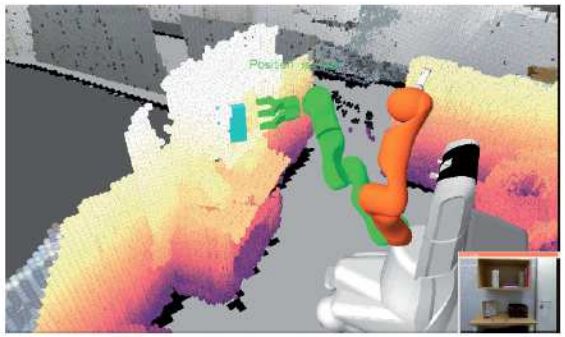
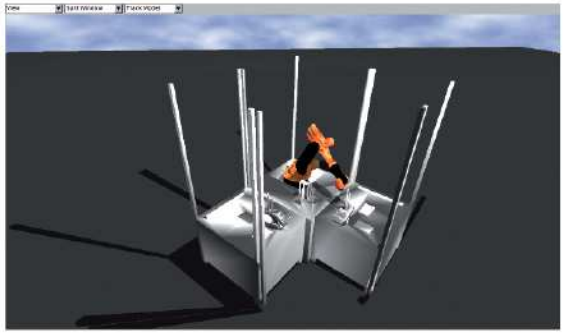
ROS INTRODUCTION

Dipl.-Ing. Florian Weißhardt, Fraunhofer IPA

Technology Seminar – ROS in Industrial Applications



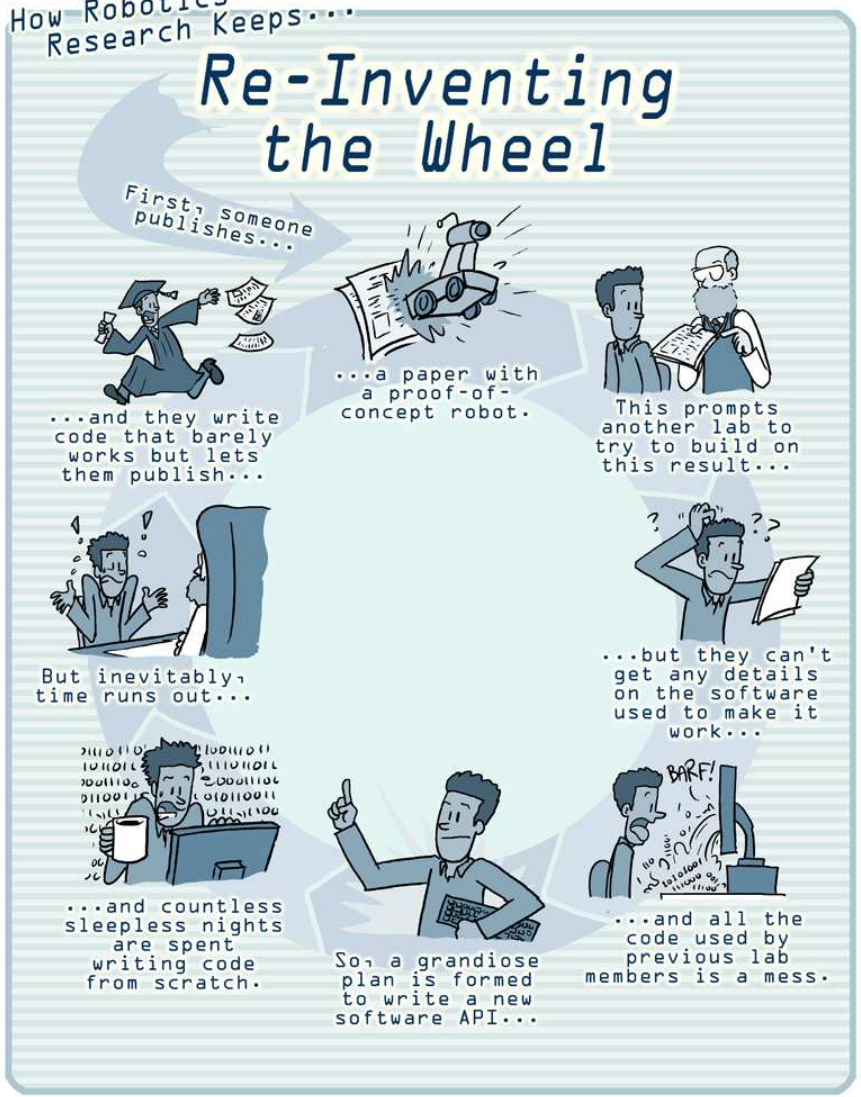


Fraunhofer IPA

Research in robotics

- Reinvention of the Wheel
- Little Commonality
- Short Lifespan
- Inability to Compare Results

