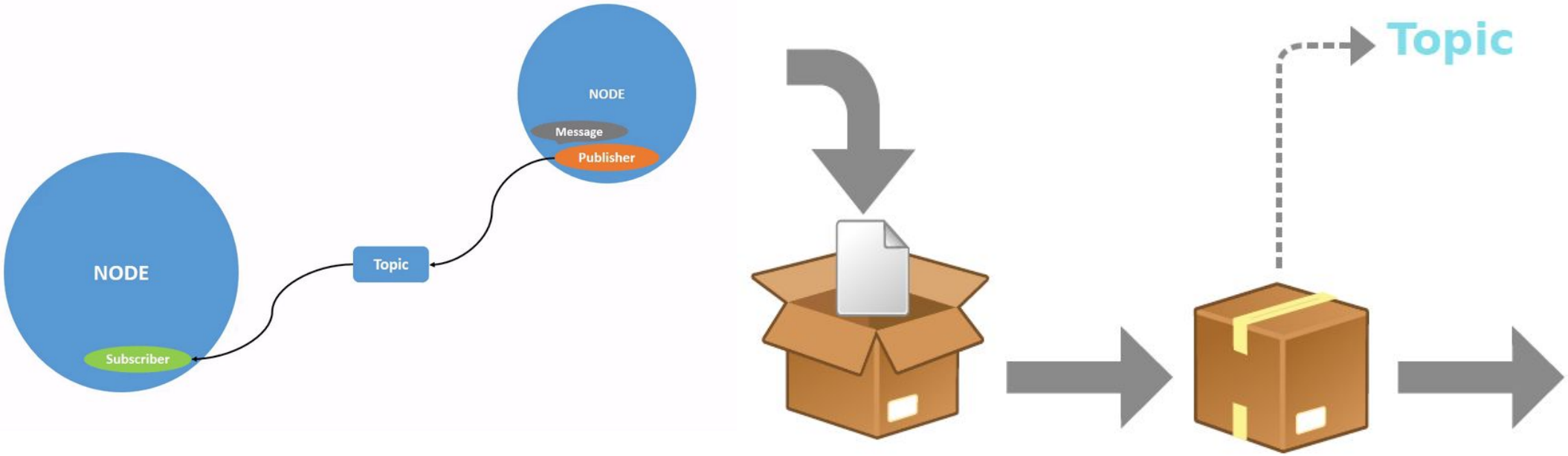


Messages .4

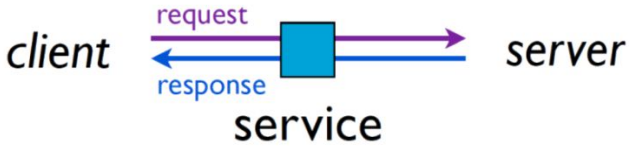


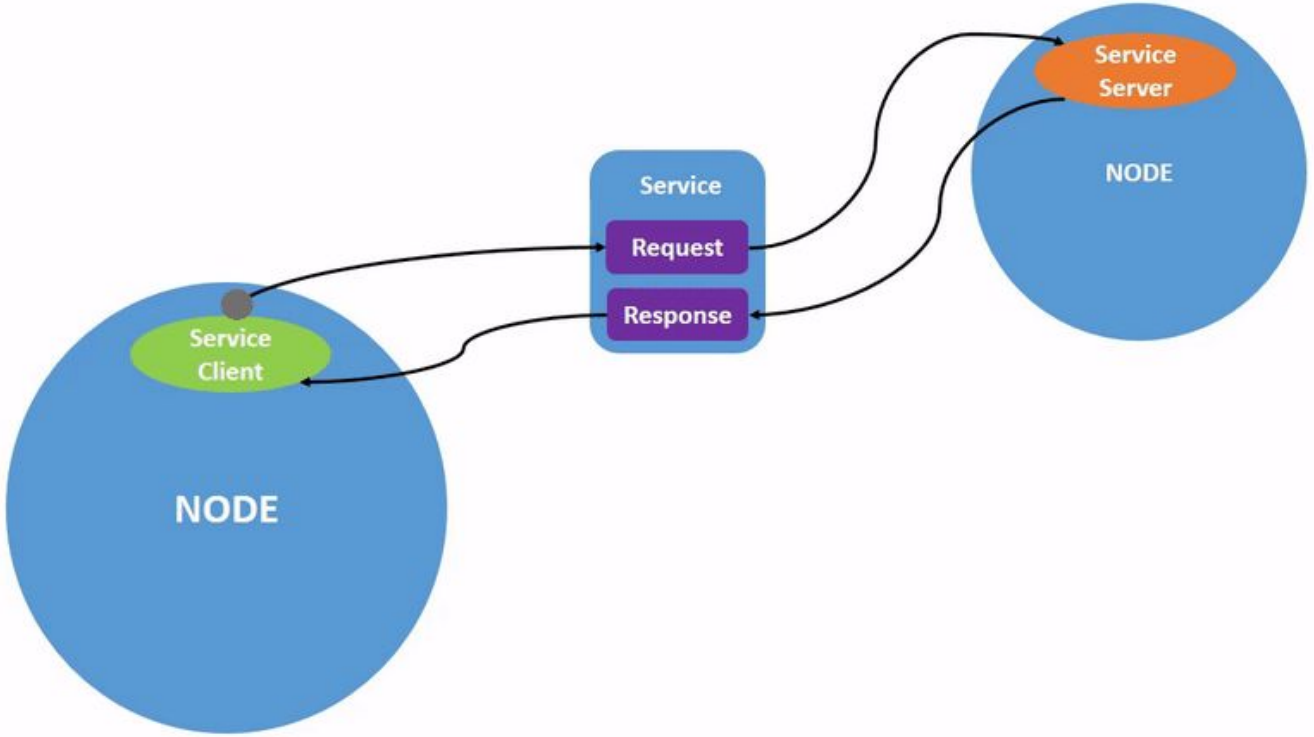


Topic .5

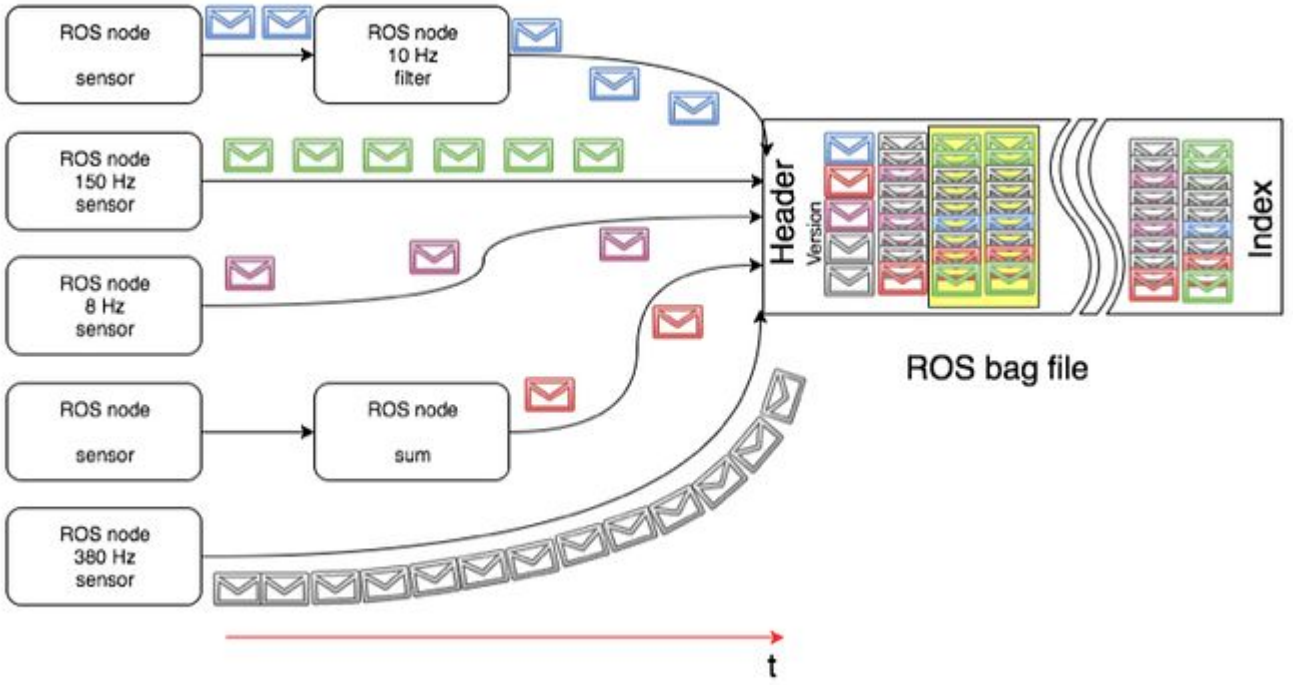


Service .6

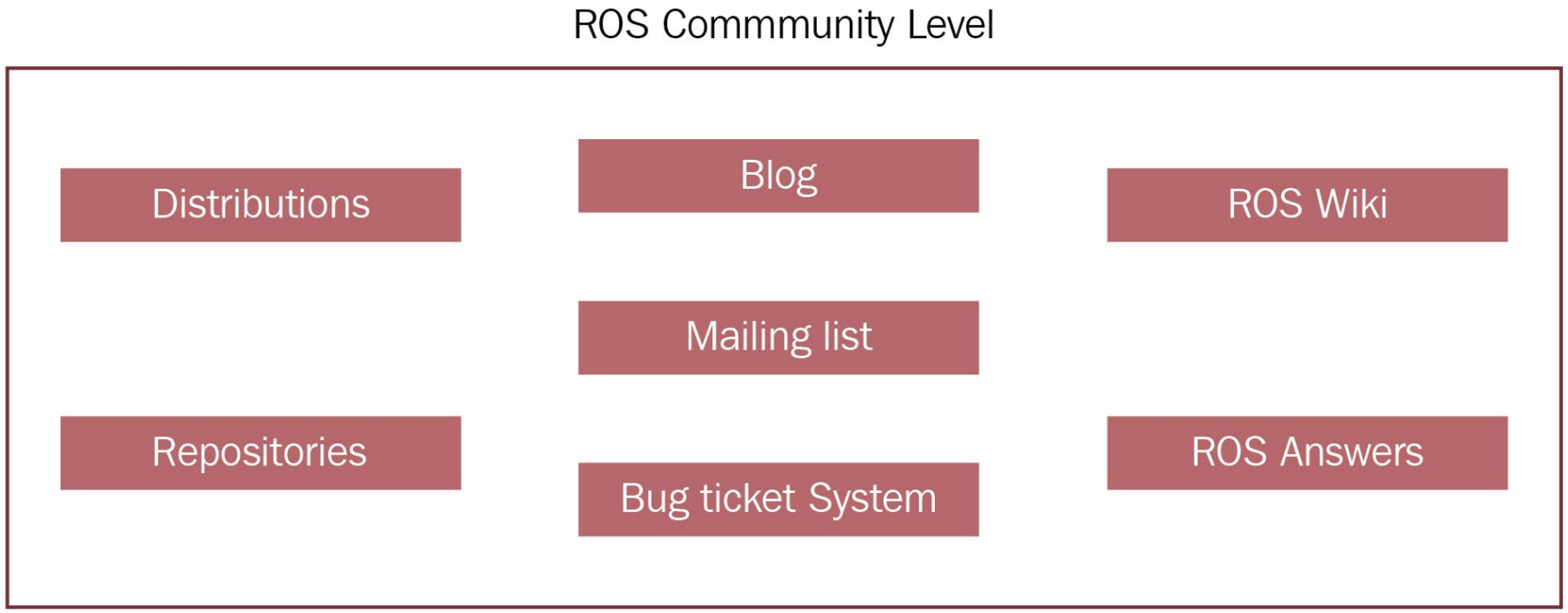

client
request
response
server
service


Service
Server
NODE
Service
Request
Response
Service
Client
NODE

Bag .7


ROS node
sensor
ROS node
10 Hz
filter
ROS node
150 Hz
sensor
ROS node
8 Hz
sensor
ROS node
sensor
ROS node
sum
ROS node
380 Hz
sensor
Header
Version
Index
ROS bag file
t

Concept ها در سطح ROS Community



پیاده سازی و کار با Conceptها

قدم اول: ساخت اولین پکیج

```
$ cd ~/catkin_ws/src

$ catkin_create_pkg <package_name> [depend1] [depend2] [depend3]
```

ROS

پیاده سازی و کار با Conceptها

Publisher node

```
 1 #!/usr/bin/env python3
