

# Ein kurze Einführung in ROS

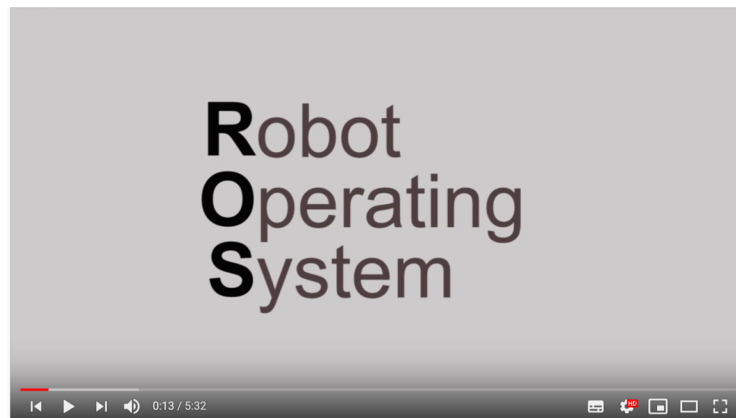
- Was ist ROS
- Interprozesskommunikation
- Werkzeuge:  
Visualisierung, Simulation, Transformationen, ...
- Pakete: Navigation, Mapping, ...
- Referenzen

# ROS Videos

---



<https://www.youtube.com/watch?v=mDwZ21Zia8s>

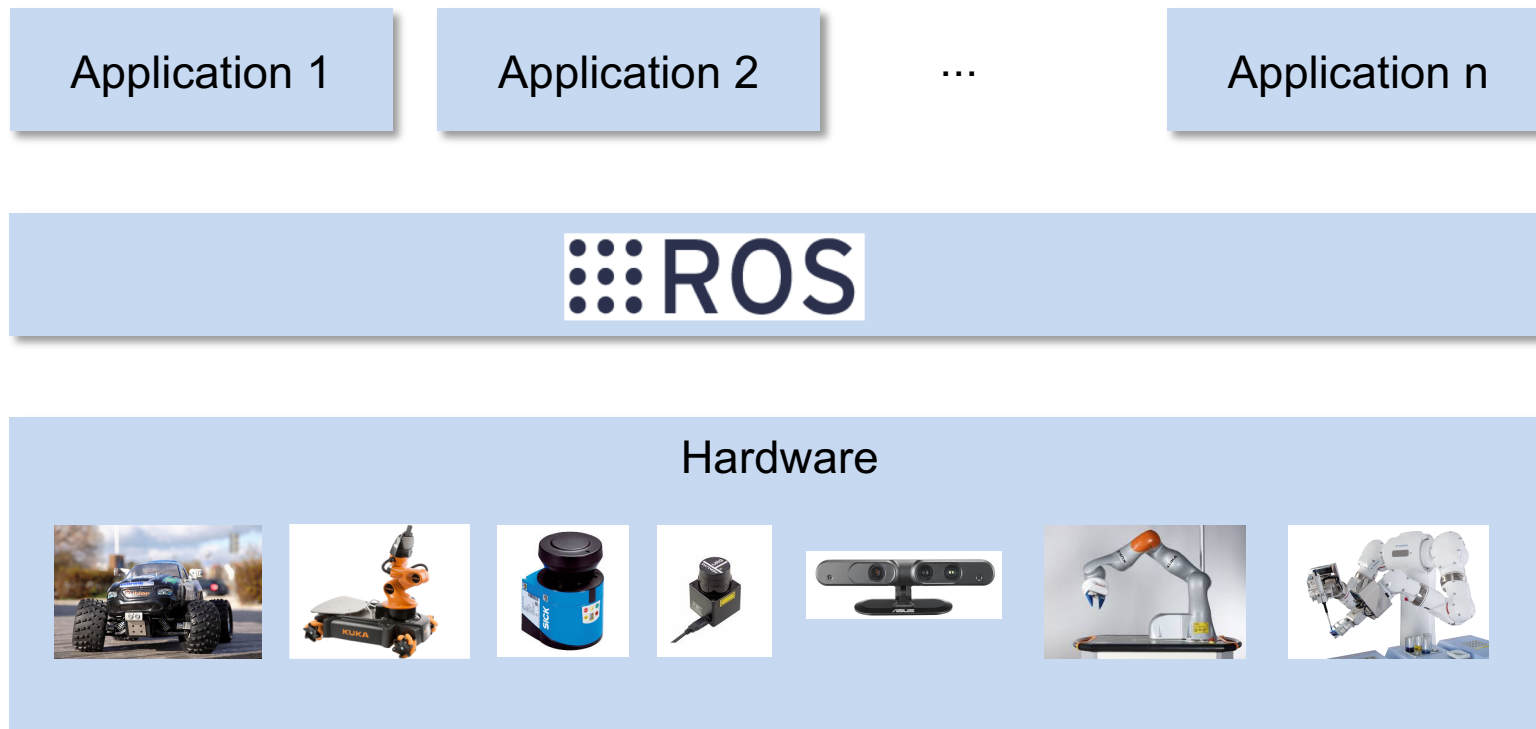


What is ROS (short description)

[https://www.youtube.com/watch?v=UL1\\_Ue4rUWs](https://www.youtube.com/watch?v=UL1_Ue4rUWs)