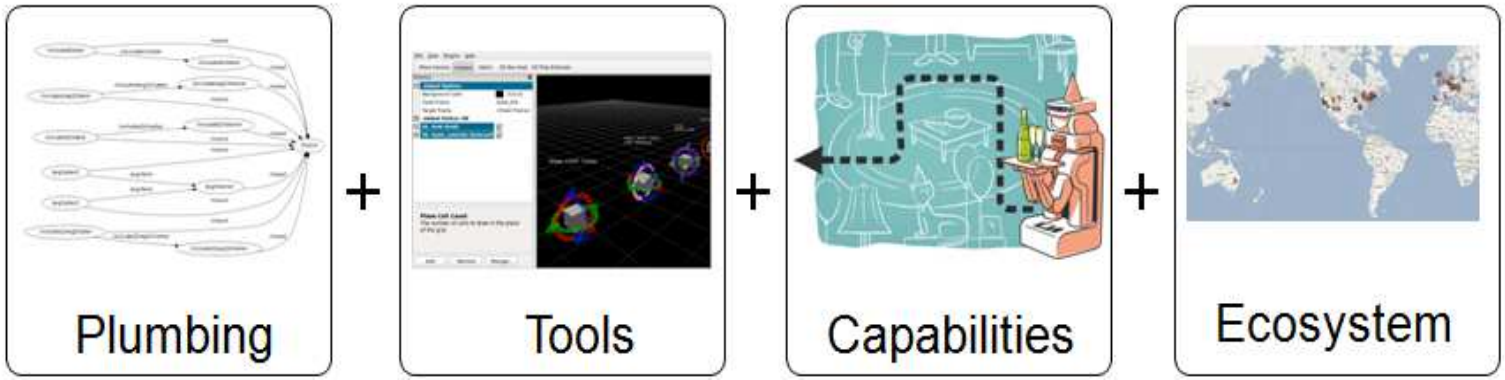
→ ROS addresses these



Fraunhofer IPA

ROS – Robot Operating System

- ROS = **R**obot **O**perating **S**ystem
- „ROS is an open-source, meta-operating system for your robot.“ [ROS-wiki]
- ROS is a “robot framework” [ROS-wiki]



