Robot Operating Systems

Robot Software Development

Abdelbacet Mhamdi

2025-05-08

ISET Bizerte

Outline

2. Getting started with ROS2

2. Getting started with ROS2

2. Getting started with ROS2

We begin by checking which shell we are using on Linux:

```bash
which $SHELL
```

Sourcing our ROS installation.

```bash
source /opt/ros/humble/setup.bash
```

Run the turtlesim_node executable from the turtlesim package.

```bash
ros2 run turtlesim turtlesim_node
```

The below command returns the list of current active topics

```bash
ros2 topic list -t
```

Details on a particular type of message can be retrieved through:

```bash
ros2 interface show geometry_msgs/msg/Twist
```

We can publish data on a topic as follows:

```bash
ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist
"{linear: {x: 1.}}"
```

`--once` means we publish only one message and exit.

```bash
ros2 topic pub -r 1 /turtle1/cmd_vel geometry_msgs/msg/Twist
"{linear: {x: 1.,y: 0.,z:0.}, angular:{x: 0.,y: 0.,z: .7}}"
```

`-r 1` means we keep publishing our message in a steady stream at 1 Hz. We can see the data being published on the `/turtle1/cmd_vel` by running:

```bash
ros2 topic echo /turtle1/cmd_vel
```

We can also use the keyboard to move around `turtle1`

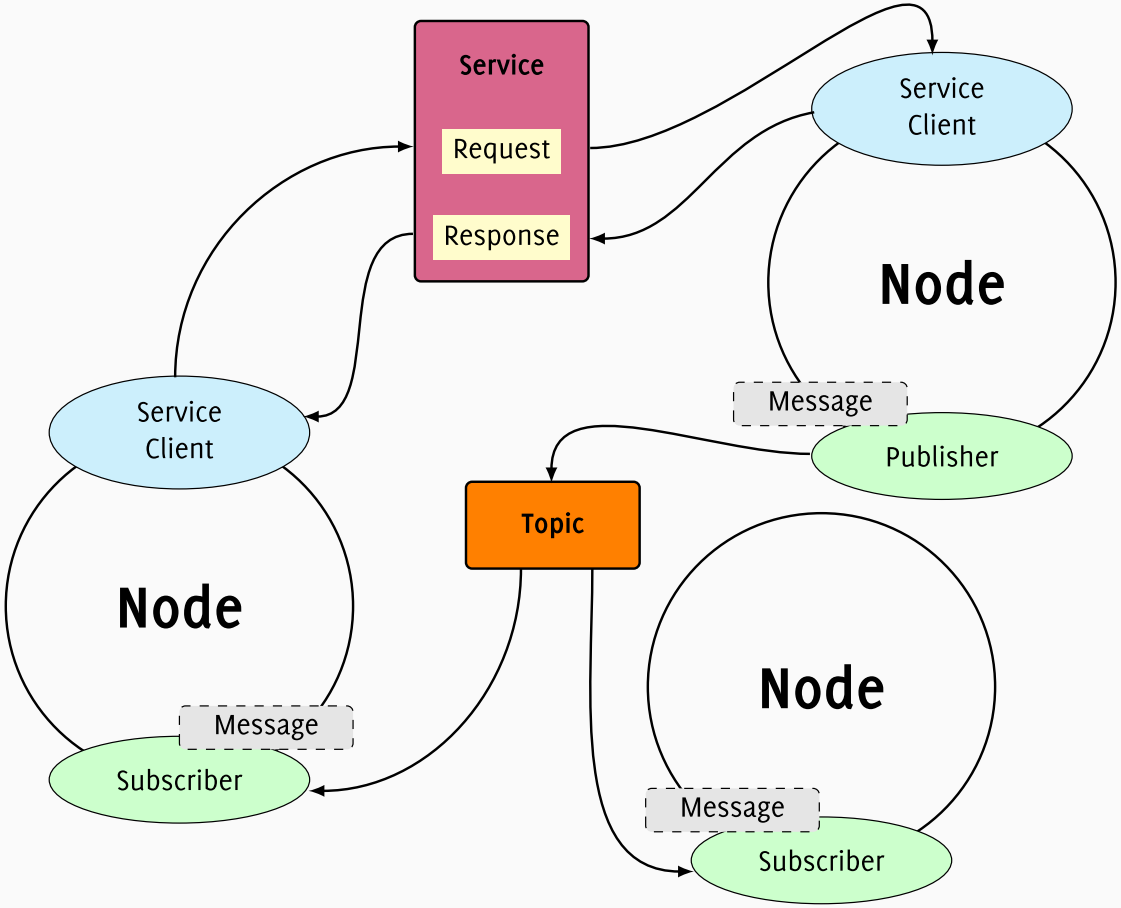
```bash
ros2 run teleop_twist_keyboard teleop_twist_keyboard --ros-args --remap cmd_vel:=/turtle1/cmd_vel
```