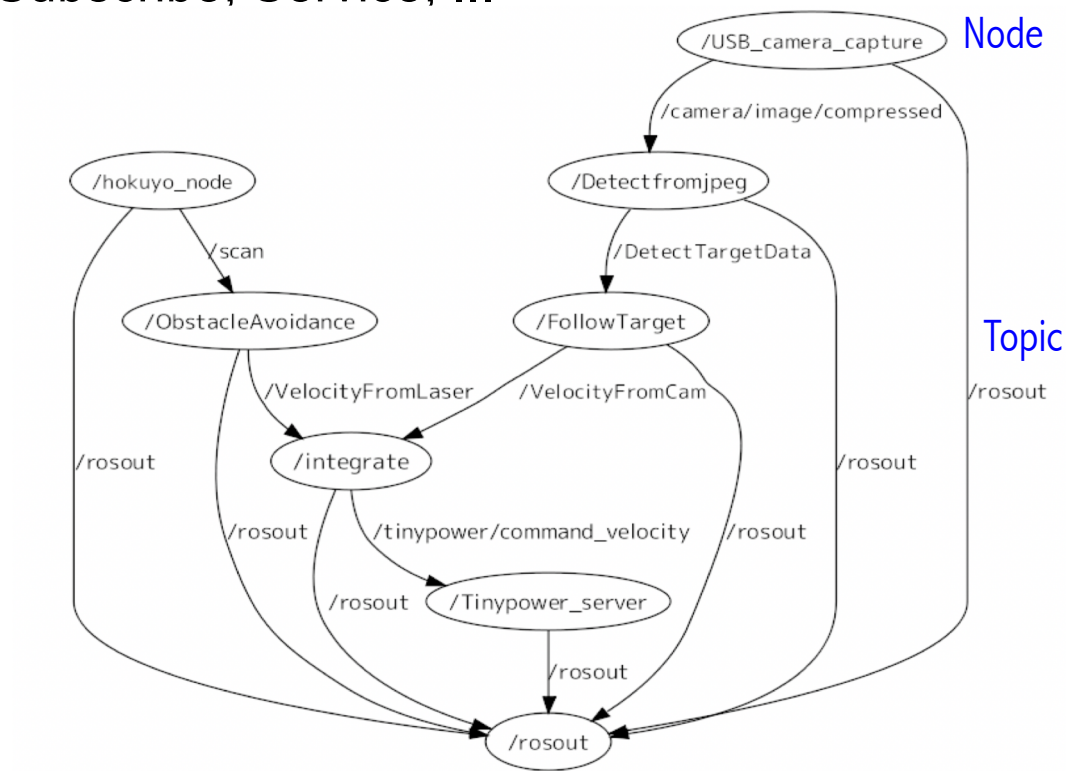
# Was ist ROS (1)

- ROS = Robot Operating System (eigentlich Meta-Betriebssystem)
- Hardwareabstraktion und Gerätetreiber für zahlreiche Roboterplattformen, Sensoren und Aktoren



# Was ist ROS (2)

- Eine ROS-Anwendung besteht aus einer Menge von Nodes (Komponenten), die miteinander über Topics kommunizieren (ROS Graph)
- ROS-Anwendung kann auch verteilt laufen
- ROS bietet dazu Kommunikationsplattform: Publish/Subscribe, Service, ...



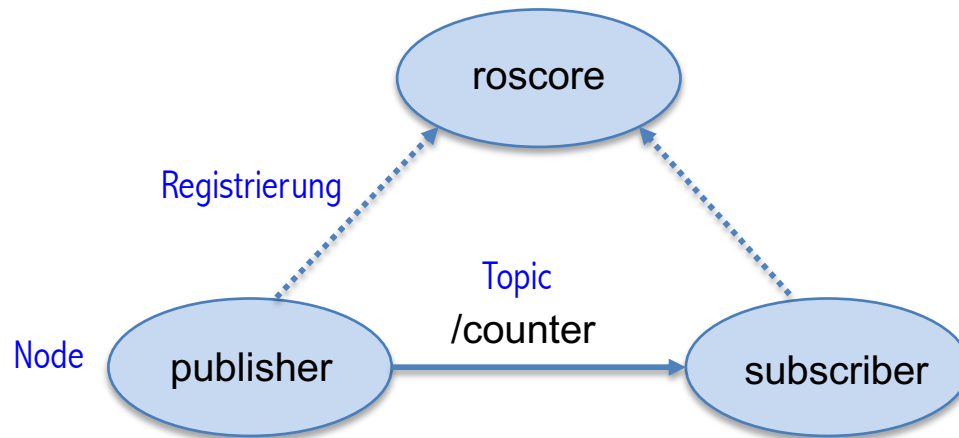


# Was ist ROS (3)

---

- Werkzeug-basiert
  - Erstellung von Paketen und Navigation durch Paketstrukturen
  - Launch-Konzept: Start mehrerer Nodes
  - Visualisierung der Interprozesskommunikation
  - Logging-Mechanismus
  - Plotten und Visualisierung von Datenströmen
- Robotersimulatoren
- Zahlreiche Bibliotheken und Pakete
- Plattform: Linux (Ubuntu, Debian) und Windows 10
- Mehrsprachig: Python, C/C++, Java, ...
- Open Source
- Große Community

# Publish/Subscribe Konzept



```
user@hostname$ roscore
user@hostname$ rosrun basics publisher.py
user@hostname$ rosrun basics subscriber.py
355
356
357
358
359
360
```

```
rospy.init_node('publisher')
pub = rospy.Publisher('counter', Int32)

rate = rospy.Rate(2)
count = 0

while not rospy.is_shutdown():
    pub.publish(count)
    count += 1
    rate.sleep()
```

