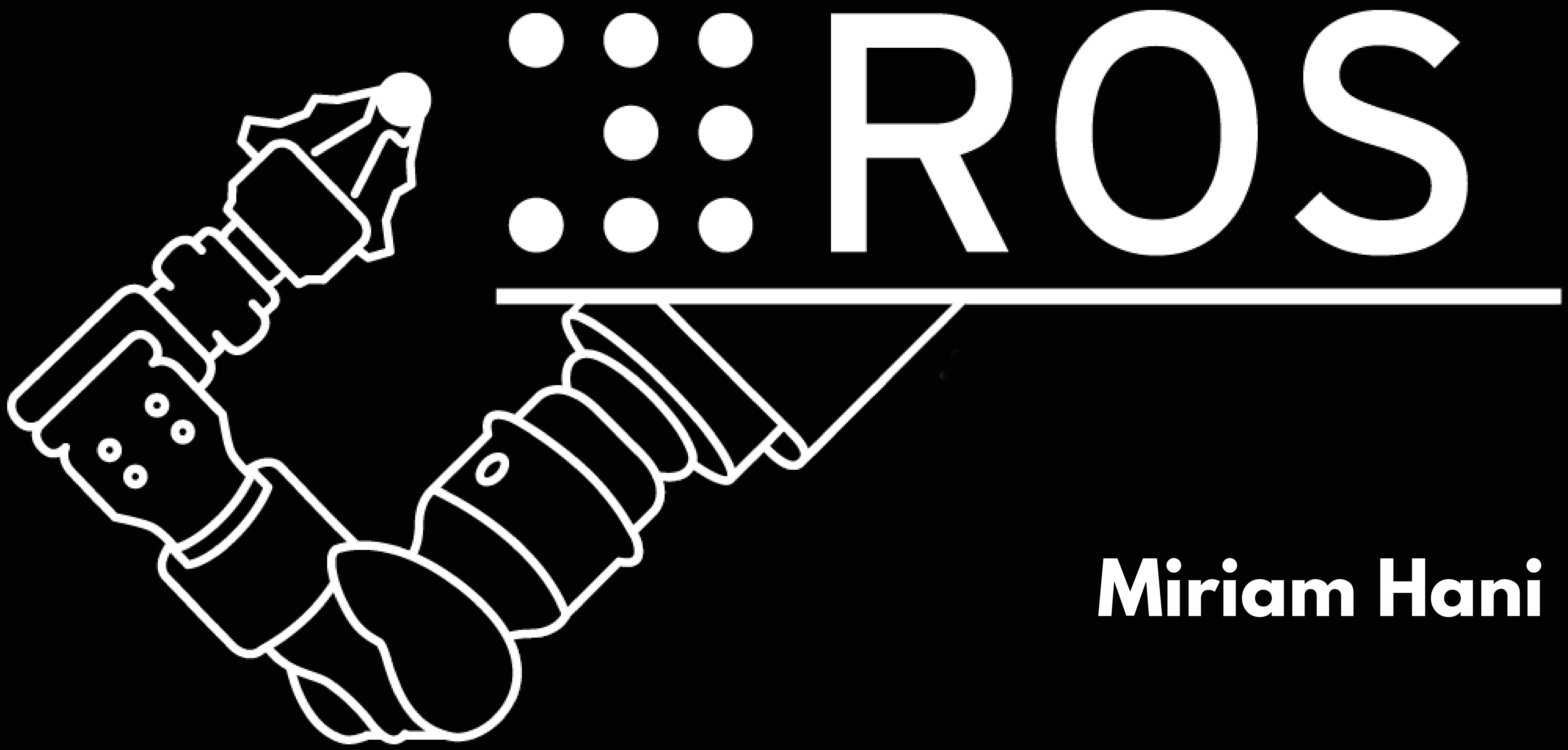
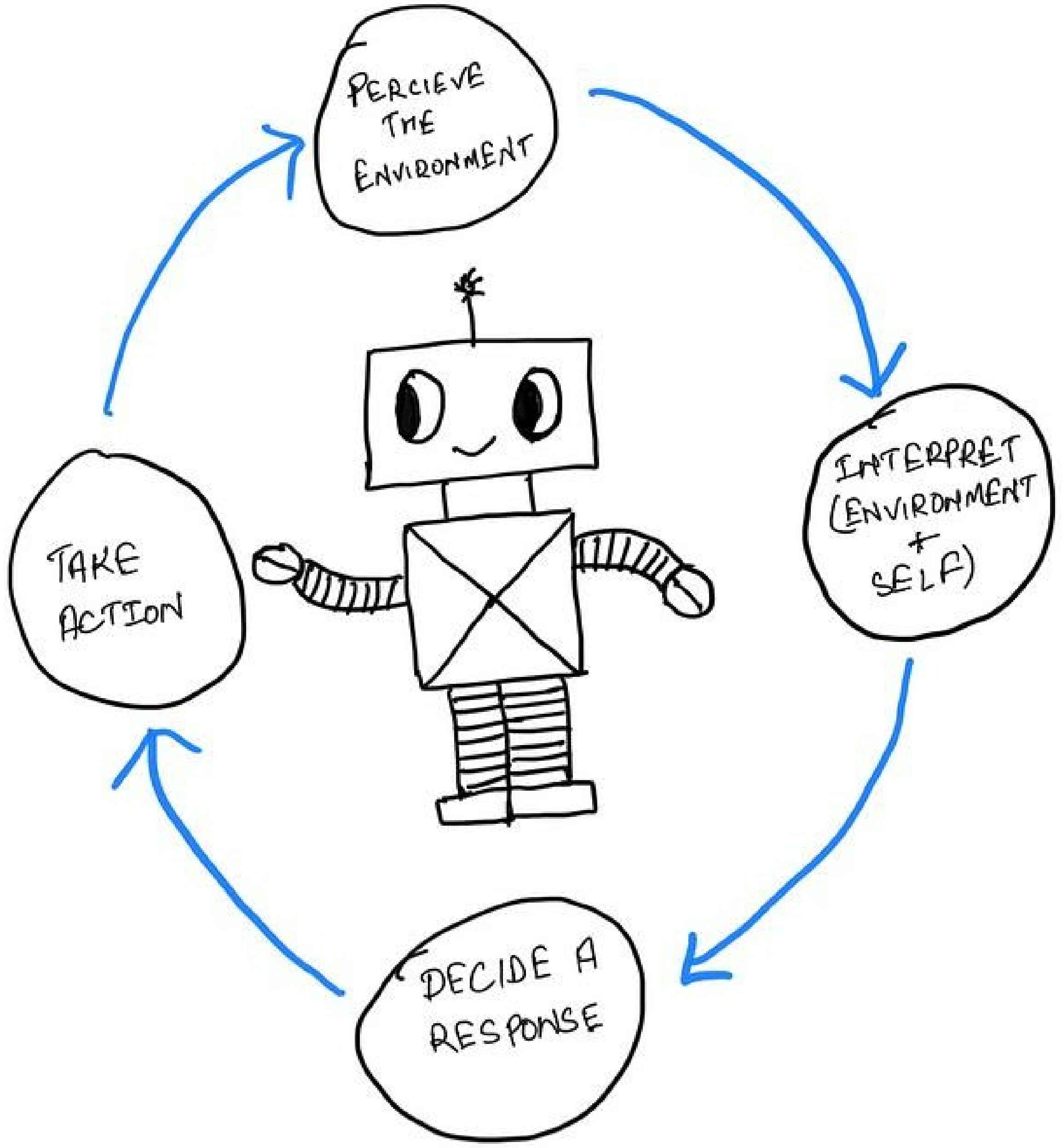
ROBOTICS CORNER

Introduction to ROS



Miriam Hani

ROBOTICS CONCEPT:

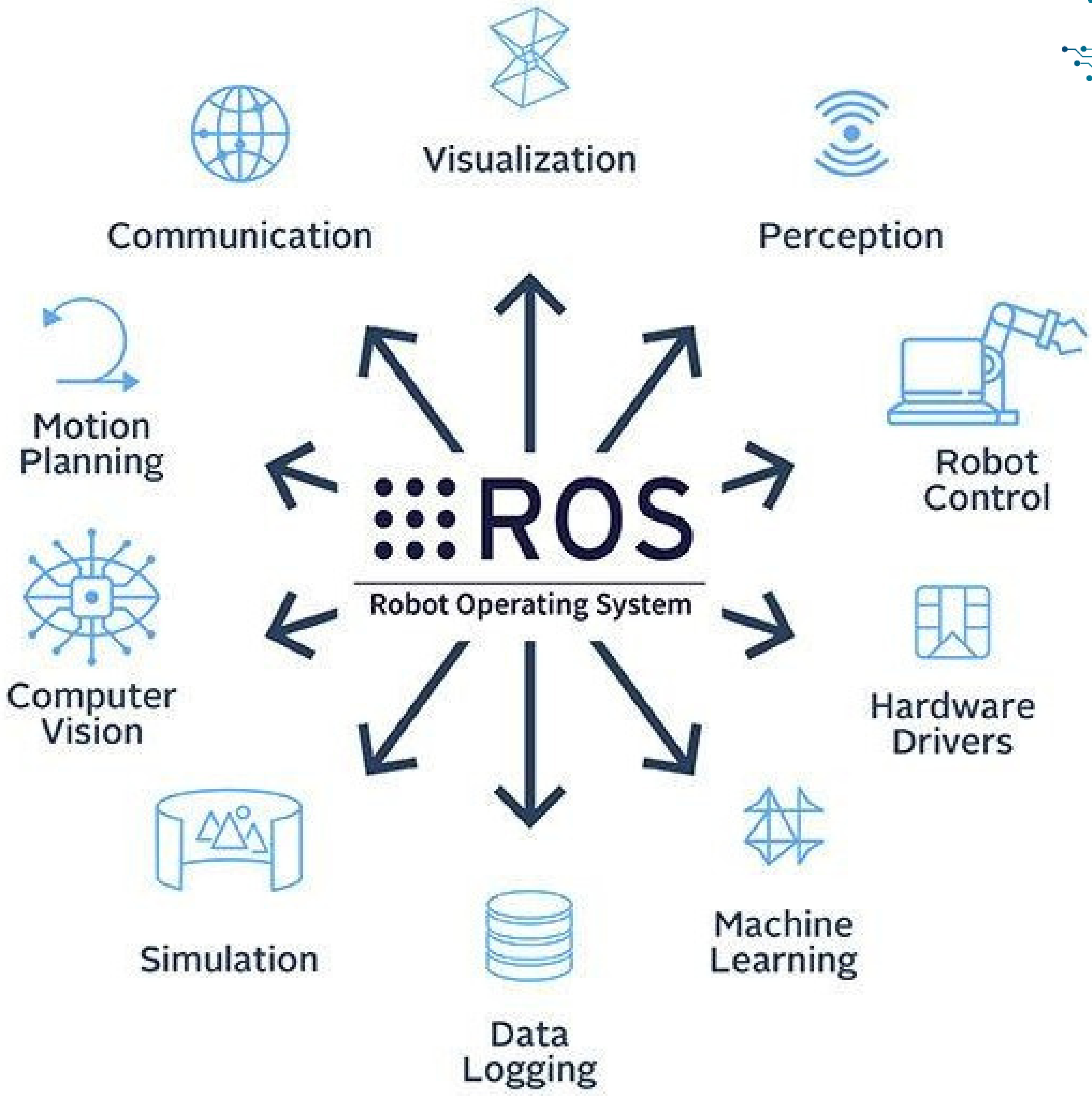


WHAT IS ROS?

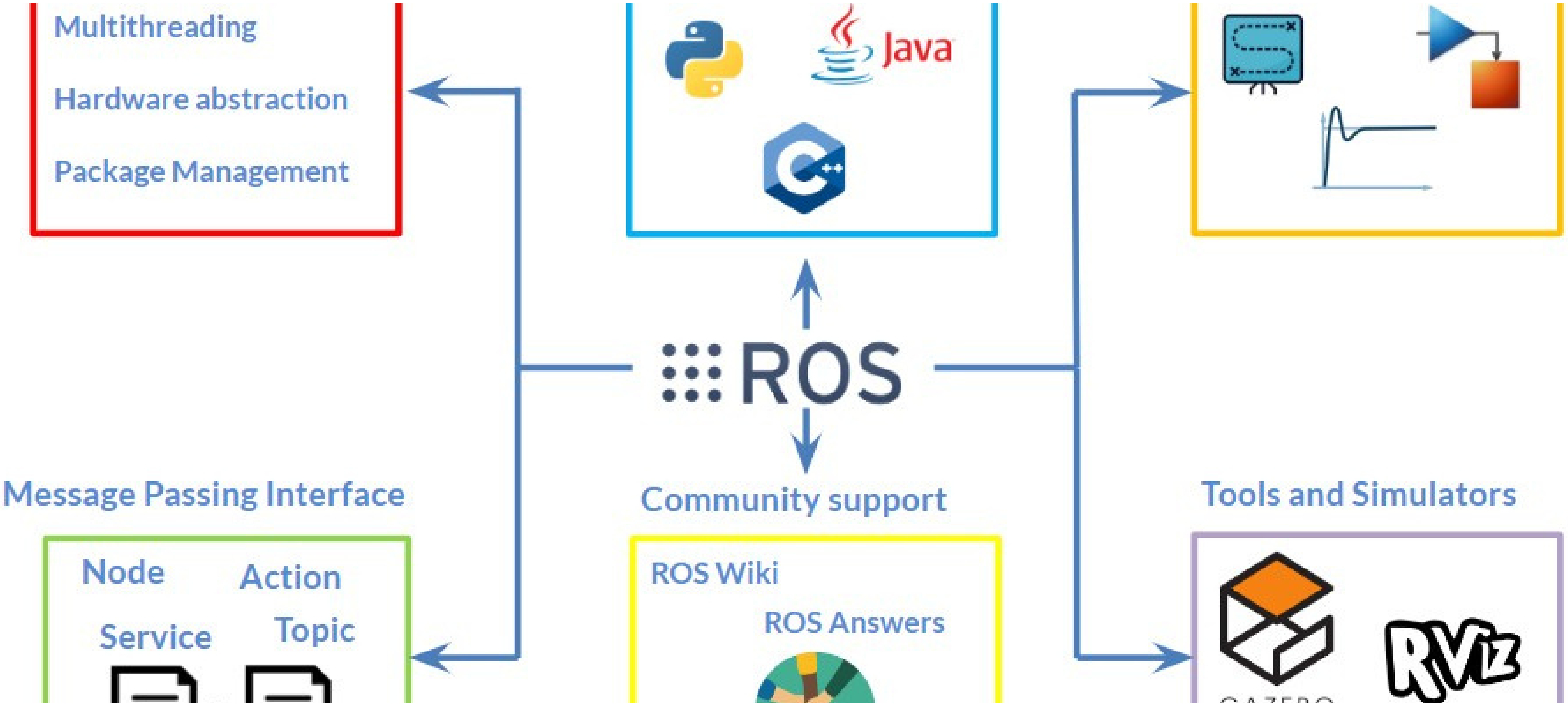
- ROS is an open-source robot operating system
- A set of software libraries and tools that help you build robot applications that work across a wide variety of robotic platforms
- Originally developed in 2007 at the Stanford Artificial Intelligence Laboratory and development continued at Willow Garage
- Since 2013 managed by OSRF (Open Source Robotics Foundation)

ROS

Robot Operating System

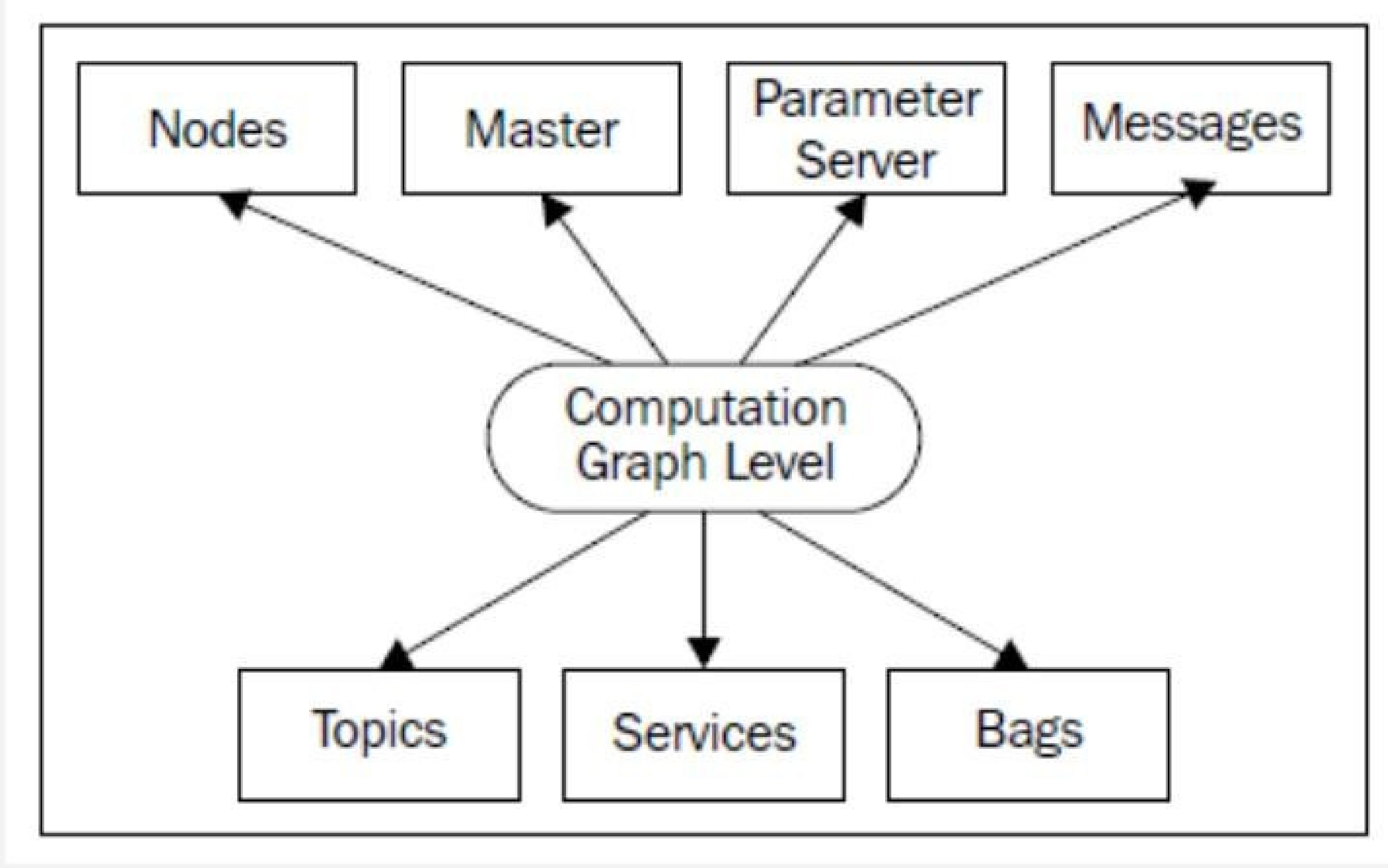

Visualization
Communication
Perception
Motion Planning
ROS
Robot Operating System
Robot Control
Computer Vision
Hardware Drivers
Simulation
Machine Learning
Data Logging

