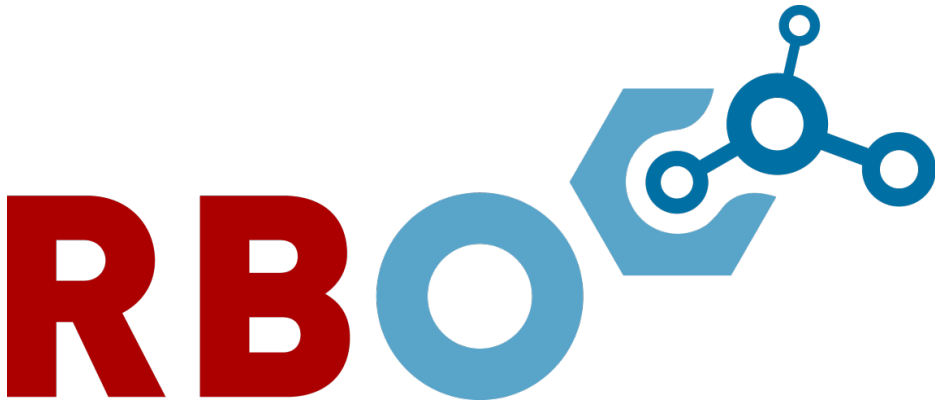


Introduction to ROS

Technische Universität Berlin

ROS: Topics

/openni/rgb/image_raw
/openni/depth/image_raw
...