www.ros.org/wiki/ROS/Introduction

Fraunhofer IPA

ROS – Video

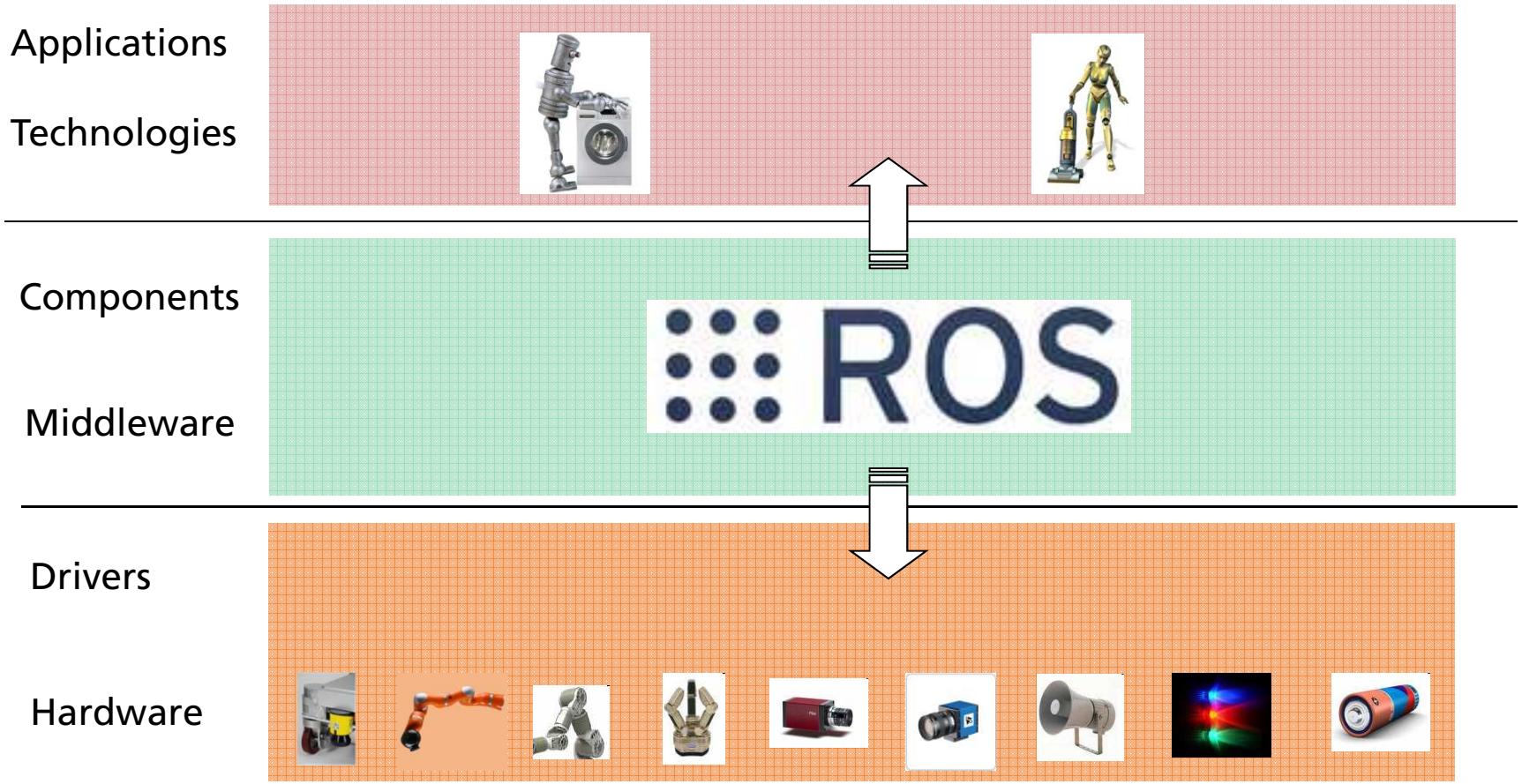
- 5 years of ROS

http://youtu.be/PGaXiLZD2KQ

ROS – Robot Operating System

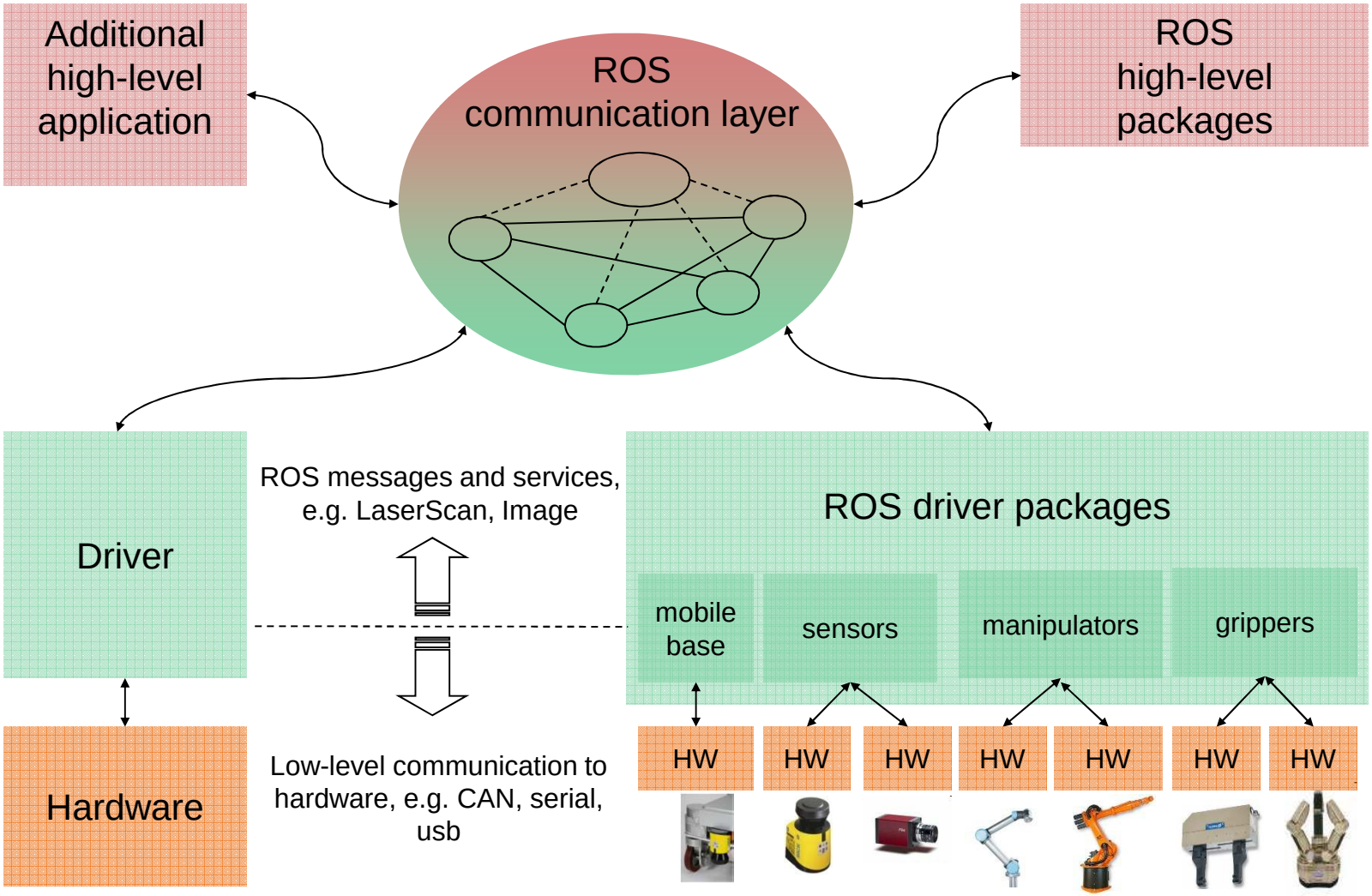
- What is ROS?
- Provides
  - Hardware abstraction
  - Low-level device control
  - Communication layer with message-passing between processes
  - Recursive package management and build system
  - Runs primarily on Linux but is intended to be cross-platform compatible to MAC OS X and Windows
- Content
  - ROS core build and runtime system
  - ROS packages, a collection of robotic algorithms