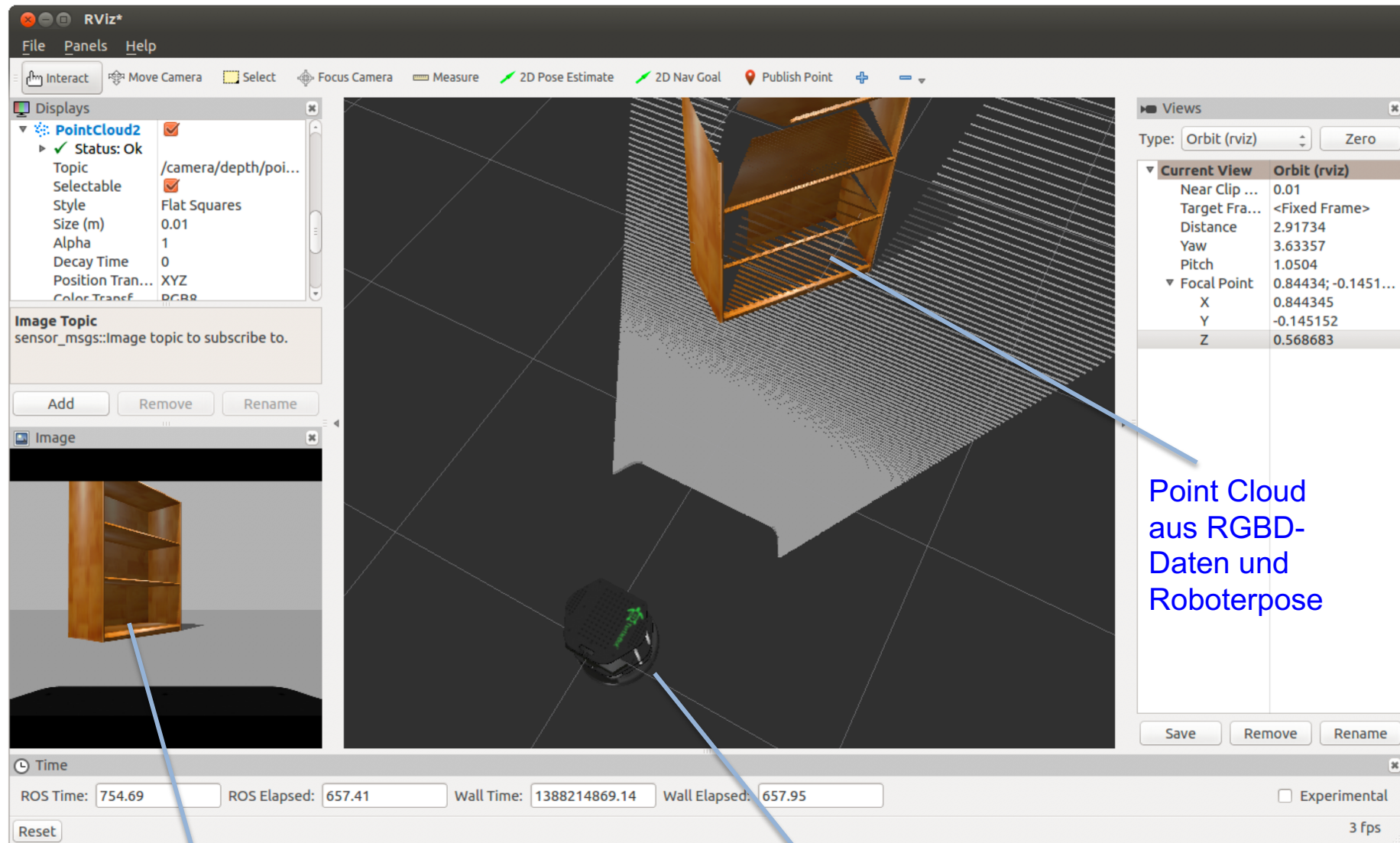
publisher.py

```
def callback(msg):
    print msg.data

rospy.init_node('subscriber')
sub = rospy.Subscriber('counter', Int32, callback)
rospy.spin()
```

subscriber.py

# Visualisierung mit rviz (1)



Point Cloud  
aus RGBD-  
Daten und  
Roboterpose

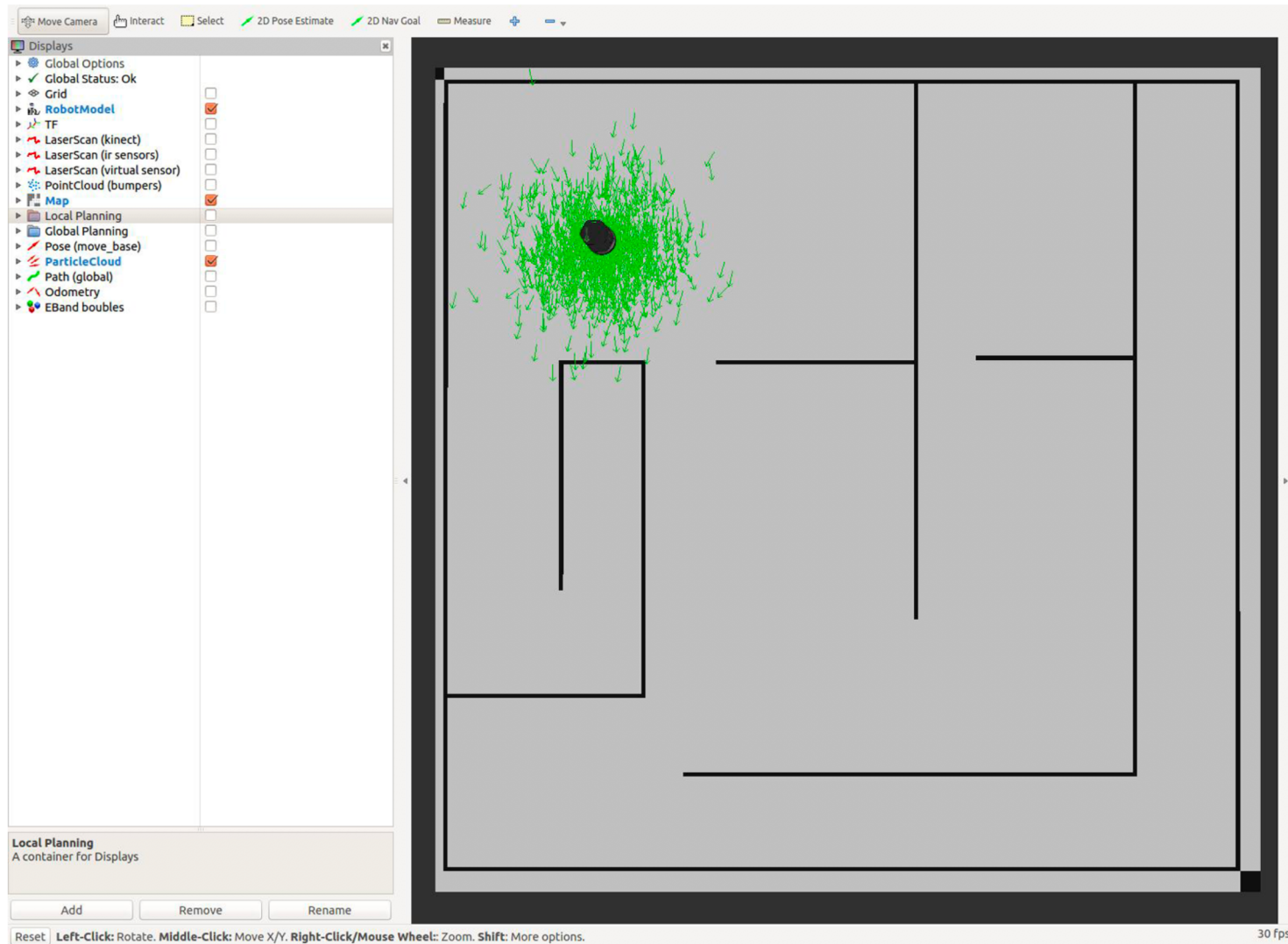
Kamerabild

Turtlebot (rviz kennt Roboterbeschreibung)