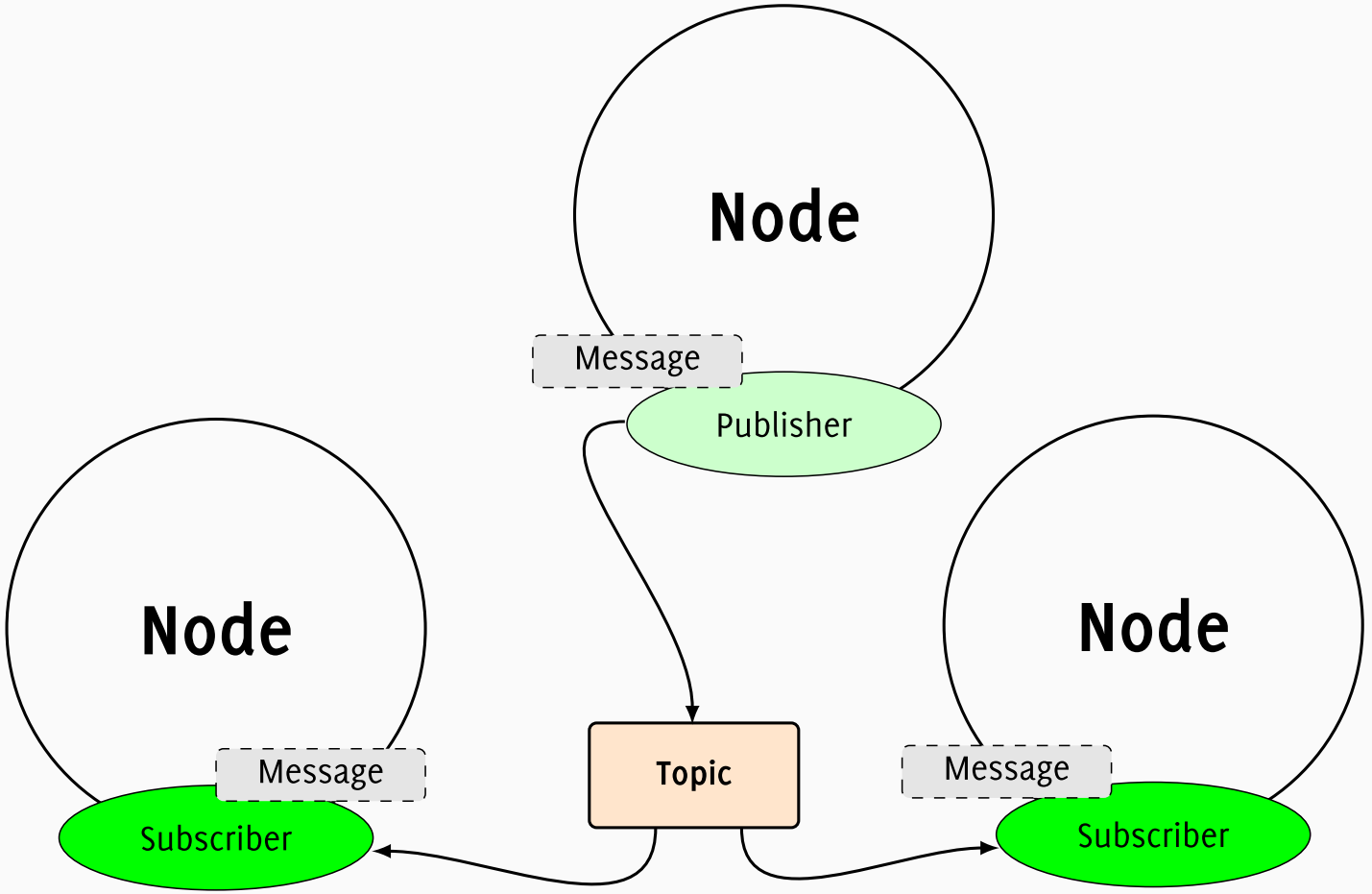
The executable `turtle_teleop_key` does the same previous thing and avoids all the hassle of remapping