 2 import rospy
 3 from std_msgs.msg import String
 4
 5 def talker():
 6
 7     pub = rospy.Publisher('my_topic', String, queue_size=10)
 8     rospy.init_node('Pub_node', anonymous=True)
 9     rate = rospy.Rate(10) # 10hz
10     while not rospy.is_shutdown():
11         hello_str = "hello world " + str(rospy.get_time())
12         rospy.loginfo(hello_str)
13         pub.publish(hello_str)
14         rate.sleep()
15
16 if __name__ == '__main__':
17     try:
18         talker()
19     except rospy.ROSInterruptException:
20         pass
```

ROS

پیاده سازی و کار با Conceptها

اجرای Publisher Node

```
# rosrun [package_name] [node_name]
$ rosrun my_package pub.py
```

ROS

پیاده سازی و کار با Conceptها

Subscriber node

```
 1 #!/usr/bin/env python3
 2 import rospy
 3 from std_msgs.msg import String
 4
 5 def callback(data):
 6     rospy.loginfo("I heard " + data.data)
 7
 8 def listener():
 9
10     # In ROS, nodes are uniquely named. If two nodes with the same
11     # name are launched, the previous one is kicked off. The
12     # anonymous=True flag means that rospy will choose a unique
13     # name for our 'listener' node so that multiple listeners can
14     # run simultaneously.
15     rospy.init_node('Sub_node', anonymous=True)
16
17     rospy.Subscriber("my_topic", String, callback)
18
19     # spin() simply keeps python from exiting until this node is stopped
20     rospy.spin()
21
22 if __name__ == '__main__':
23     listener()
```

ROS

پیاده سازی و کار با Conceptها

اجرای Subscriber node

```
# rosrun [package_name] [node_name]
$ rosrun my_package sub.py
```

ROS

ممنون از توجه شما