Quickley u.a., Programming Robots with ROS, 2015

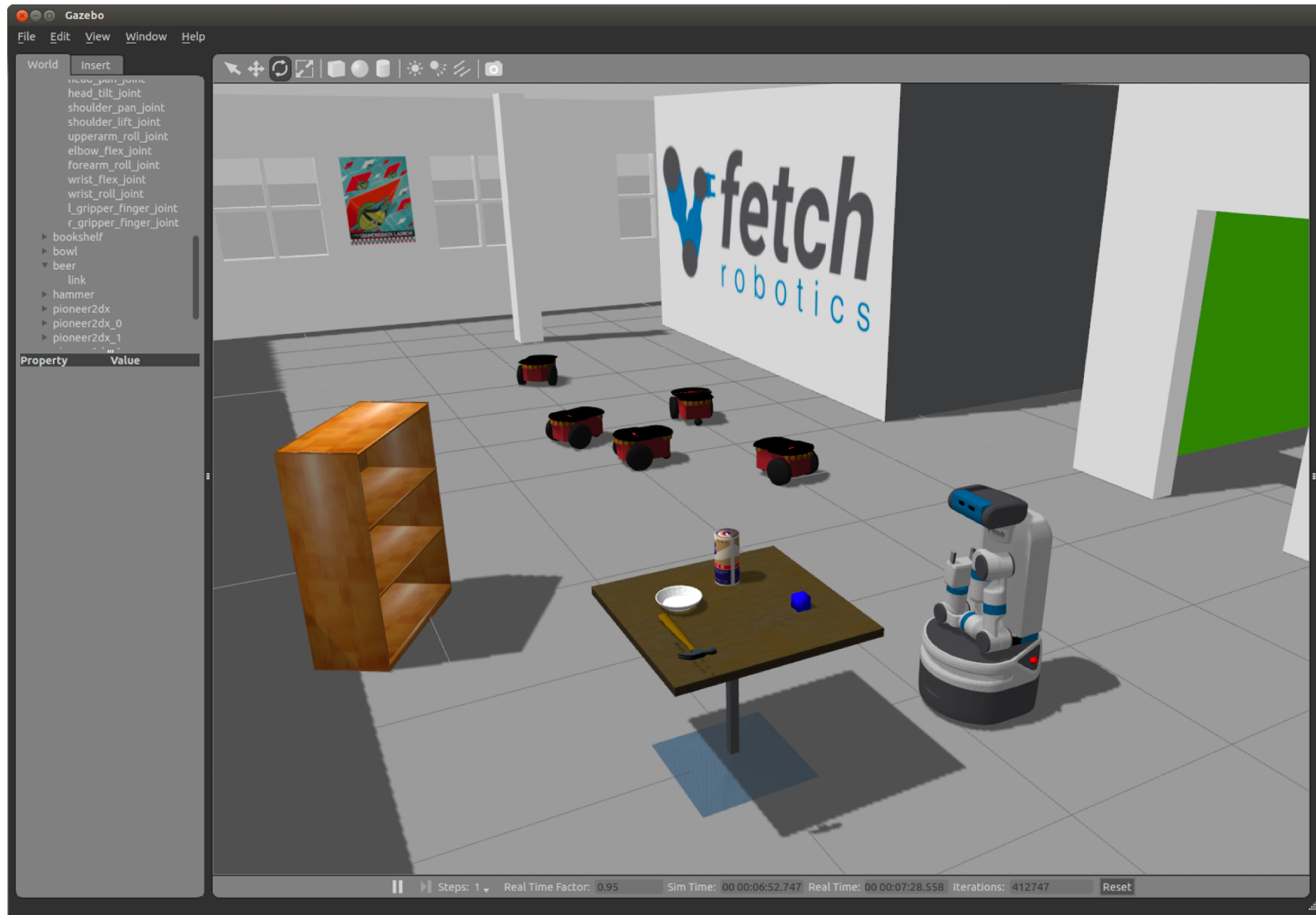
# Visualisierung mit rviz (2)

Lokalisierung von Turtlebot mit einem Partikelfilter und einer bereitgestellten CAD-Karte.