ROS CAPABILITIES



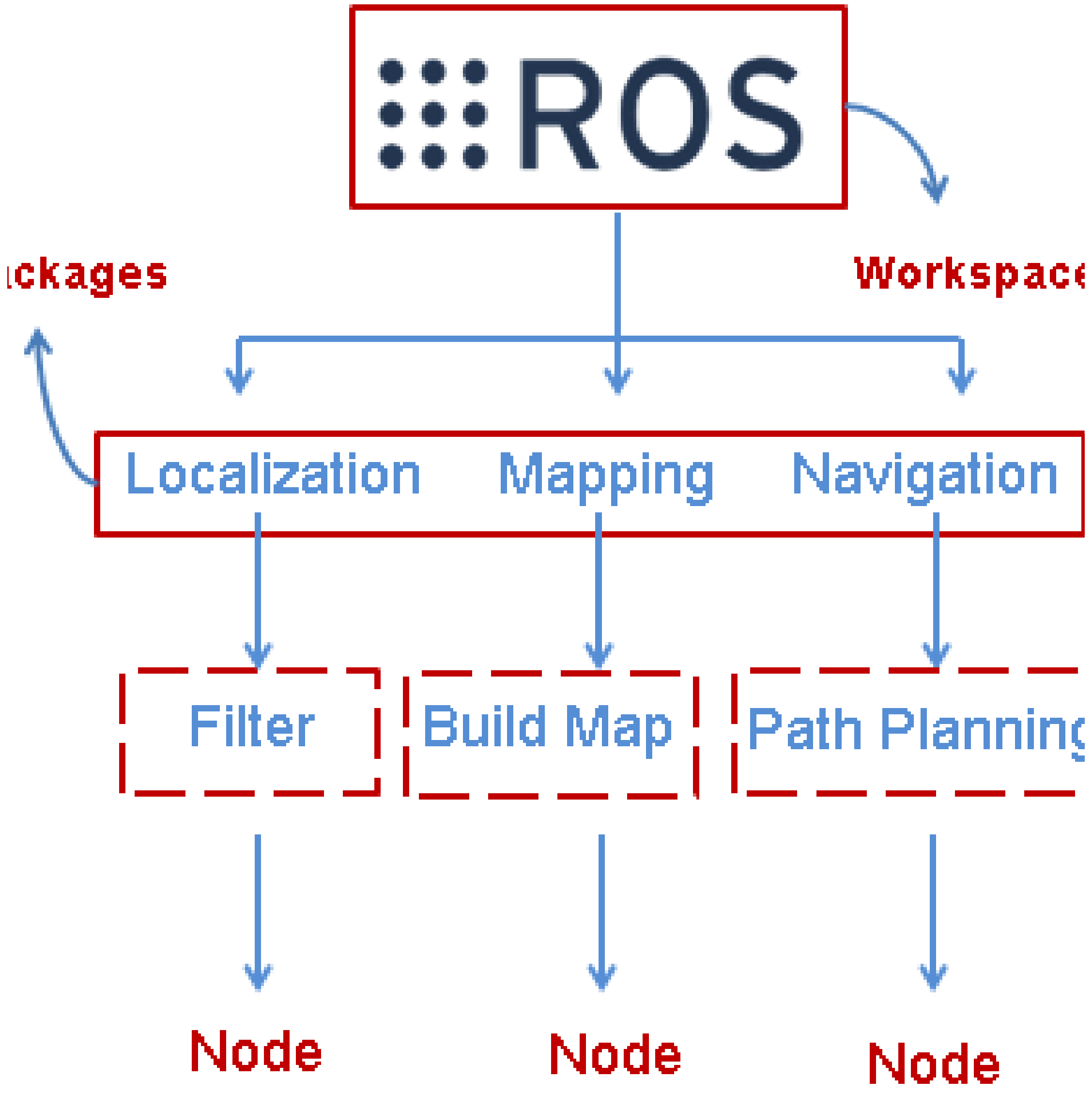
ROS COMPUTATION GRAPH LEVEL

- ROS creates a network where all the processes are connected. Any node in the system can access this network, interact with other nodes, see the information that they are sending, and transmit data to the network.



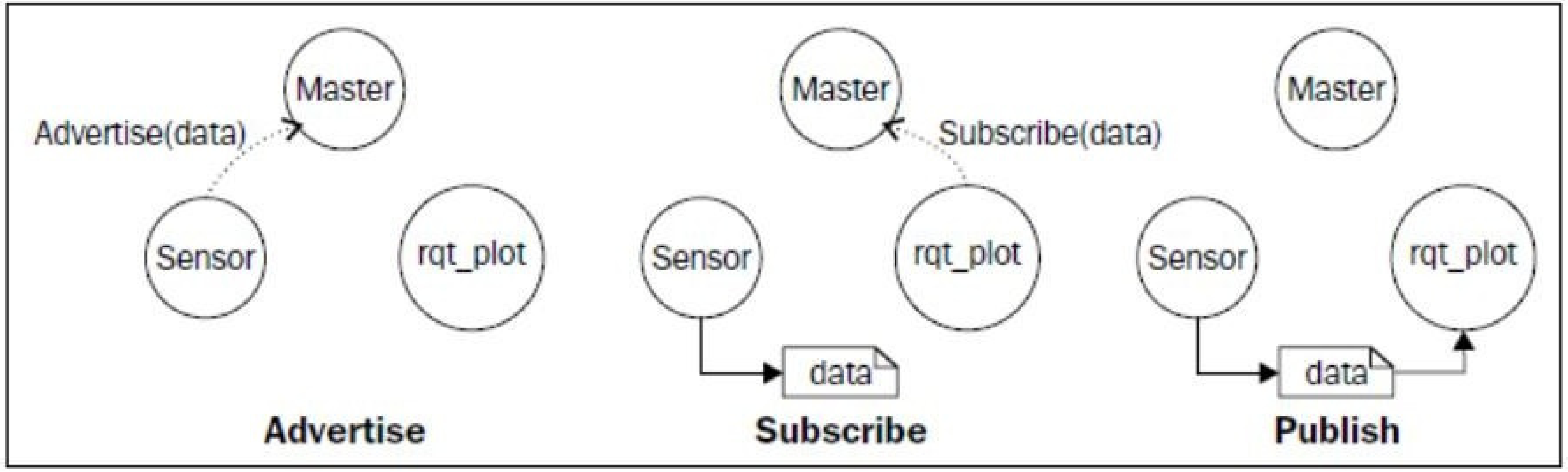
1. NODE

- A node refers to the smallest unit of processor running in ROS. Think of it as one executable program. For example, in case of mobile robots, the program to operate the robot is broken down into specialized functions. Specialized node is used for each function such as path planning , filtering and map building
- Nodelets are special nodes designed to run multiple nodes in a single process, with each nodelet being a thread
- A node must have a unique name in the system. This name is used to permit the node to communicate with another node using its name without ambiguity. A node can be written using different libraries, such as roscpp and rospy