Layered architecture



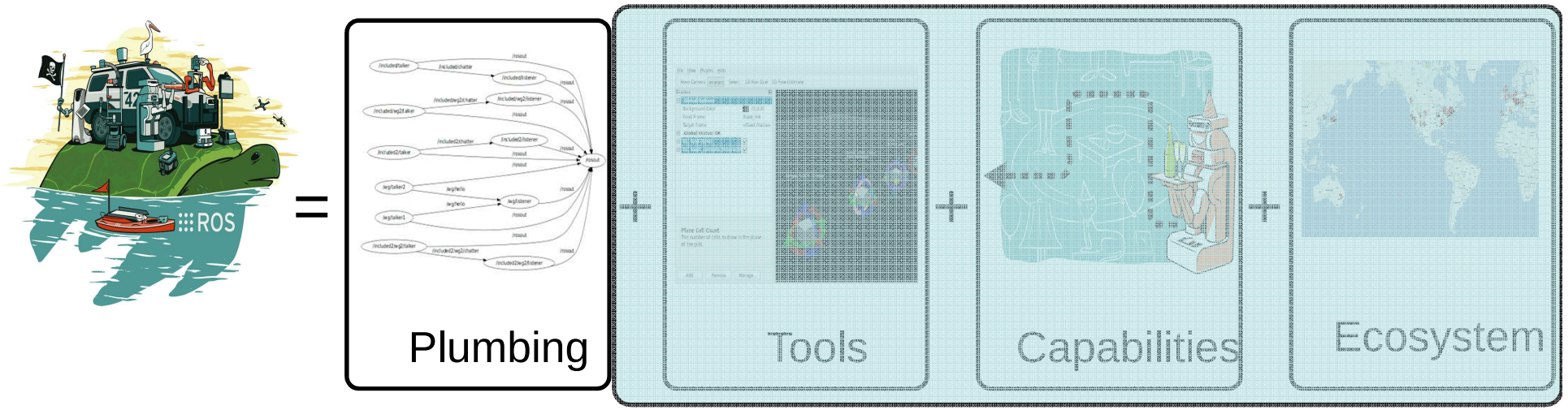
Fraunhofer IPA

Control Structure

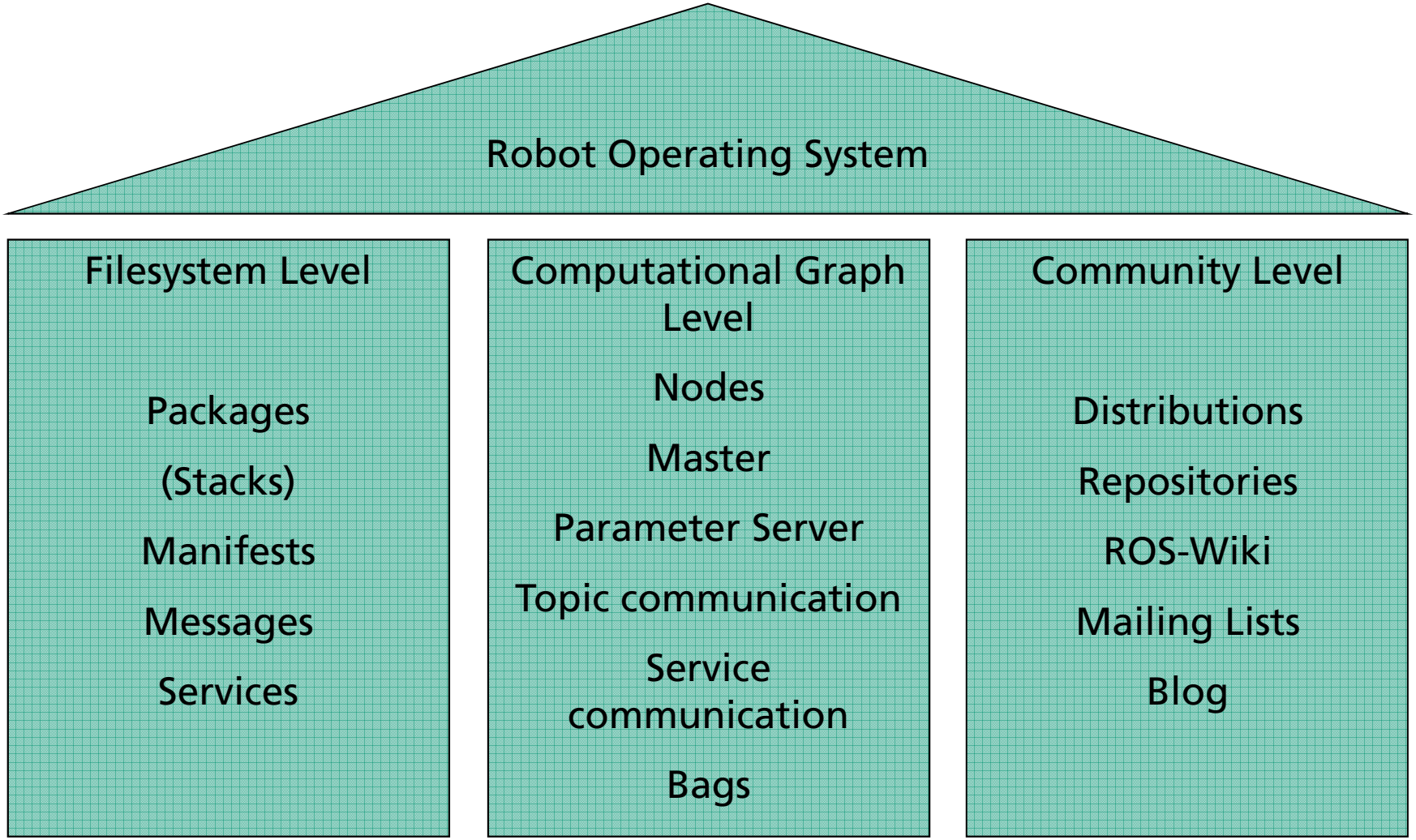


Fraunhofer IPA

Computational Graph/Plumbing



Three levels of ROS concepts



http://ros.org/wiki/ROS/Concepts

Filesystem Level

- Packages
  - Main unit for organizing software
  - Typically one functionality, e.g. localisation or path planning
  - Contains: runtime processes (nodes), libraries, datasets, configuration files, …
- Meta-packages (former stacks)
  - Collection of packages
  - Aggregate functionality, e.g. navigation stack
  - Releases and versioning
- Package- Manifests (*.xml)
  - Provide Metadata about a package/meta-package, e.g. license information and dependencies to other packages/meta-packages

http://ros.org/wiki/ROS/Concepts

Filesystem Level

- Messages types (*.msg)
  - Message descriptions, define data structures used for message communication
  - Language independent