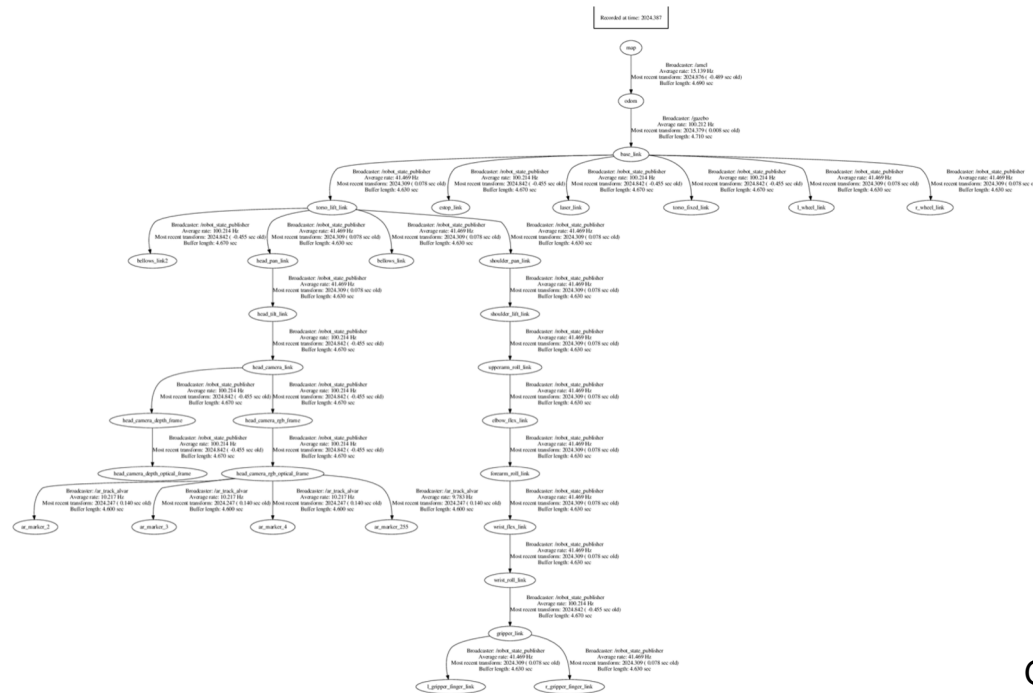
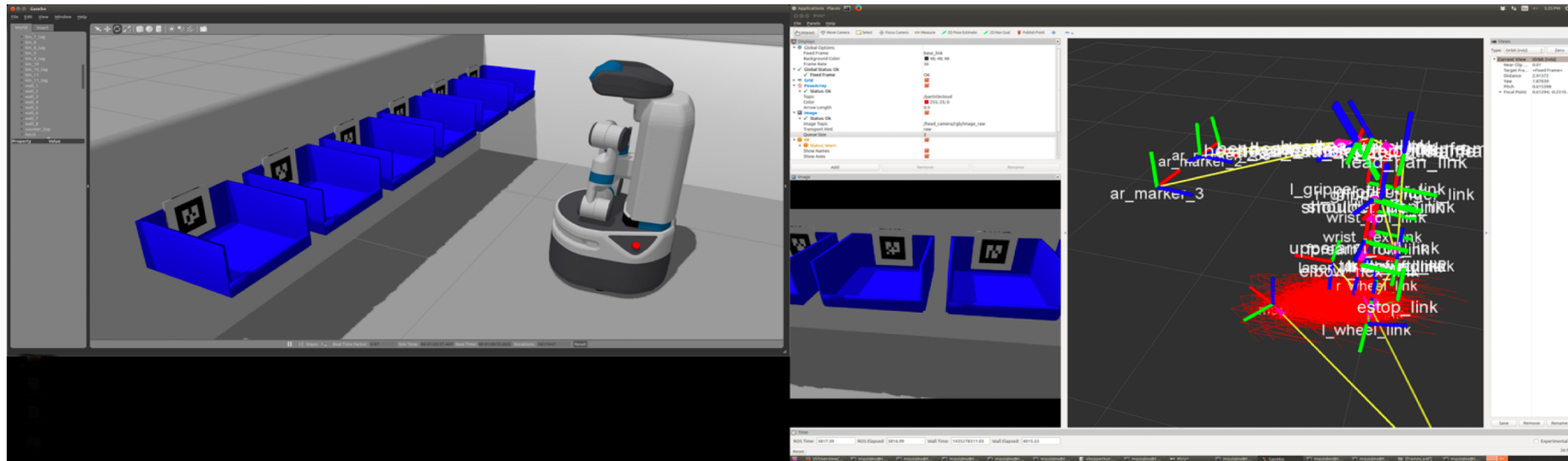
Quickley u.a., Programming Robots with ROS, 2015

# Simulation mit gazebo



Quickley u.a., Programming Robots with ROS, 2015

# ROS tf: Verwaltung von KS-Transformationen



- Transformationsbaum
- Knoten: KS
- Kante: Transformation mit Zeitstempel, Rate und Herausgeber (broadcaster)

Quickley u.a., Programming Robots with ROS, 2015



# ROS tf: Beispiel

## tf Listener

```
import rospy
import tf

listener = tf.TransformListener()

(trans,rot) = listener.lookupTransform('world', 'robot_base', rospy.Time(0))

angles = tf.transformations.euler_from_quaternion(rot, 'sxyz')
```

- Hole von tf aktuellste Transformation von world nach robot\_base
- Translation trans = (x,y,z)
- Rotation rot = (x,y,z,w) als Quaternion
- angles: Eulerwinkel im xyz-Drehsystem

## tf Broadcaster

```
import rospy
import tf

x = ...
y = ...
theta = ...

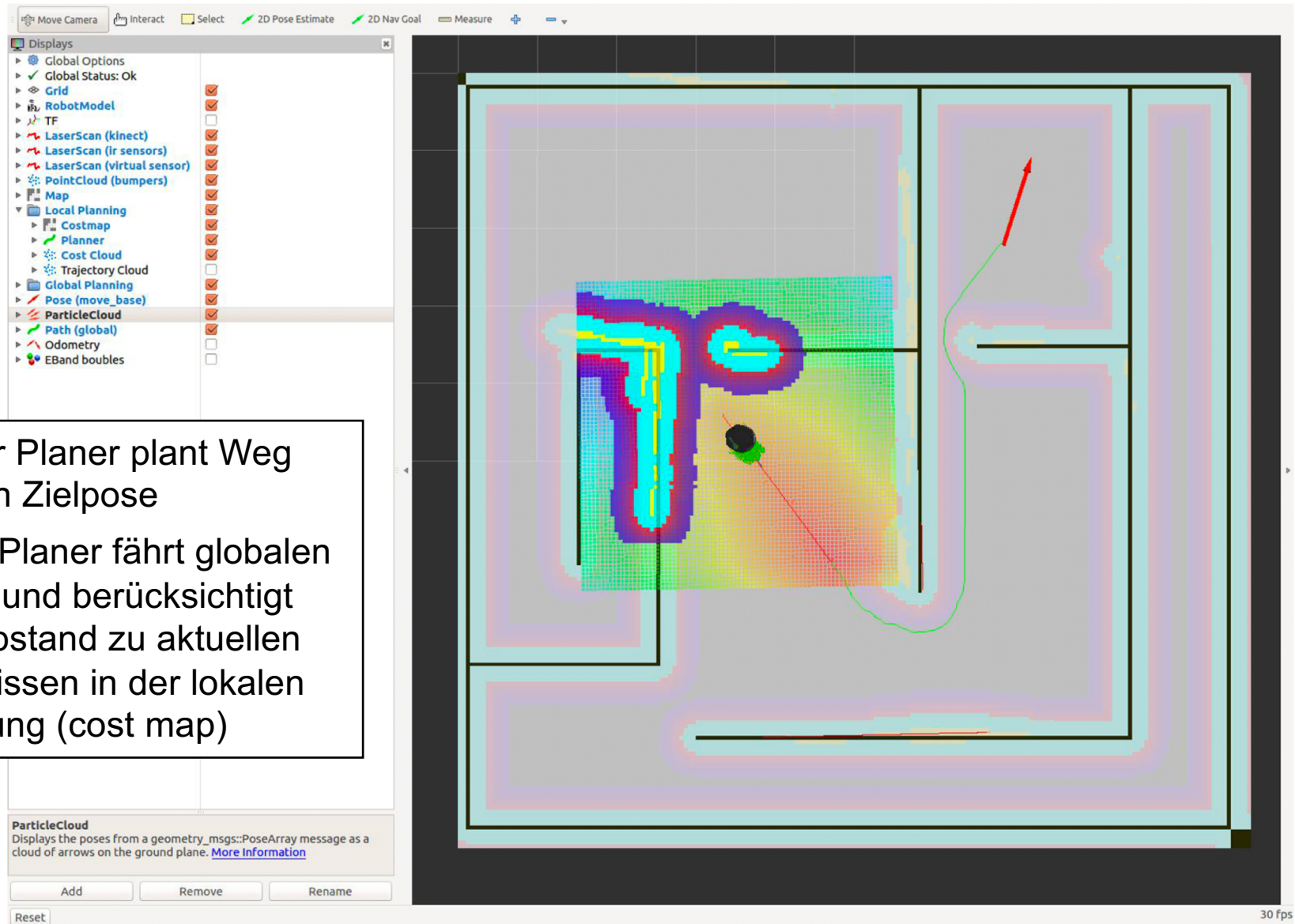
trans = (x, y, 0)
rot = tf.transformations.quaternion_from_euler(0, 0, theta, 'sxyz'),

broadcaster = tf.TransformBroadcaster()

broadcaster.sendTransform(trans, rot, rospy.Time.now(), 'robot_base', 'world')
```