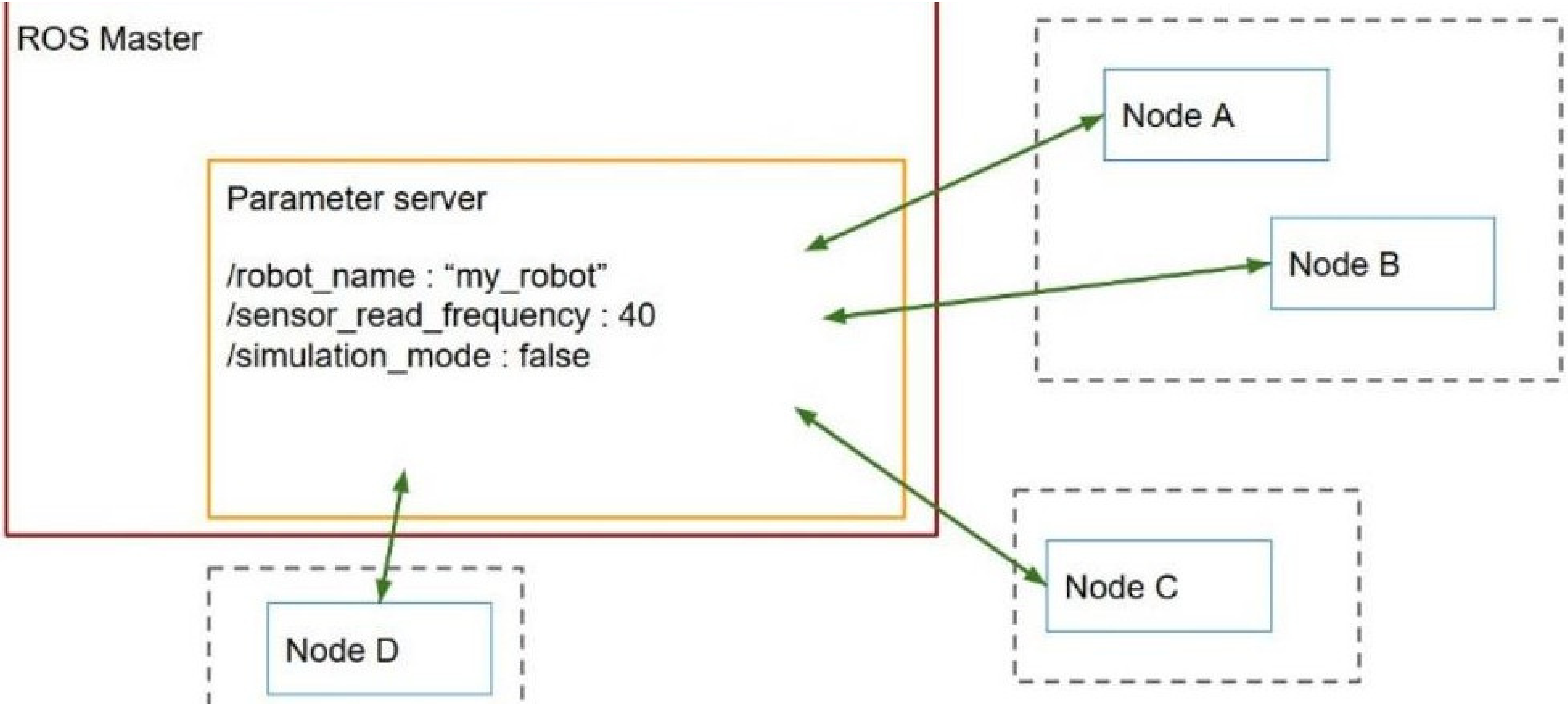


2. MASTER

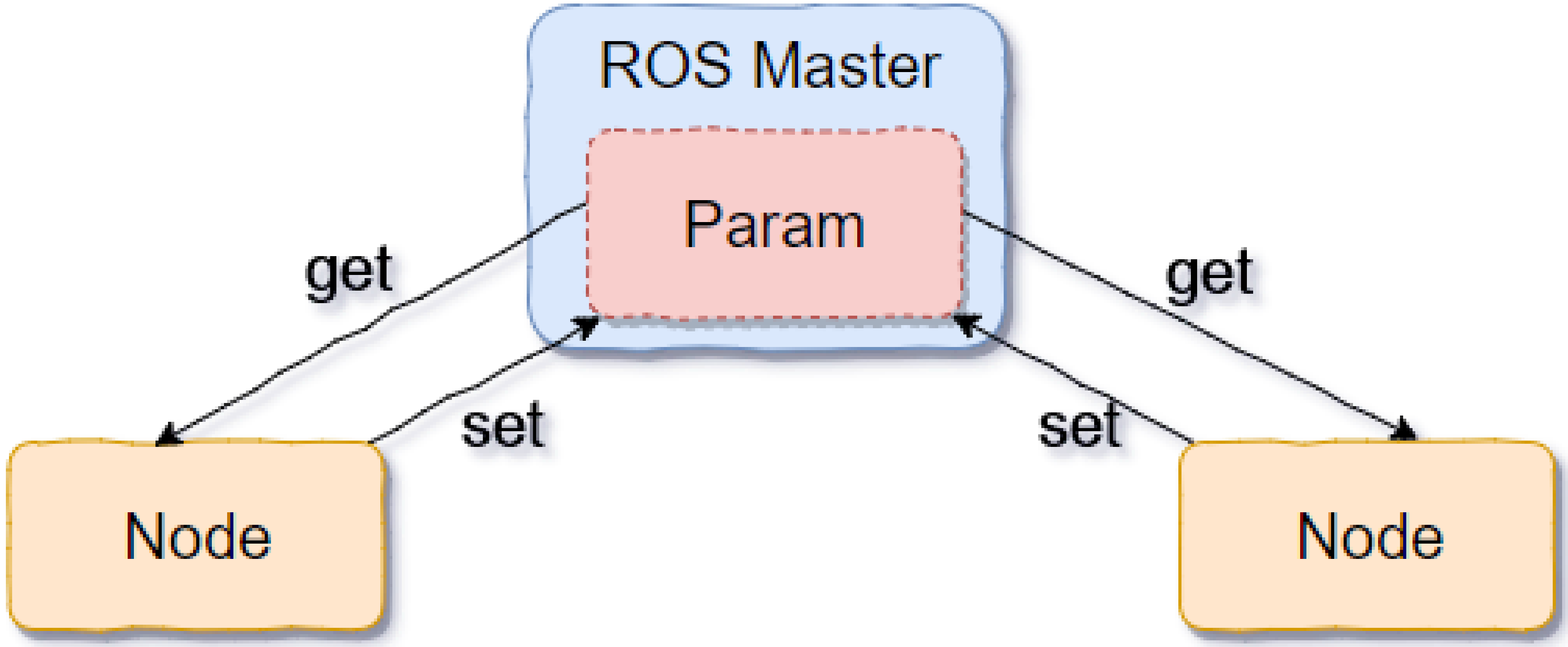
- The ROS master provides naming and registration services to the rest of the nodes in the ROS system.
- It tracks publishers and subscribers to topics as well as services.
- The role of the master is to enable individual ROS nodes to locate one another. Once these nodes have located each other; they communicate with each other in a peer-to-peer fashion.



3. PARAMETER SERVER

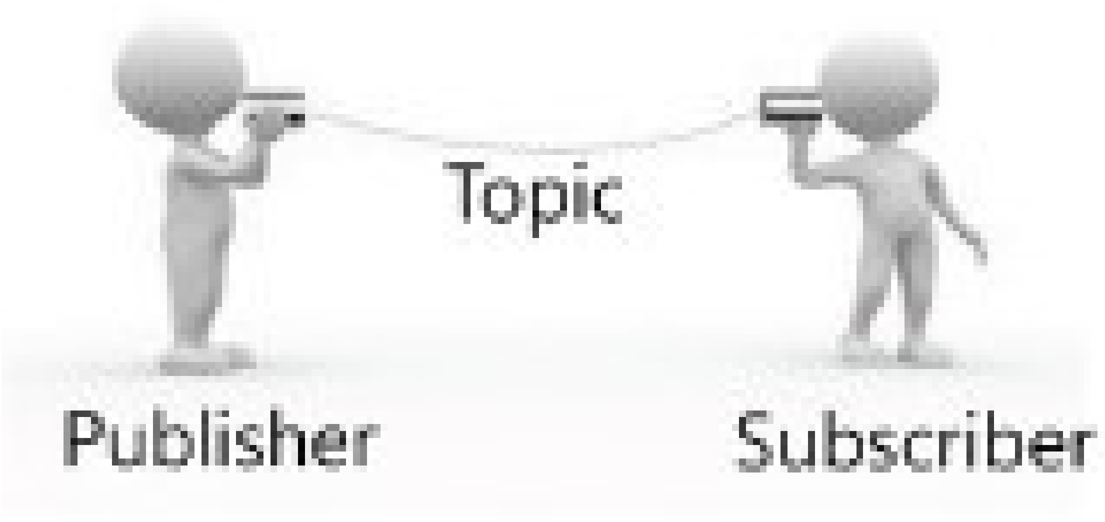


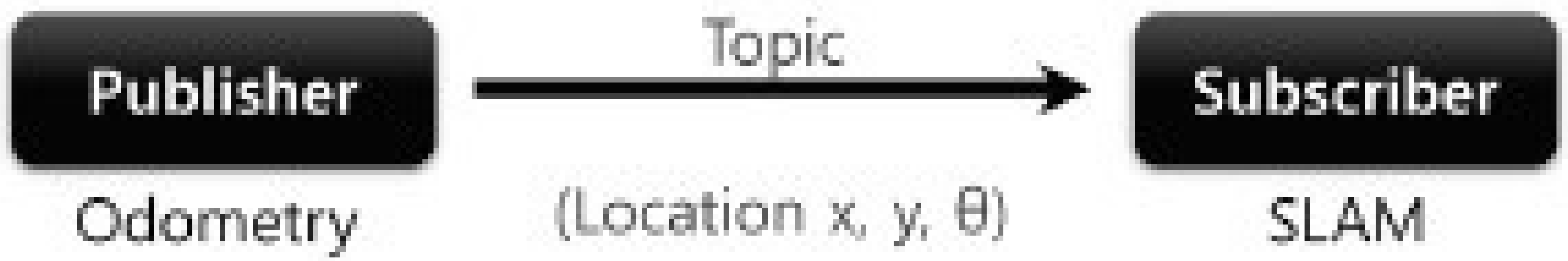
- The parameter server is basically a **dictionary containing global variables which are accessible from anywhere in the current ROS environment**.
- At any time, **a node can read a parameter, modify a parameter, and can create new ones**.
- A ROS parameter is basically just one of the shared variable stored in the parameter server.