/scan

/odom

ROS ***topics*** contain ***messages*** of a certain type

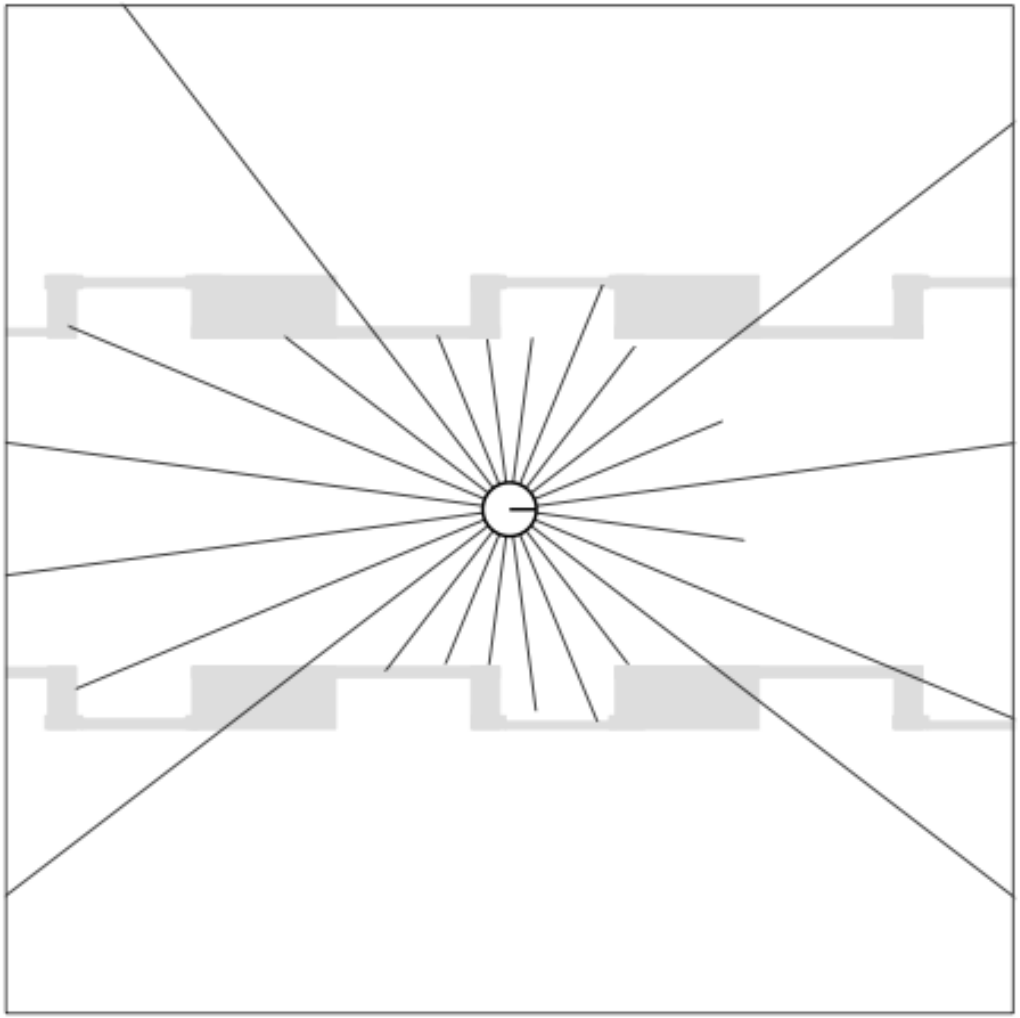
A middle-ware for robots

- Open-source, C++ and Python
- Framework
- Drivers, Hardware abstraction
- Contributors from all over the world

