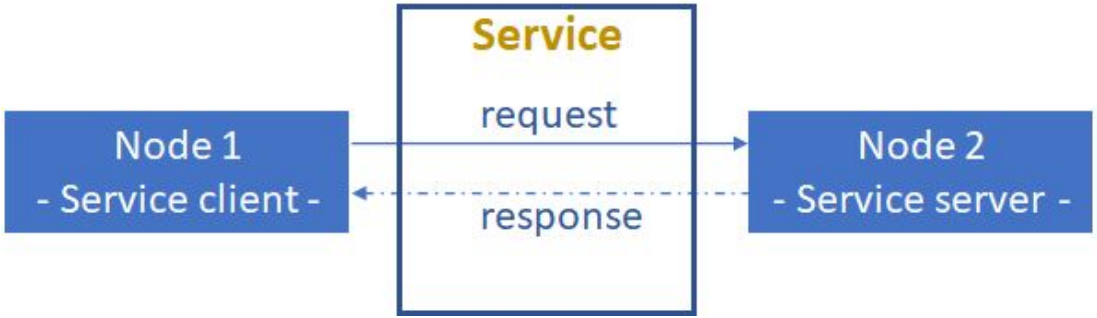
Path planning of mobile robots

Lab 5 - Autonomous Robots

ROS Services

- ROS service is based on a request and response communication, so it is always two nodes communicating with each other.
- The node that provides the service is called server while the node that asks for it is called client.



- The service definition is done in .srv files which has the following format:

```
Request
---
Response
```

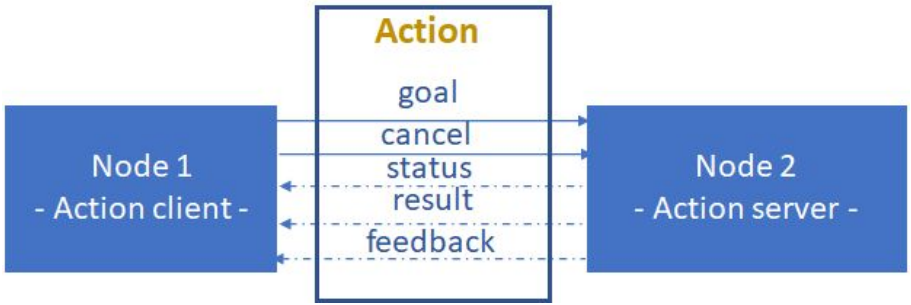
ROS Services

Similar to ROS topics, the following commands may be used to get information about the current services:

```
$ rosservice list # list all available services
$ rosservice type /service_name# show the type of service
$ rosservice call /service_name args# call a service with the request contents
```

ROS actions (actionlib)

- Actions are similar to service calls, but can be used in the case of slow tasks since they provide the possibility to cancel the task (preempt) and also to receive feedback on the task progress.



Similar in structure to services, actions are defined in .action files, having the following format:

Goal

Result

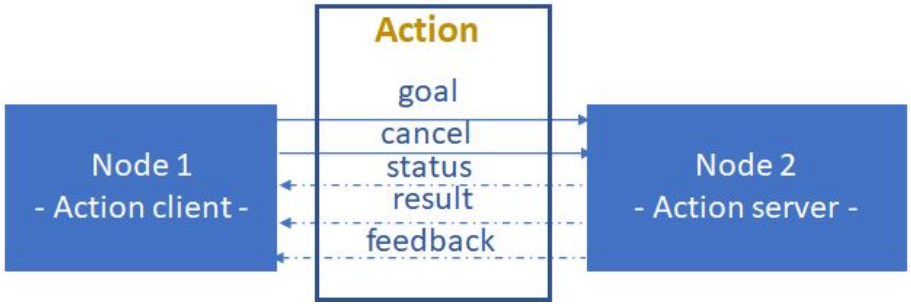
Feedback

ROS actions (actionlib)

In code:

- Needs to implement callbacks for goal and cancel

- Needs to provide status, result and feedback in its action function



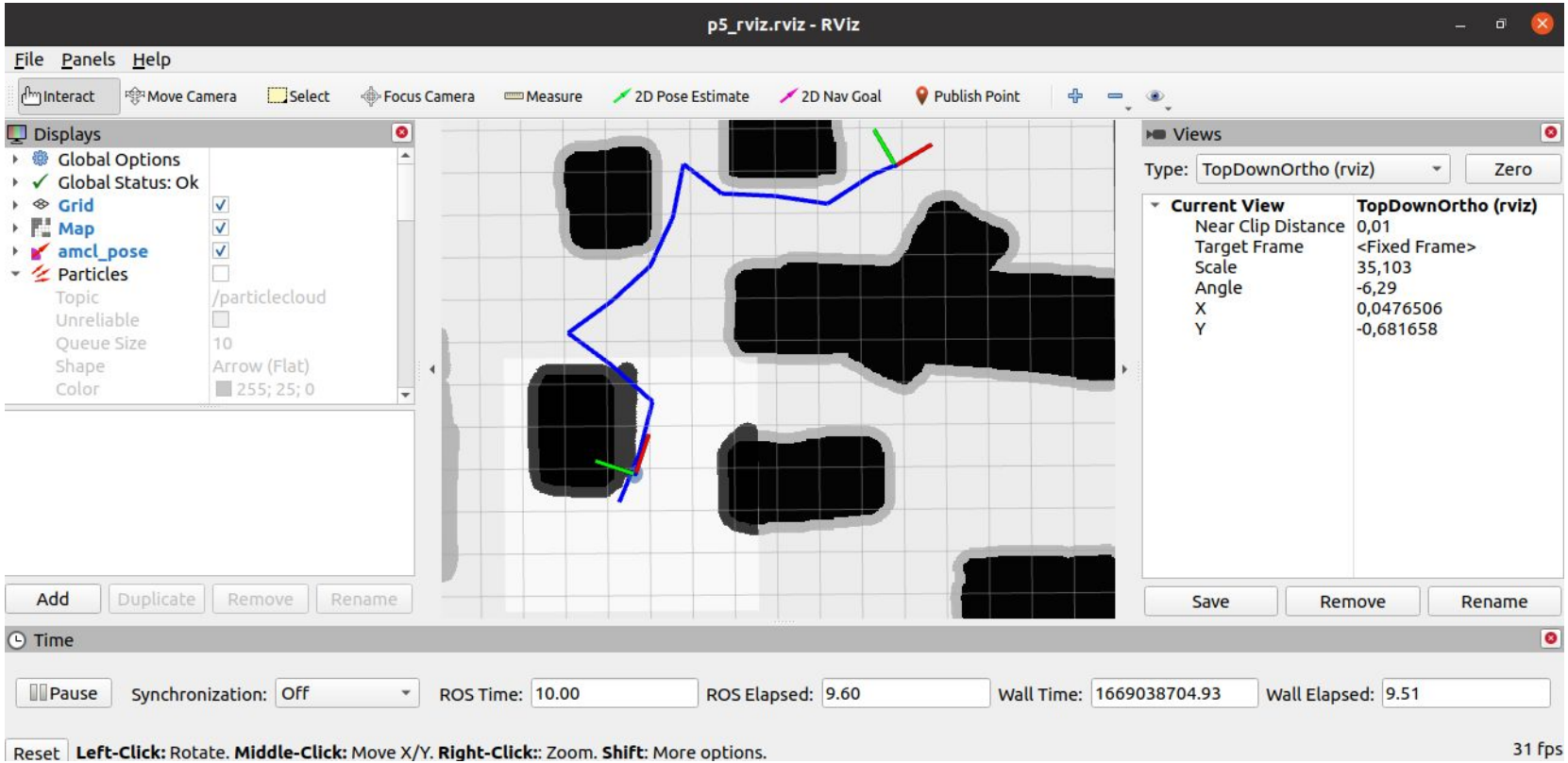
ROS Parameter Server

- Accessible via network
- Used by the nodes to store and retrieve parameters on configuration and during the execution
- Used for static constants (configuration parameters) that are globally viewable by any tool
    - Extensive use of namespaces
- Parameters can be defined:
    - In YAML files and can be loaded with `rosparam` command
    - Within a launch file (http://wiki.ros.org/rosparam)

Exercise 1

- Execute a ROS service available in the roscpp tutorial.

Implementation of the RRT algorithm



Exercise 2

- Download arob_lab5 package from github **https://github.com/luisriazuelo/arob_lab5.git**
- Complete the implementation of the RRT algorithm in file **rrt_global_planner.cpp**
- Send a goal through RViz or publishing a message on the topic **/move_base_simple/goal**

Remember that this global planner should be registered as a plugin in ROS to be used in move_base.

Exercise 3

- Test the system using also the plugin of your low level controller (implemented in lab4) as the local planner in move_base.

Laboratory 5 evaluation

- **Submit** the **code** for all exercises. Send the complete *arob_lab5* package before the beginning of the next session.

- **Multiple-choice test** through Moodle at the **beginning** of the **next session**. Test will be conducted **individually**, **without** any **extra material.**