黑马程序员

Communication using ROS Topics:



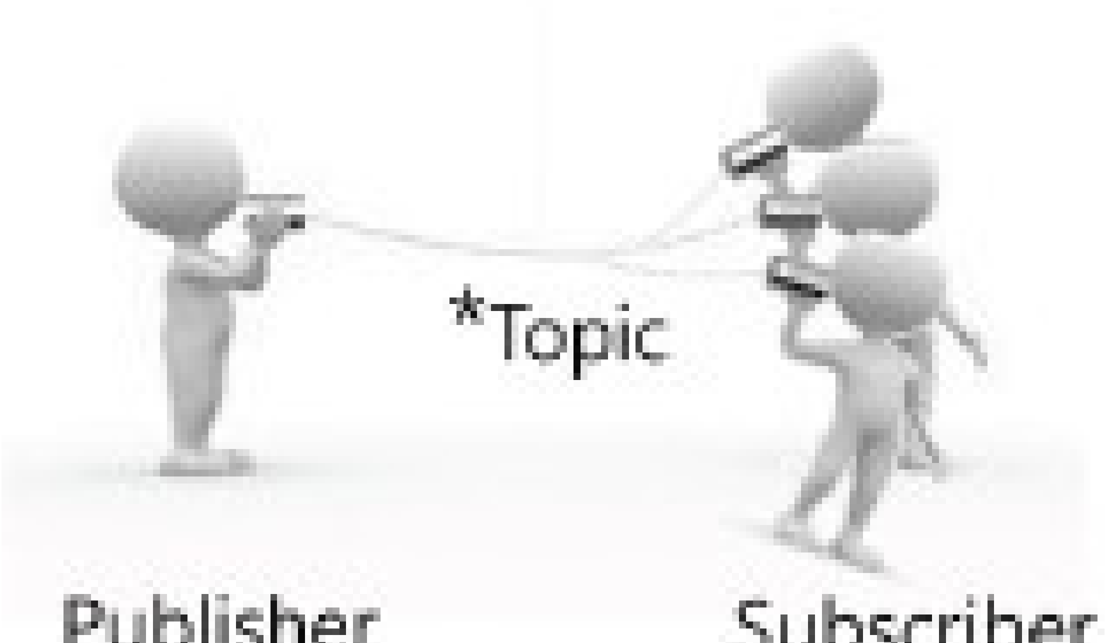


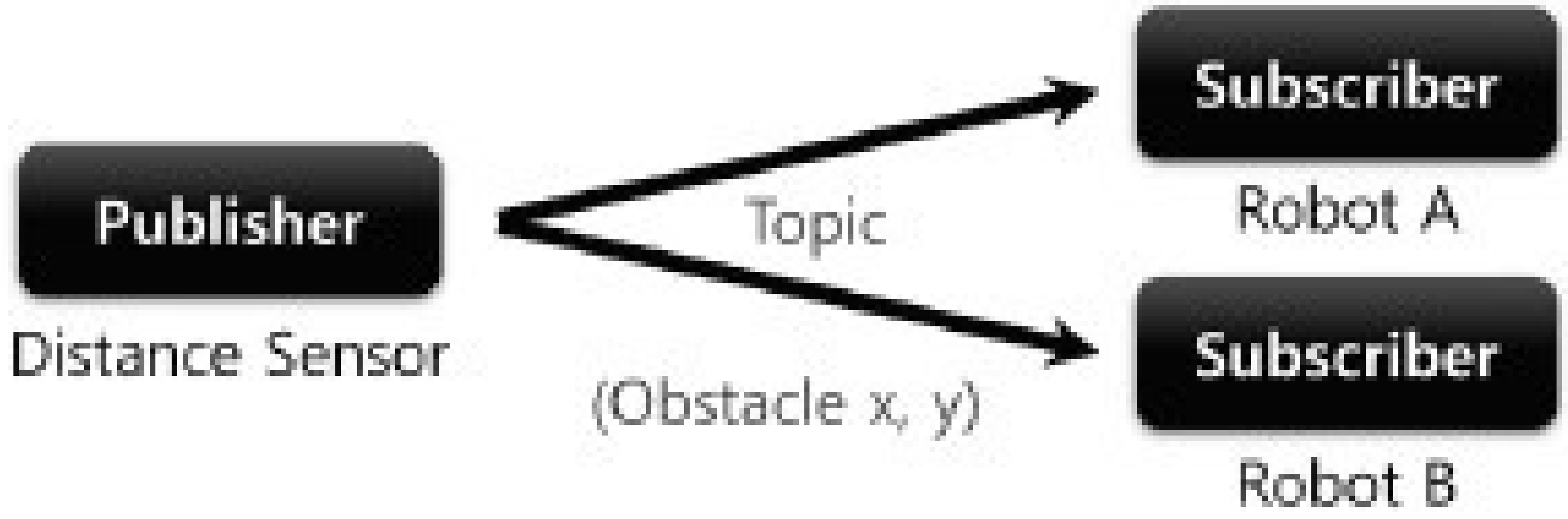
Publisher → Topic → Subscriber

Odometry

(Location x, y, θ)

SLAM





Publisher → Topic → Subscriber, Subscriber

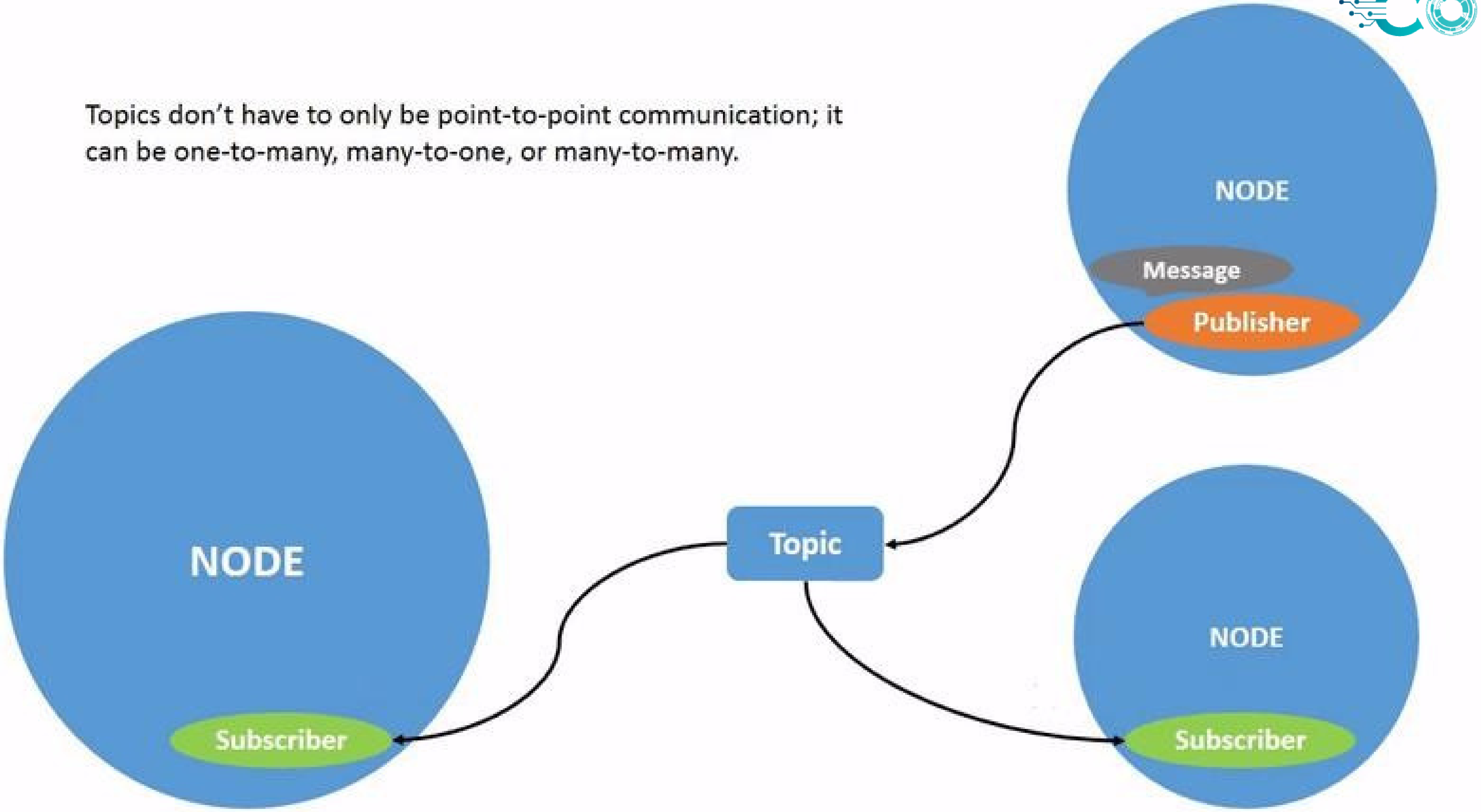
Distance Sensor

(Obstacle x, y)