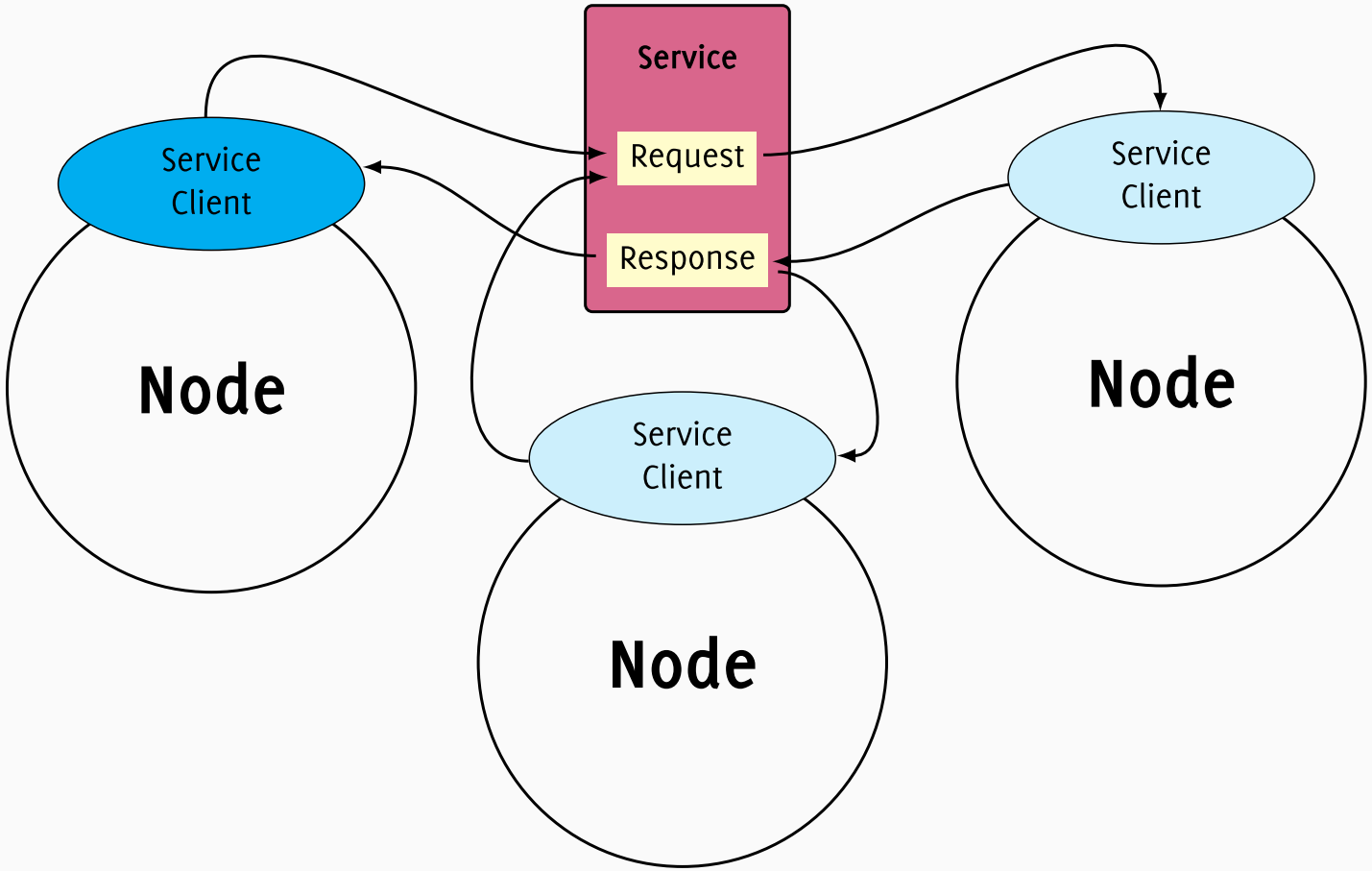
```bash
ros2 run turtlesim turtle_teleop_key
```

This utility allows us to display the graph of data flow between our running nodes;

```bash
rqt_graph
```


Service
Request
Response
Service Client
Node
Message
Publisher
Topic
Service Client
Node
Message
Subscriber
Node
Message
Subscriber


Node
Message
Publisher
Node
Message
Subscriber
Topic
Node
Message
Subscriber


Service
Request
Response
Service Client
Service Client
Service Client
Node
Node
Node

Thank you for your attention!

References