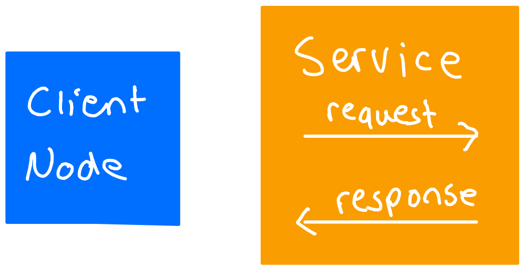
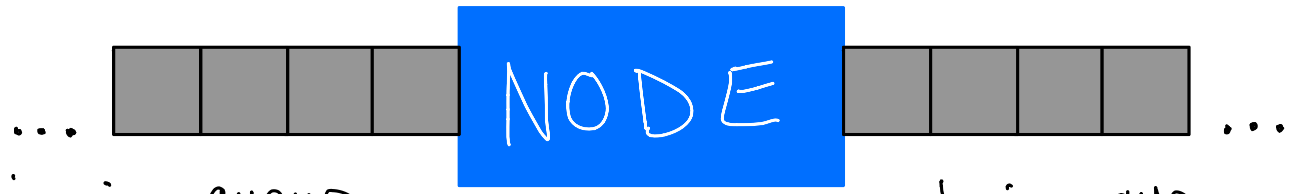
ROS

- Services have always a Request & Response
- Topics are asynchronous $\leftrightarrow$ Services are synchronous



```
$ rosservice list
$ rosservice call <service-name> <service-msg>
```



ingoing queue   outgoing queue

→ queue size can be defined but as long as it is full, new incoming requests/messages are discarded

- ros::spinOnce(); needs to be called so that requests that get pushed into queue can immediately be processed in a new thread

- void myFunction(int &value);

  ↳ automatically dereferencing pointer

ROS BAGS

- collect (record) messages from topics
  ↳ can be played again → used to resimulate some behaviour

```
$ rosbag record <topic_name>
$ rosbag compress <bag_file>
$ rosbag play <bag_file>
```

ADDITIONAL COMMANDS

- include and run other launch file within launch file
  `<include file="$(find <package_name>)/launch/<file.launch>`
- run shell/bash scripts with launch file by using node
  ↳ can run any executable file → type = filename

`$ rosmsg show <msg_file>` → see structure of message

`$ export | grep ROS` → show paths where ROS looks for packages