TargetPoses.msg

```
Header header
Std_msgs/String name
Geometry_msgs/Pose2D[] poses
```

Pose2D.msg

```
Float64 x
Float64 y
Float64 theta
```

- Services types (*.srv)
  - Service descriptions, define request and response data structures used for service communication
  - Language independent

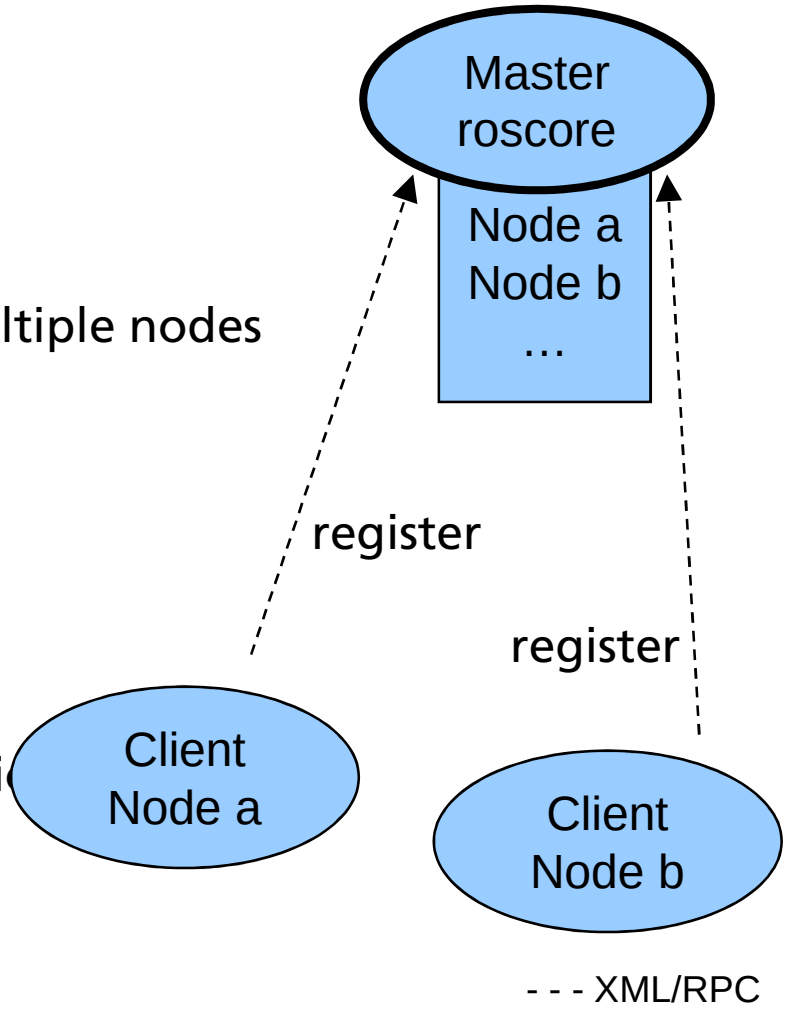
GetPose.srv

```
std_msgs/String name
--
Geometry_msgs/Pose2D pose
```

http://ros.org/wiki/msg, http://ros.org/wiki/srv

Computational Graph Level

- Nodes
  - Processes to perform computation
  - A robot control system consists of multiple nodes
  - Nodes can be inserted, changed and removed at runtime
  - Written by a ROS client library, e.g. roscpp, rospy, …
- Master
  - Coordinating processes and communi
  - Name registration and lockup
- Parameter Server
  - Central location for storing data