- Veröffentliche auf tf Transformation von world nach robot\_base

# ROS Navigation Stack (1)

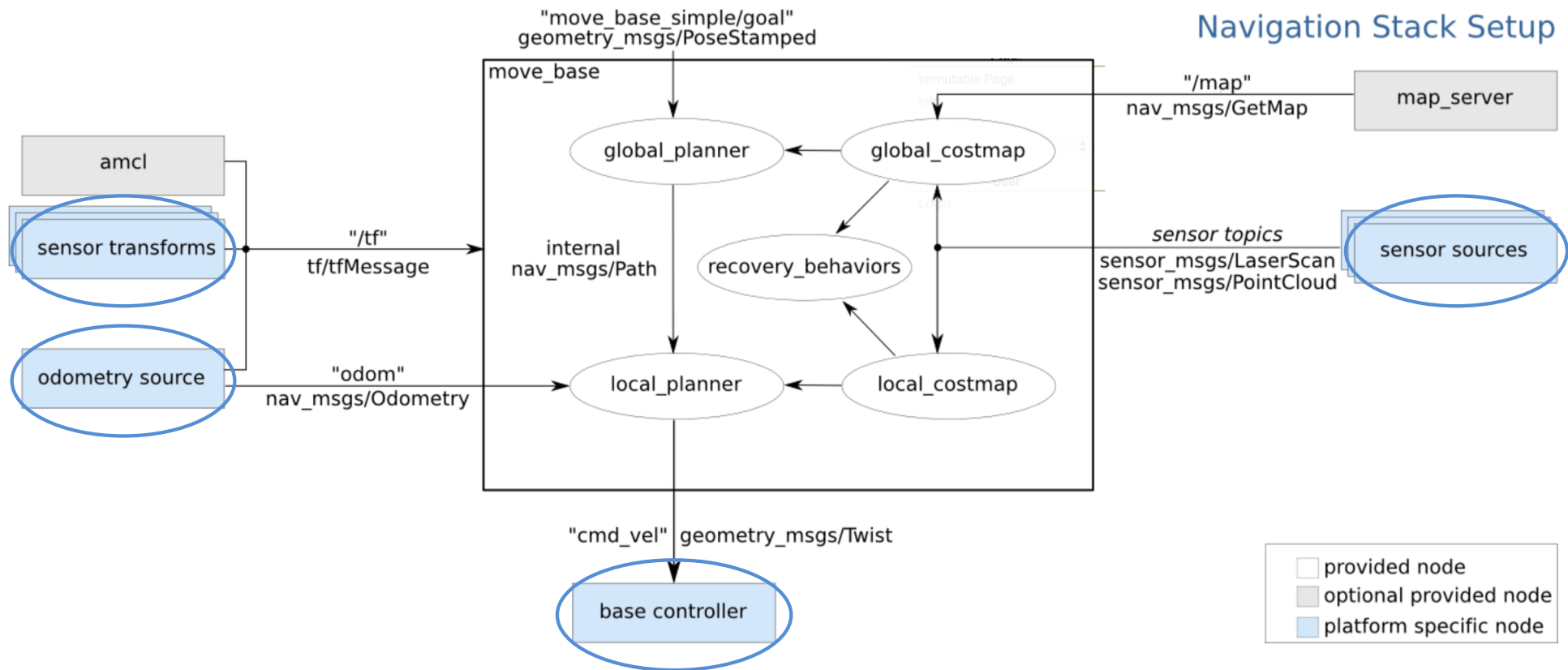


- Globaler Planer plant Weg zur roten Zielpose
- Lokaler Planer fährt globalen Plan ab und berücksichtigt dabei Abstand zu aktuellen Hindernissen in der lokalen Umgebung (cost map)