Robot A

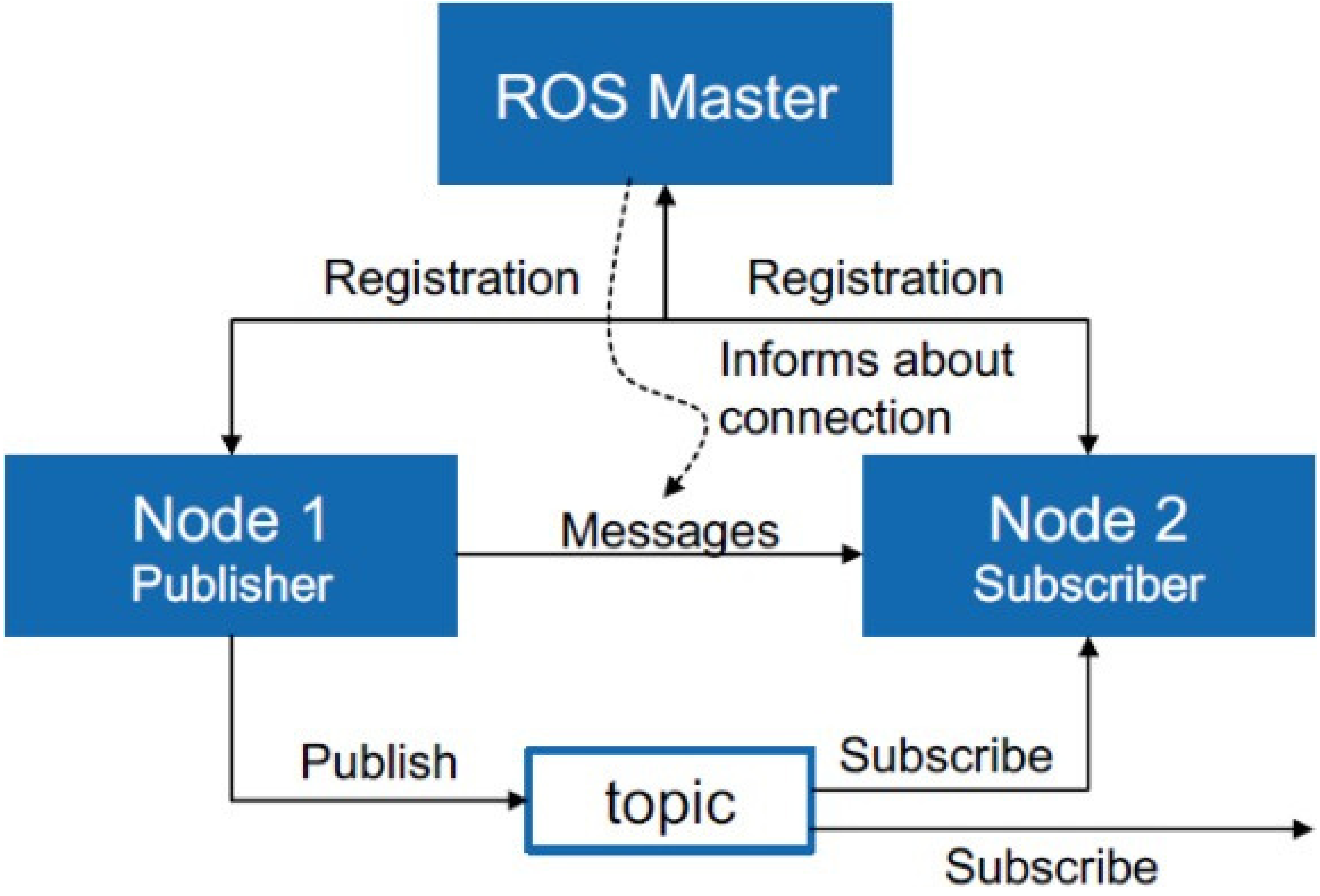
Robot B

Topics don’t have to only be point-to-point communication; it can be one-to-many, many-to-one, or many-to-many.

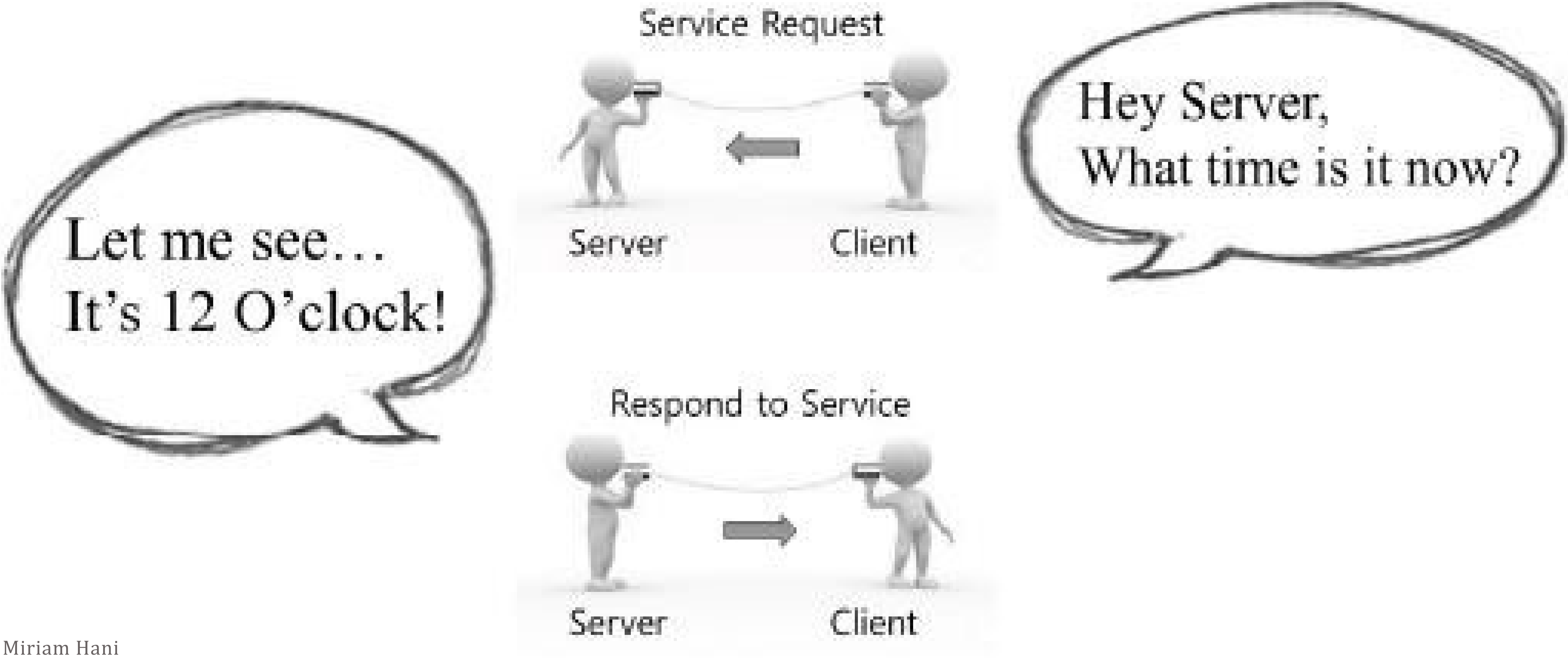


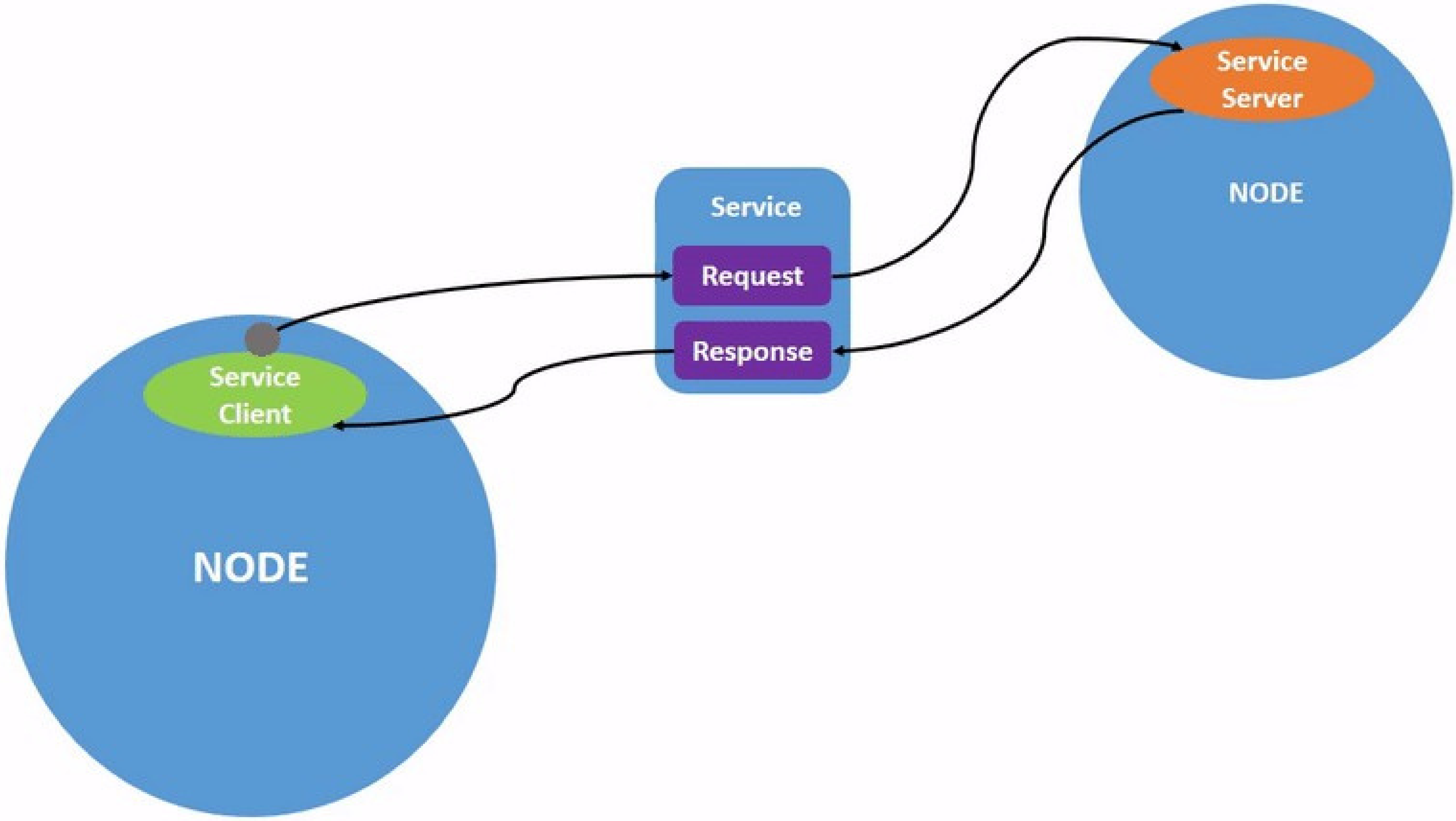
Eng. Miriam Hani

- Topics are buses used by nodes to transmit data.
  Topics can be transmitted without a direct connection between nodes.
- A topic can have various subscribers and can also have various publishers
  Each topic is strongly typed by the ROS message type used to publish it, and.
  A node can subscribe to a topic only if it has the same message type.