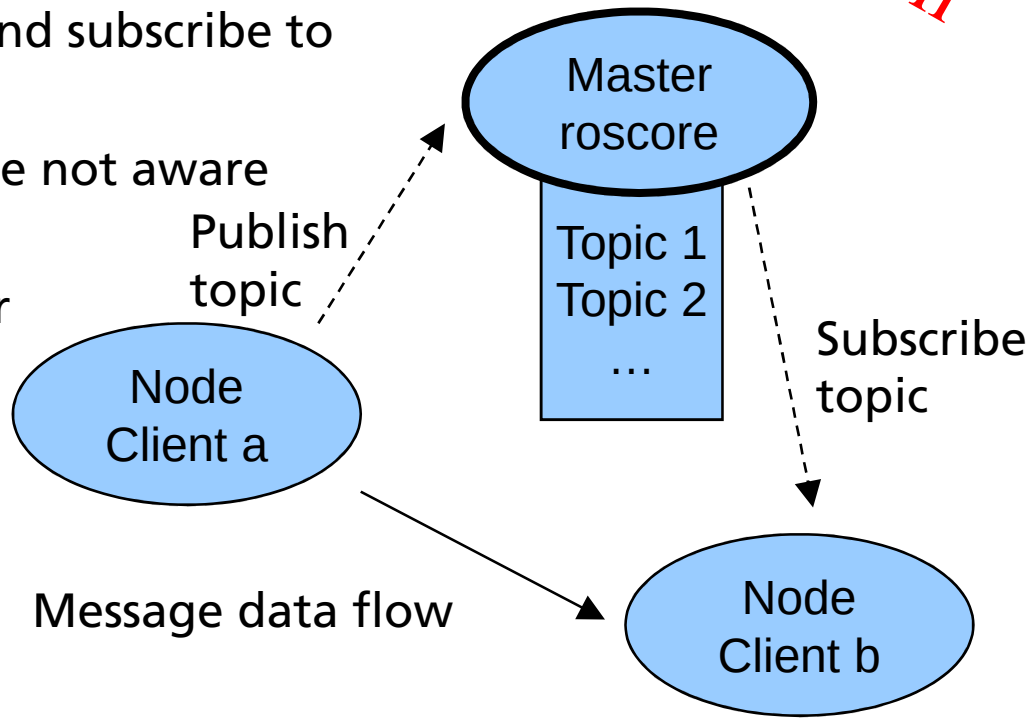


http://www.ros.org/wiki/ROS/Technical%20Overview

Communication concepts – topics

Many-to-many, one-way communication

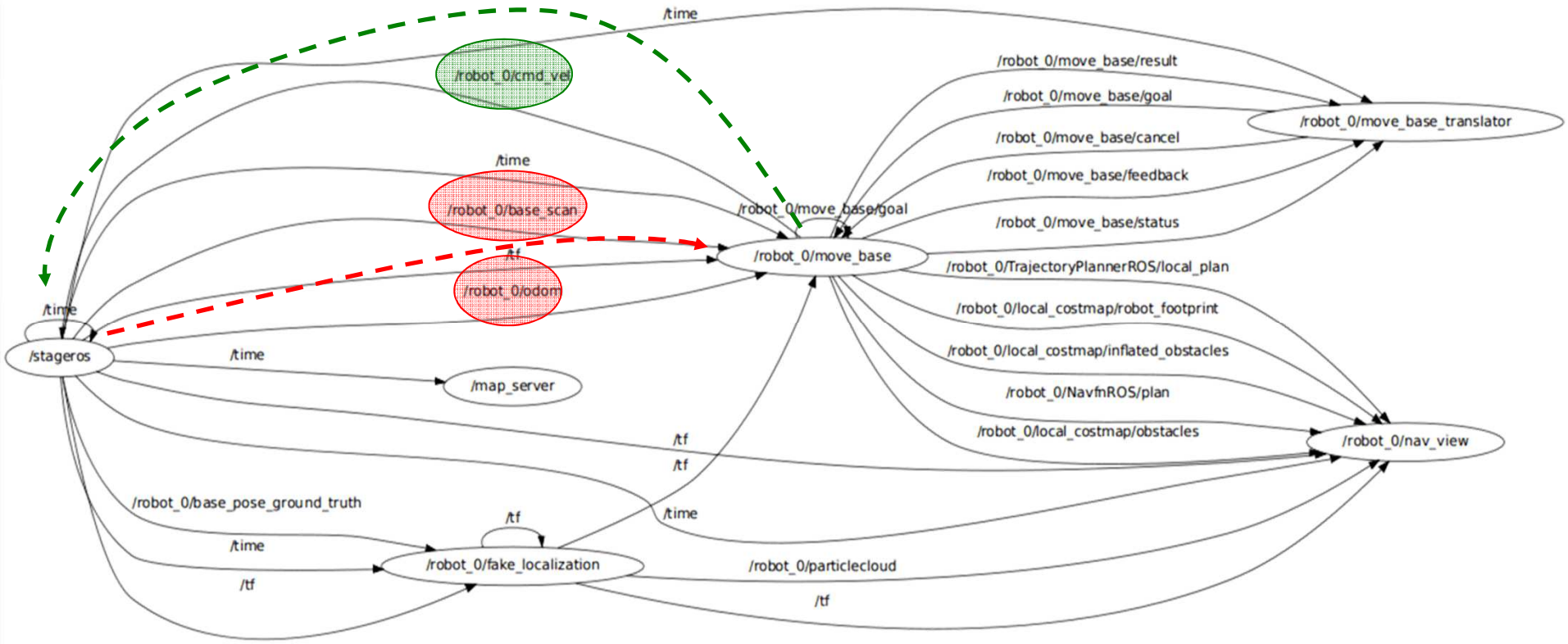
- Topics (asynchronous streaming)
- Multiple concurrent publishers and subscribers for one topic
- A single node can publish and subscribe to multiple topics
- Publisher and subscribers are not aware of each others' existence
- Decoupling between sender and receiver
- Works like a "chat room"