Quickley u.a., Programming Robots with ROS, 2015



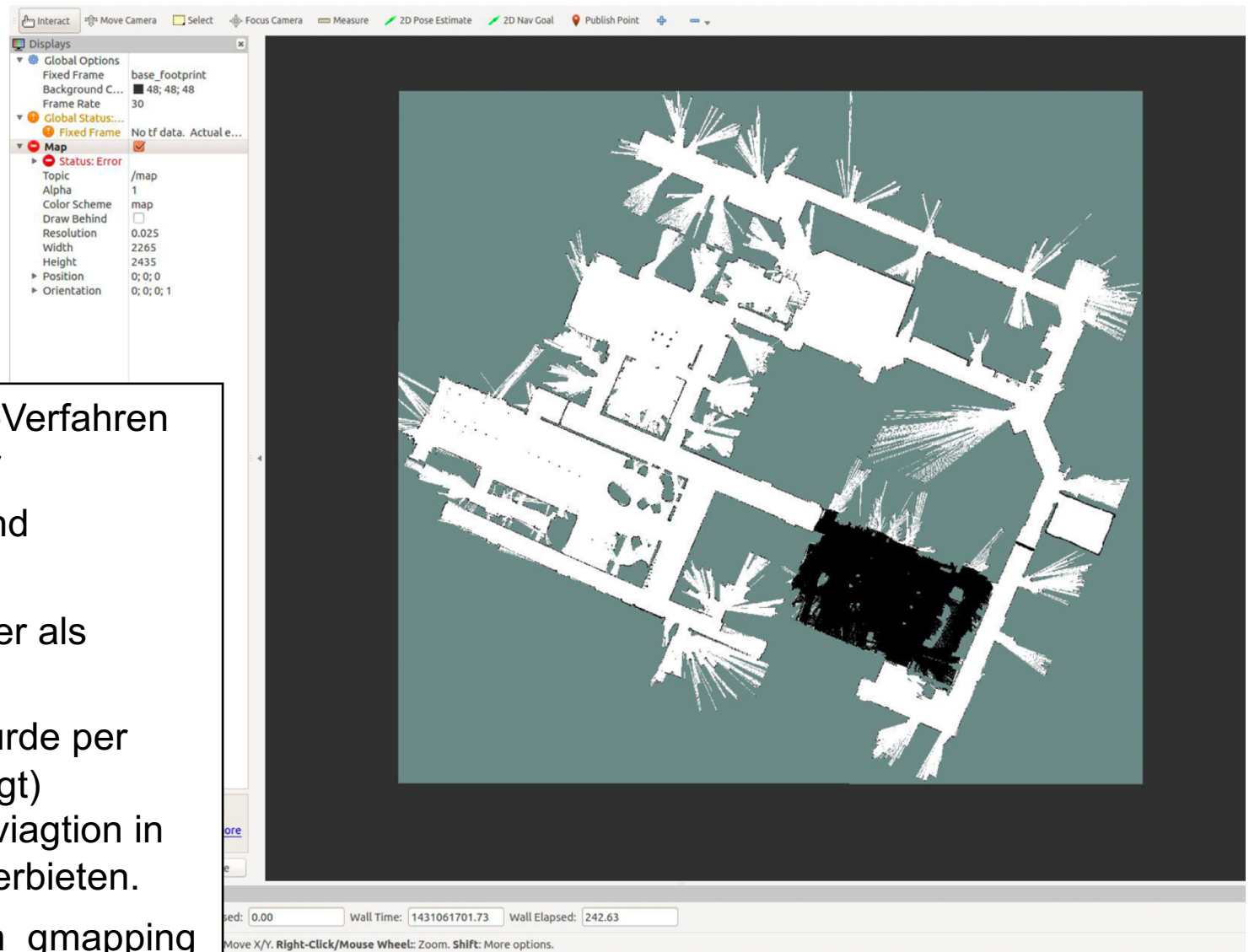
# ROS Navigation Stack (2)



<http://wiki.ros.org/navigation>

- Wenige plattformspezifische Teile sind bereit zu stellen
- Roboter, der Geschwindigkeitsbefehle abonniert (`cmd_vel`)
- Roboter, der Odometriedaten veröffentlicht (`odom` + `tf`)
- Sensor-Pose in `tf` veröffentlichen
- Laser oder RGBD-Sensor, der Sensordaten veröffentlicht (`sensor_topic`)

# Kartierung mit ROS gmapping

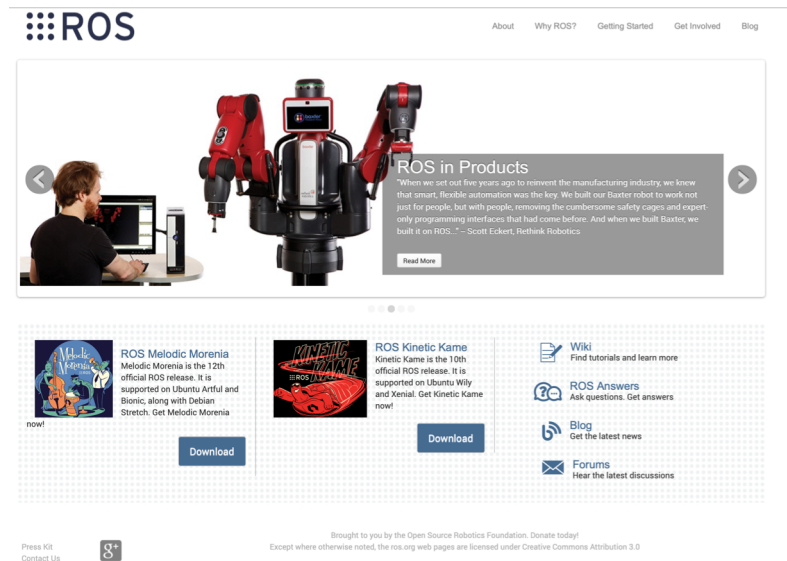


- Gitterbasiertes SLAM-Verfahren mit einem Partikelfilter
- Benötigt Odometrie und Laser-Scans
- Erstellt Belegtheitsgitter als png-Datei.
- Teil der Umgebung wurde per Hand schwarz (= belegt) übermalt, um eine Navigation in diesem Bereich zu verbieten.
- [http://wiki.ros.org/slam\\_gmapping](http://wiki.ros.org/slam_gmapping)

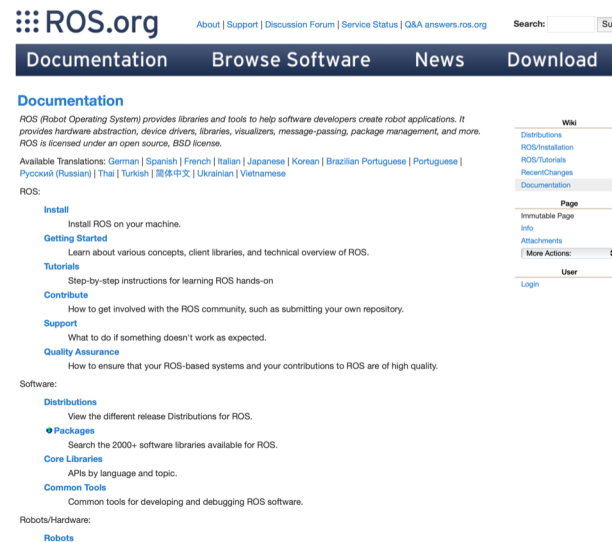
Quickley u.a., Programming Robots with ROS, 2015