http://www.ros.org/

Tutorials: http://www.ros.org/wiki/ROS/Tutorials

Example: Laser Scanner

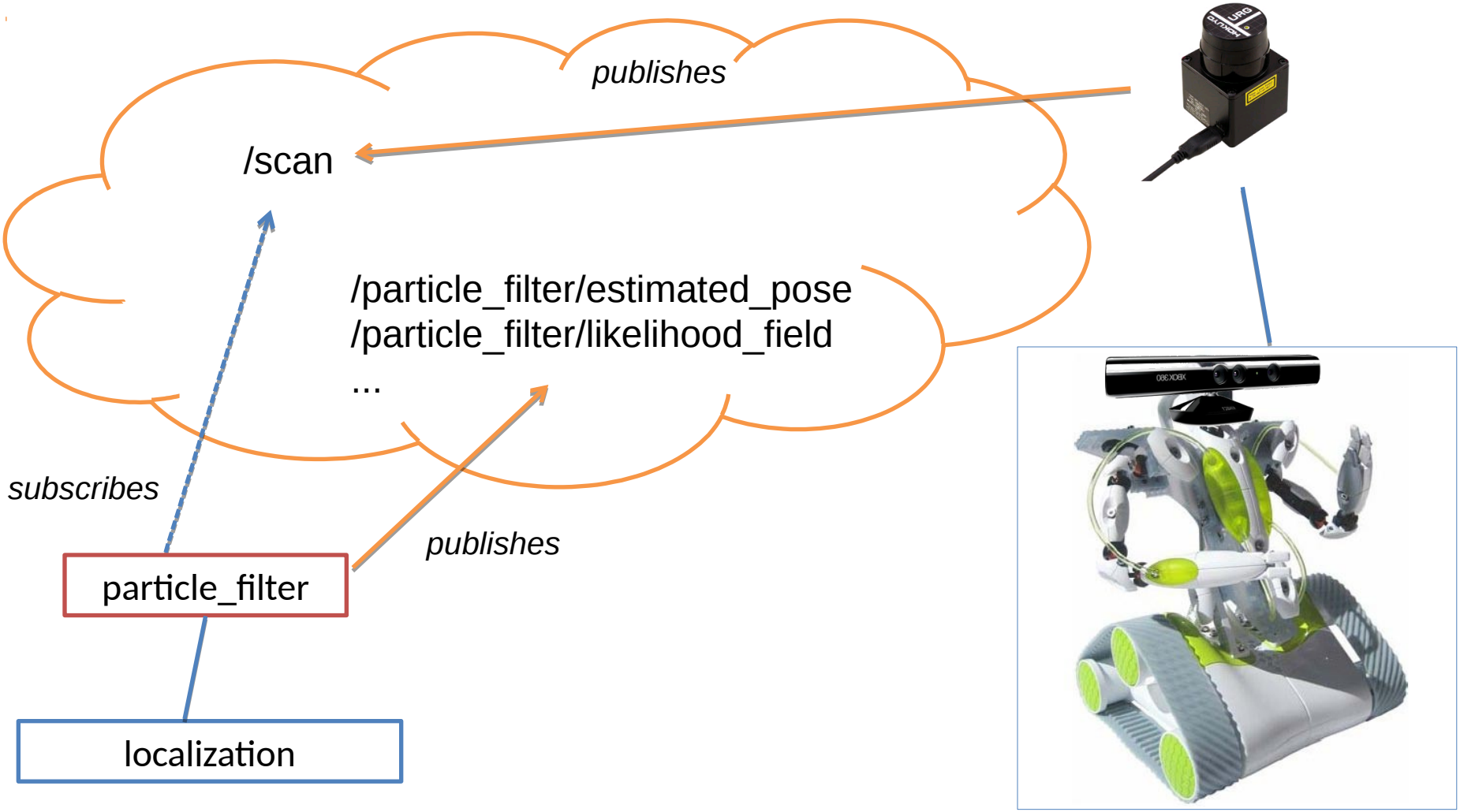


ROS: Messages

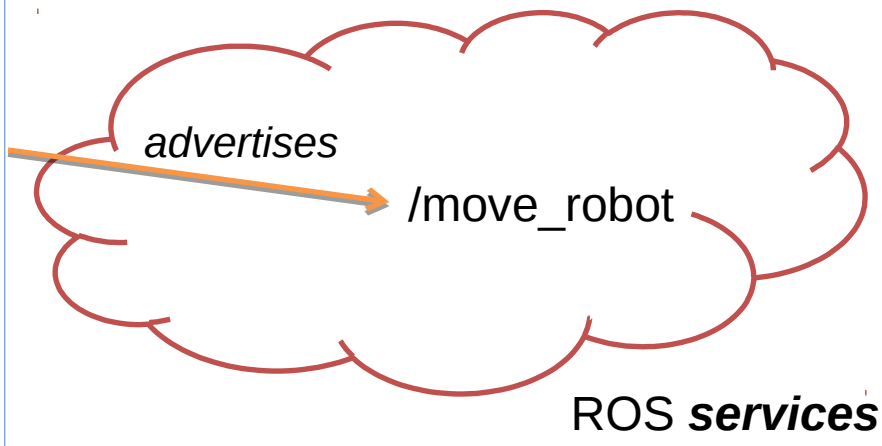
http://www.ros.org/doc/api/sensor_msgs/html/msg/LaserScan.html:

```
# ...
float32 angle_min          # start angle of the scan [rad]
float32 angle_max          # end angle of the scan [rad]
float32 angle_increment    # angular distance between
                # measurements [rad]

float32 range_min          # minimum range value [m]
float32 range_max          # maximum range value [m]

float32[] ranges           # range data [m] ...
```

ROS: Subscribing to Topics



ROS: Services

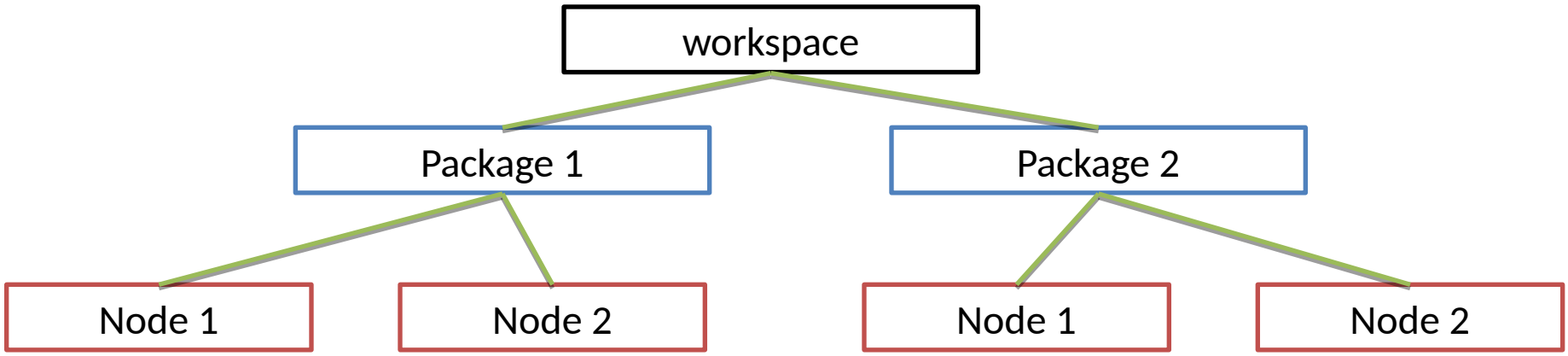



```
user@pc$:~/ rosservice call /move_robot 5.4 3.0
```