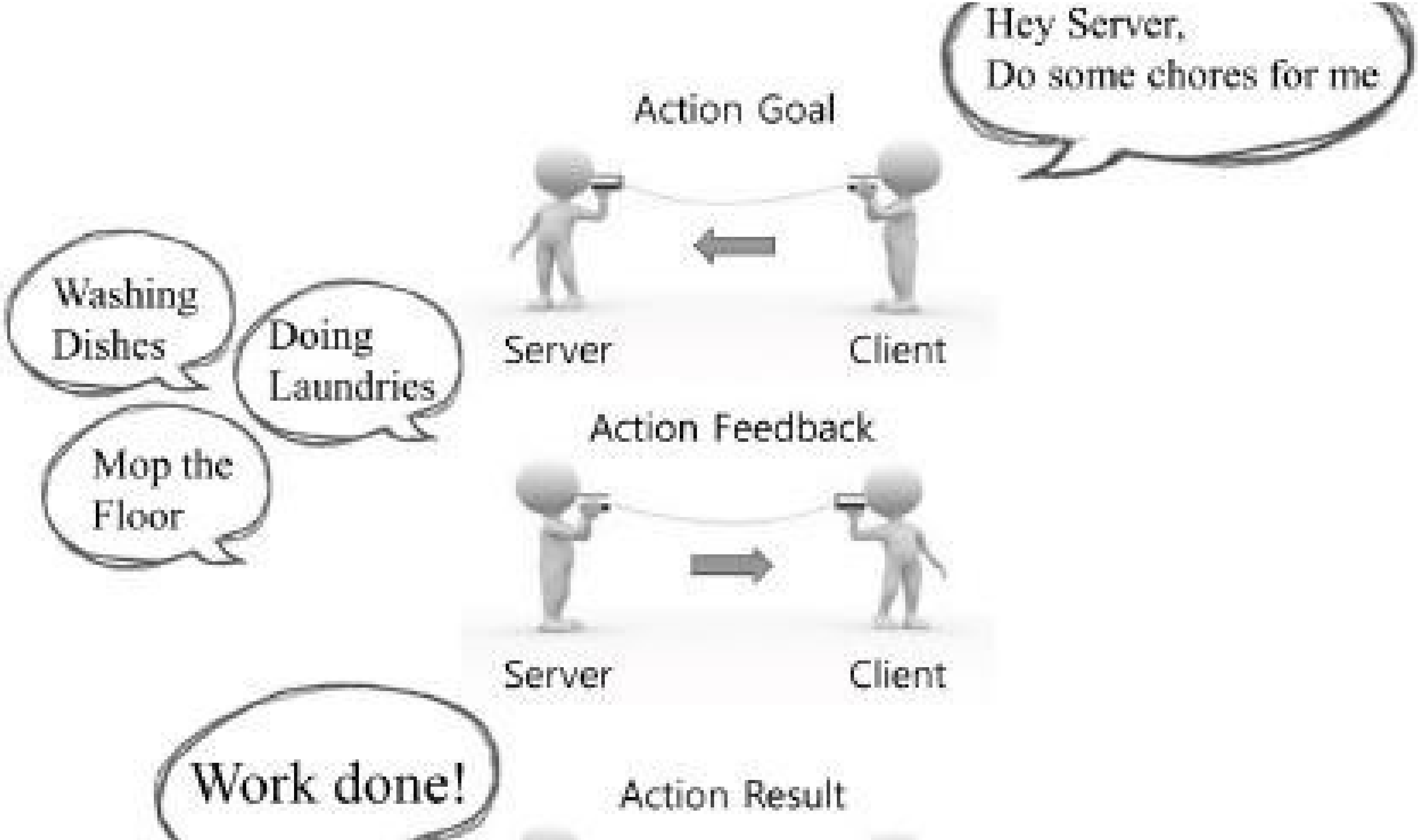


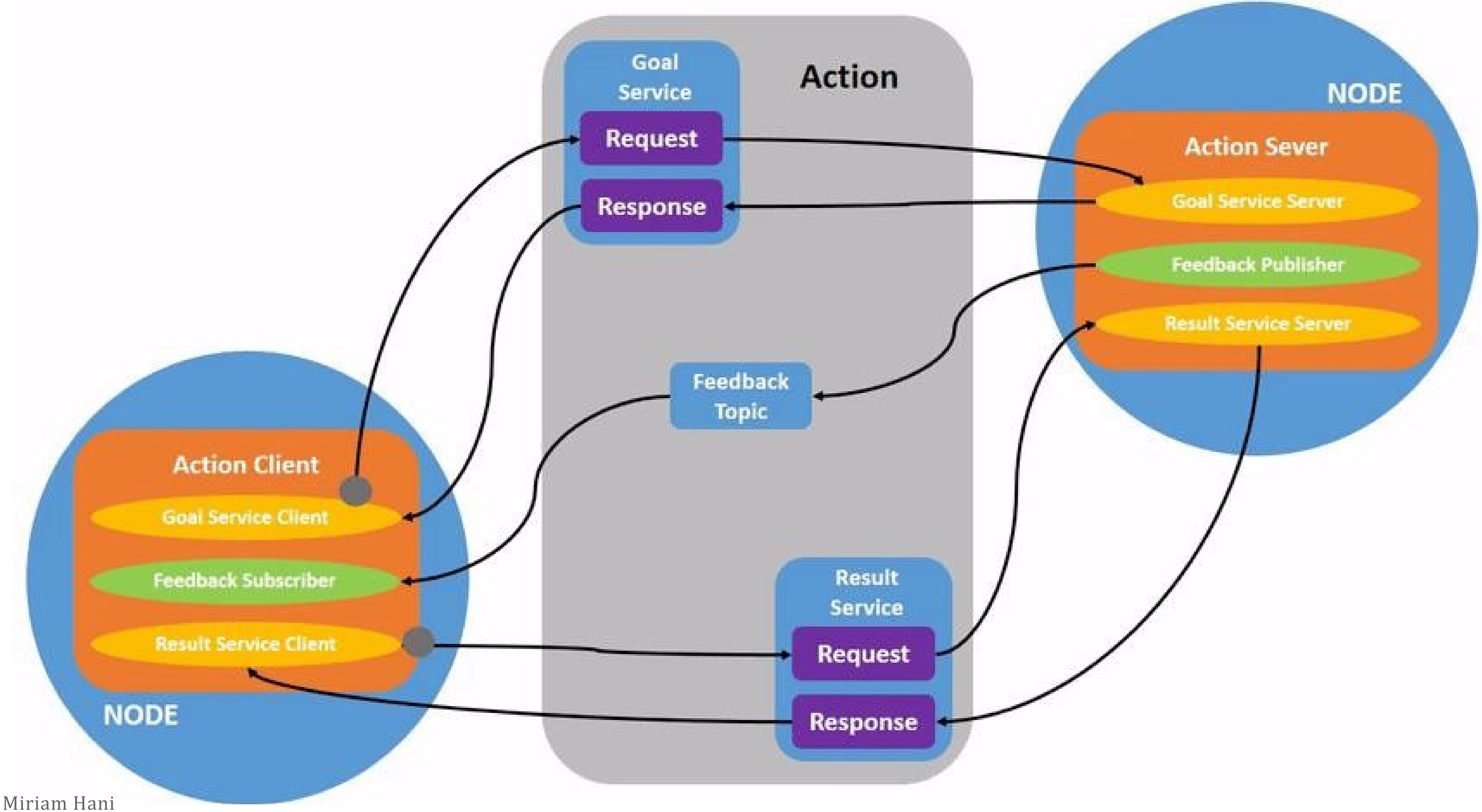
Communication using ROS Services:




Service
Server
NODE
Service
Request
Response
Service
Client
NODE

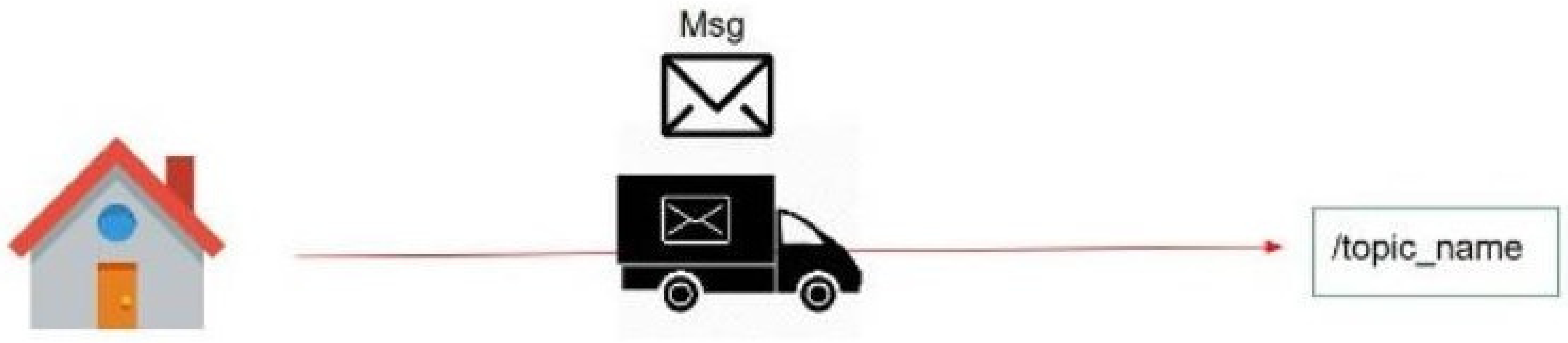
Communication using ROS Actions:



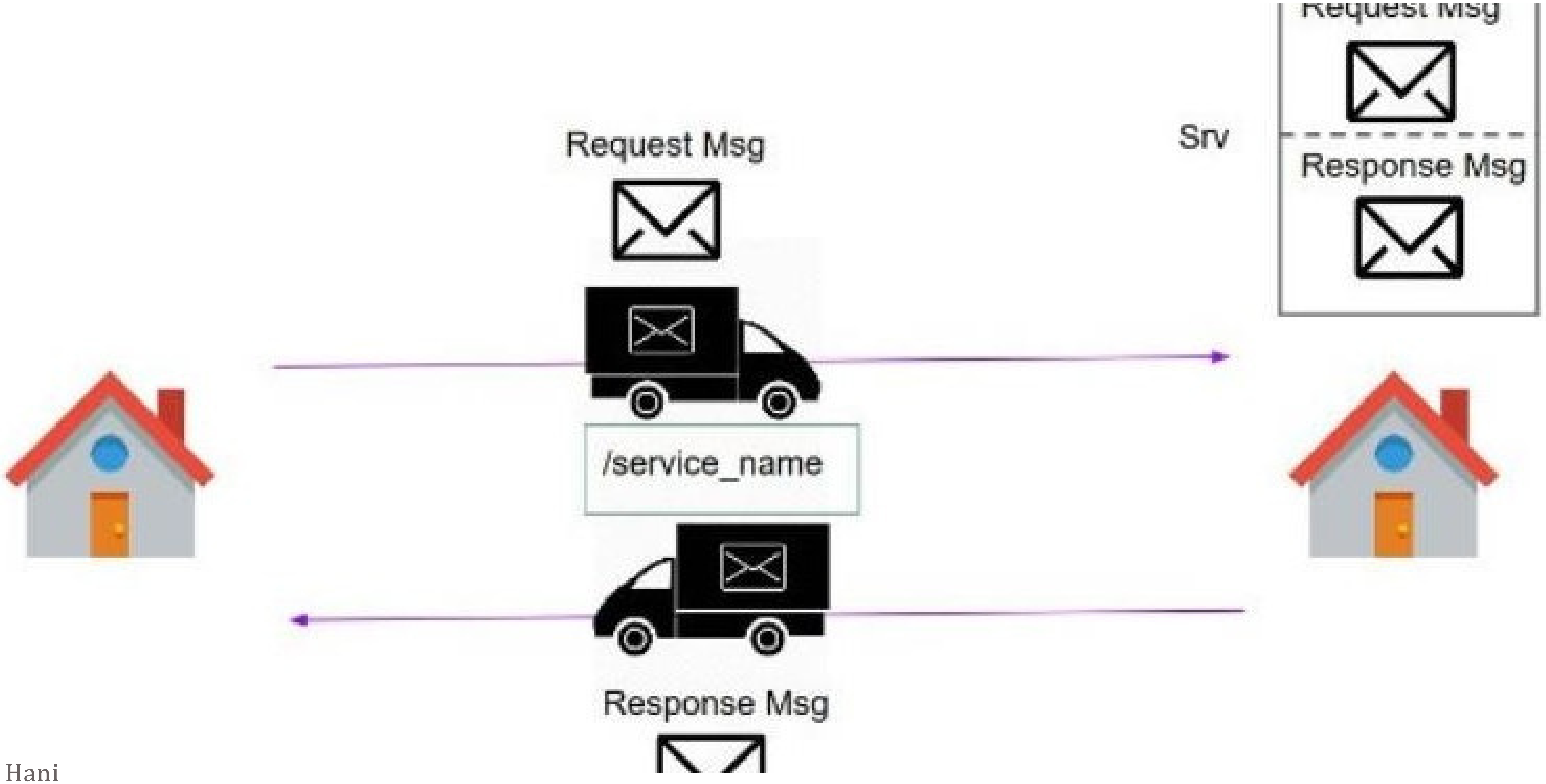

Goal Service
Request
Response
Action
NODE
Action Sever
Goal Service Server
Feedback Publisher
Result Service Server
Feedback Topic
Action Client
Goal Service Client
Feedback Subscriber
Result Service Client
NODE
Result Service
Request
Response

REAL LIFE EXAMPLE !

- When you send a mail, the transport company will transport your letter. The content of this letter is the analogy of a ROS message.

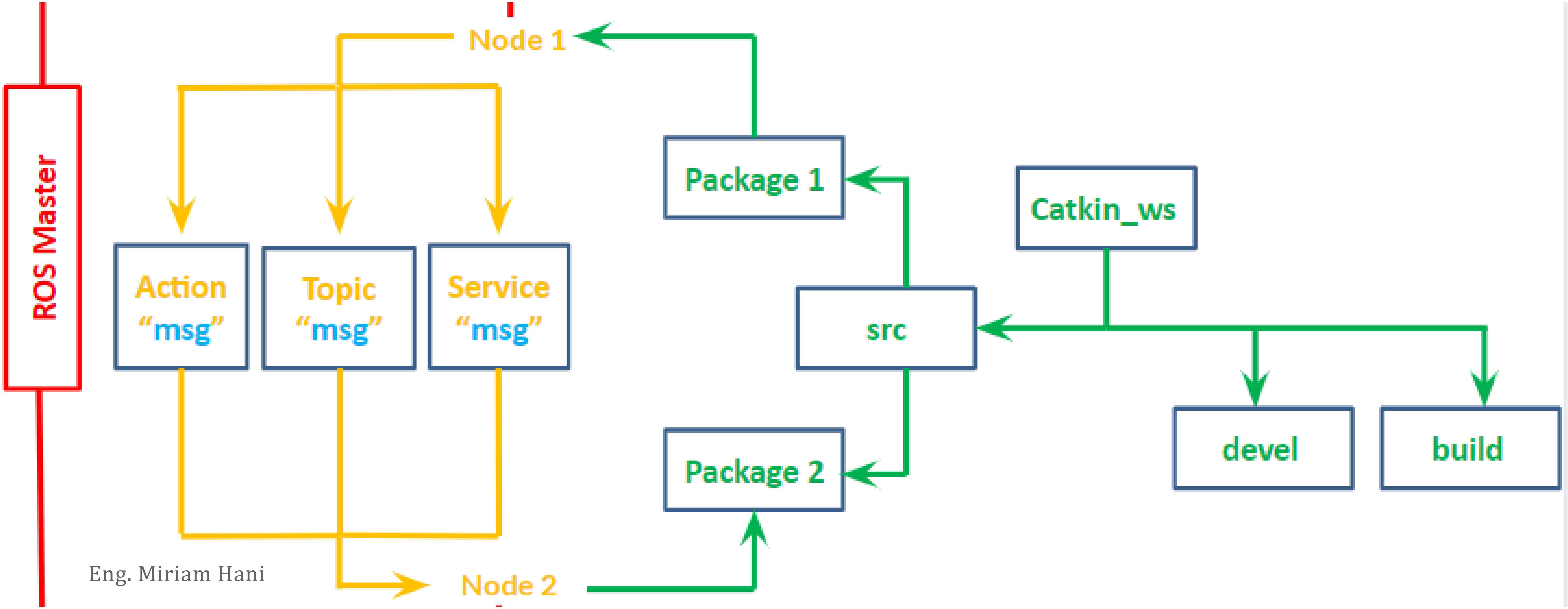


- When you send a letter and you wait for a response, then the first letter contains a Request message, and the letter that you receive back contains a Response message.
- The combination of the two message definitions is the service definition.



ROS MESSAGE COMMUNICATION

- There are three different methods of exchanging messages: a topic which provides a unidirectional message transmission/ reception, a service which provides a bidirectional message request/response and an action which provides a bidirectional message goal/result/feedback.



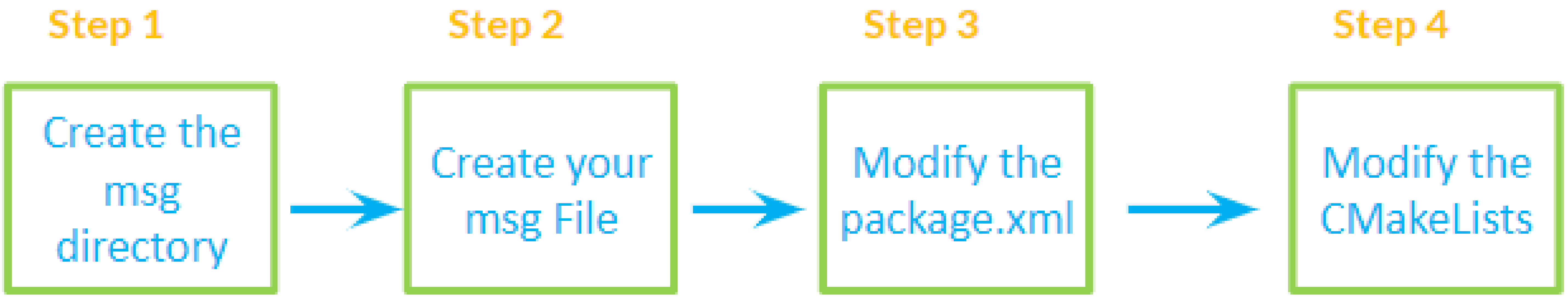
Eng. Miriam Hani

Message file

- msg files are simple text files that describe the fields of a ROS message. They are used to generate source code for messages in different languages. msgs are just simple text files with a field type and field name per line.

  - Types:
    - int8, int16, int32, int64 (plus uint*)
    - float32, float64
    - String
    - time, duration
    - variable-length array and fixed-length array

HOW TO BUILD A CUSTOM MESSAGE?!



Step 1

Create the msg directory

Step 2

Create your msg File

Step 3

Modify the package.xml

Step 4

Modify the CMakeLists