# Referenzen (1)

- ROS: <https://www.ros.org>

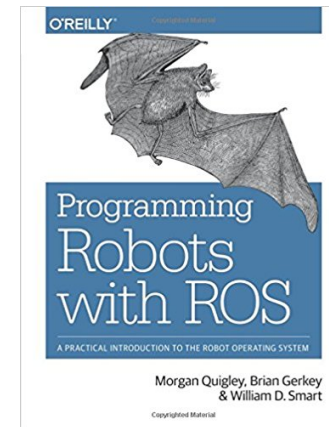


- ROS wiki: <http://wiki.ros.org>



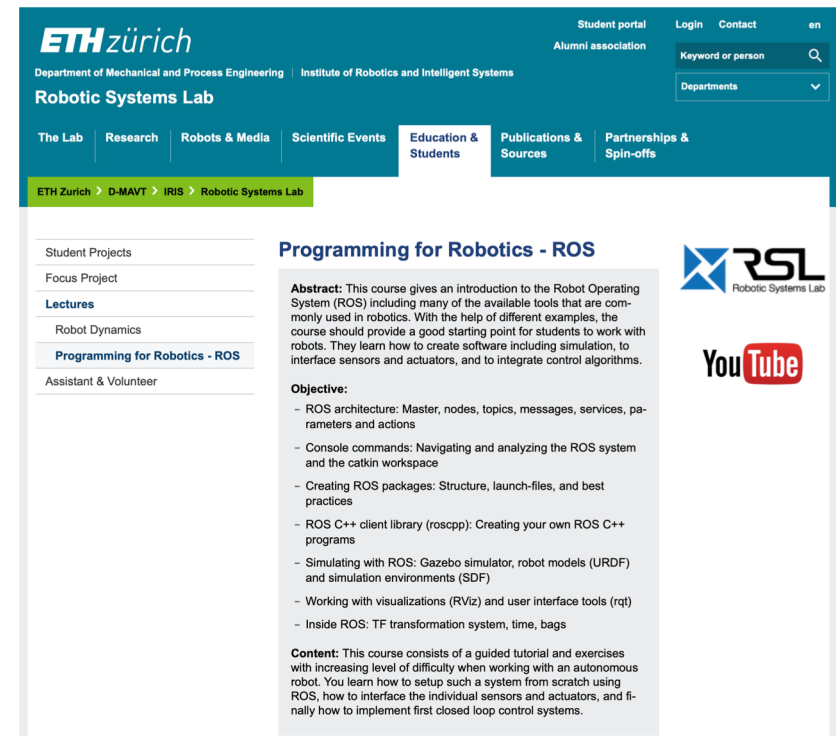
# Referenzen (2)

- Morgan Quigley, Brian Gerkey, William Smart, *Programming Robots with ROS*, O'Reilly, 2015
- als E-Book im Rahmen der Vorlesung ausleihbar
- Sehr gute Einführung in ROS
- Beispiele mit Python



- Sehr guter ROS-Kurs in C/C++ (mit Videos, Folien und Übungen)

<https://rsl.ethz.ch/education-students/lectures/ros.html>



**ETH zürich** Student portal Login Contact en  
Department of Mechanical and Process Engineering | Institute of Robotics and Intelligent Systems  
Alumni association Keyword or person Departments

**Robotic Systems Lab**

The Lab Research Robots & Media Scientific Events Education & Students Publications & Sources Partnerships & Spin-offs

ETH Zurich > D-MAVT > IRIS > Robotic Systems Lab

Student Projects  
Focus Project  
Lectures  
Robot Dynamics  
**Programming for Robotics - ROS**  
Assistant & Volunteer

### Programming for Robotics - ROS

**Abstract:** This course gives an introduction to the Robot Operating System (ROS) including many of the available tools that are commonly used in robotics. With the help of different examples, the course should provide a good starting point for students to work with robots. They learn how to create software including simulation, to interface sensors and actuators, and to integrate control algorithms.

**Objective:**

- ROS architecture: Master, nodes, topics, messages, services, parameters and actions
- Console commands: Navigating and analyzing the ROS system and the catkin workspace
- Creating ROS packages: Structure, launch-files, and best practices
- ROS C++ client library (roscpp): Creating your own ROS C++ programs
- Simulating with ROS: Gazebo simulator, robot models (URDF) and simulation environments (SDF)
- Working with visualizations (RViz) and user interface tools (rqt)
- Inside ROS: TF transformation system, time, bags

**Content:** This course consists of a guided tutorial and exercises with increasing level of difficulty when working with an autonomous robot. You learn how to setup such a system from scratch using ROS, how to interface the individual sensors and actuators, and finally how to implement first closed loop control systems.

**RSL** Robotic Systems Lab  
**YouTube**

# Referenzen (3)

- „ROS-Industrial is an open-source project that extends the advanced capabilities of ROS software to manufacturing.“
- <https://rosindustrial.org/>



ROS-Industrial is an open-source project that extends the advanced capabilities of ROS software to manufacturing.

Interested in learning more about or even [joining](#) the ROS-Industrial Consortium? Start over at the [Consortium FAQ](#), or you can find the respective region agreements below "Consortium" in the banner above!




ROS-INDUSTRIAL HAD THE OPPORTUNITY TO COLLABORATE WITH MICROSOFT AND BMW TO ADDRESS A BURNING NEED FOR AGILE AND FLEXIBLE LOGISTICS AUTOMATION SOLUTIONS THAT CAN INTEROPERATE EFFICIENTLY AND BE DEPLOYED AT SCALE. WE HOPE YOU FIND THIS A COMPELLING EXAMPLE OF OPEN-SOURCE DELIVERING END USER VALUE AND HOW RESEARCHERS AND FOR-PROFITS CAN WORK TOGETHER TO SOLVE BIG PROBLEMS!


## ROS-INDUSTRIAL

ROS-Industrial is an open source project that extends the advanced capabilities of the [Robot Operating System \(ROS\)](#) software to manufacturing.

## Upcoming Events

 [ROS-Industrial Training \(EU\)](#)

Sep 23, 2019 – Sep 27, 2019

 [ROS-Industrial Training \(APAC\)](#)

Sep 24, 2019 – Sep 27, 2019

 [ROS-Industrial Training \(Americas\) October 2019](#)

Oct 8, 2019 – Oct 10, 2019

 [ROS-Industrial EU Fall'19 Workshop](#)

Oct 9, 2019 – Oct 10, 2019

 [Industrial Transformation ASIA PACIFIC](#)

Oct 22, 2019 – Oct 24, 2019