- Services are very similar to messages
- „Remote procedure call“: Services can have return values

ROS: Structural concepts

- A **package** contains **nodes**, **messages**, and **services**
- A **node** publishes+subscribes to topics, offers+calls services
- [A **workspace** contains packages (and meta-packages)]



ROS AND YOU

Installing ROS on your computer (native)

- Download, install and configure ROS kinetic, option „desktop full“
  Ubuntu (native): http://wiki.ros.org/kinetic/Installation

- Install additional packages
  - `sudo apt-get update`
  - `sudo apt-get install ros-kinetic-navigation ros-kinetic-slam-gmapping ros-kinetic-rviz ros-kinetic-roslib libwxgtk3.0-de`

Using a ROS Workspace

- Get our ROS workspace from ISIS and unpack it locally (e.g. in `/home/user/assignment4`)
- Setup your workspace with:
  - catkin_make -j1

  (catkin is a wrapper for cmake)
- Setup your shell to use the workspace:
  - source devel/setup.sh

  (you can also append this to your ~/.bashrc)

  see also:
  http://www.ros.org/wiki/ROS/Tutorials/InstallingandConfiguringROSEnvironment

- Build all packages in your Workspace:
  - catkin_make

RBO

TU berlin

ROS COMMANDS

How to start a ROS network

- start server with **`roscore`**
- nodes connect to the roscore at `$ROS_MASTER_URI`
  - Default: http://localhost:11311
  - To connect remotely to a roscore on another machine/robot: **`ROS_MASTER_URI=destination-url:11311`**
- **`rosrun`**: run a node (executable) from a certain package
  - **`rosrun turtlesim turtlesim_node`**

ROS Messages

- **`rostopic`**
  - list: Display a list of current topics
  - echo <topic-name> : Display messages
    - **`rostopic echo /turtle1/pose/`**
  - type <topic-name> : Display message type
    - **`rostopic type /turtle1/pose/`**

- **`rosmsg`**
  - show <msg-name> : Display a message definition
    - **`rosmsg show turtlesim/Pose`**
  - Online documentation: http://docs.ros.org/api/

ROS Service

- **`rosservice`**

  (service counterpart to rostopic)

  - list: Display a list of available services
  - call <srv-name> <params>: Call a ROS service
    - `rosservice call /turtle1/teleport_relative 3.0 1.0`

- **`rossrv`**

  (service counterpart to rosmsg)

  - show <srv-name>: Display a service definition
    - **`rossrv show turtlesim/TeleportRelative`**

ROS File System Tools

- **`roscd`**
  - Jump immediately to a package/stack directory
    `roscd stack-or-package[/subdir]`
  - If this doesn‘t work, make sure you sourced your workspace
    `source devel/setup.bash`

ROS Tools

- **`roslaunch`**: launches a set of nodes
  - defined by an XML configuration file
    - **`roslaunch [<node>] <launch-file>`**
  - *roscore* is started automatically (if not yet running)

- **`catkin_make`**: is a tool aware of ros packages dependencies

- **`rosparam`**: enables getting and setting parameter server values

ROS logging

- Log files are called *bag files*
- **`rosbag play <file>`**
  - Pause playing with space key and stepping with s
- **`rosbag record <topic-names>`**
  - will generate a ".bag" file
  - Use option –a to record all topics