- - - XML/RPC

----- TCP/IP or UDP

Computation Graph Level – Example

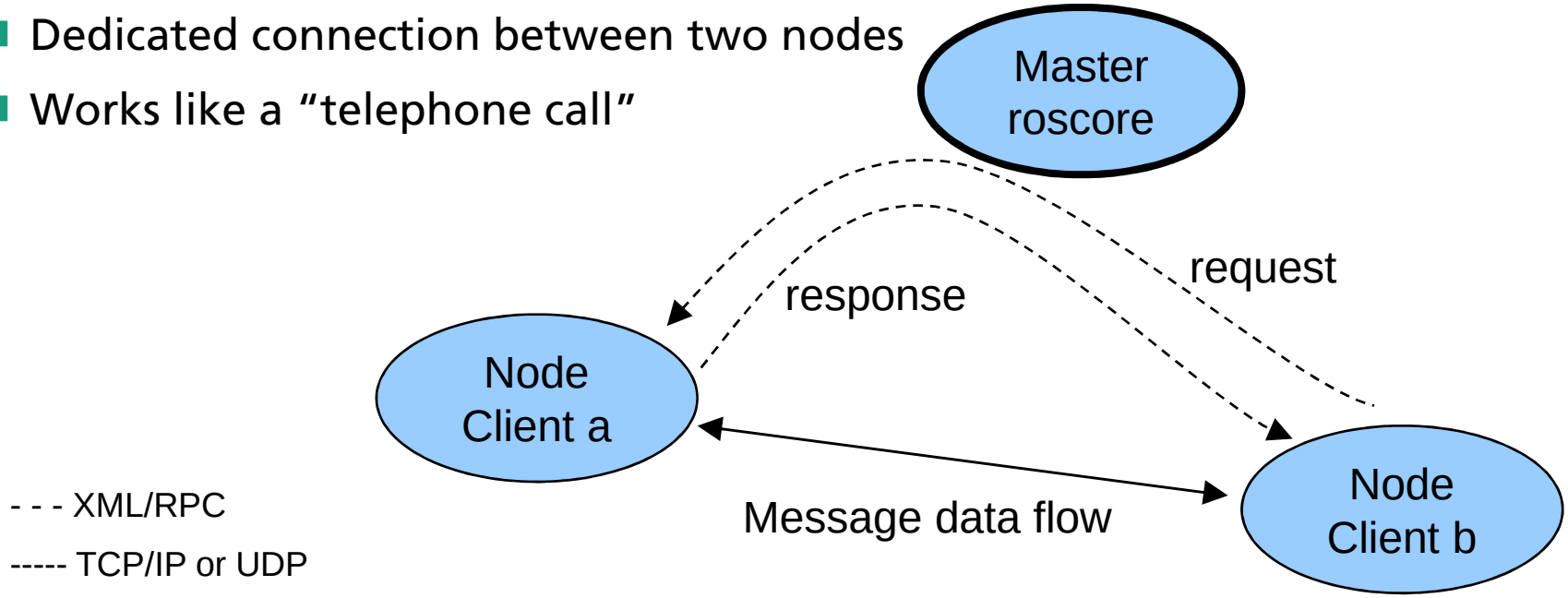


Fraunhofer IPA

Communication concepts – services

One-to-one, two-way communication

- Services (synchronous communication)
- Request and reply interaction
- Defined by a pair of message structures: request and reply
- Dedicated connection between two nodes
- Works like a “telephone call”



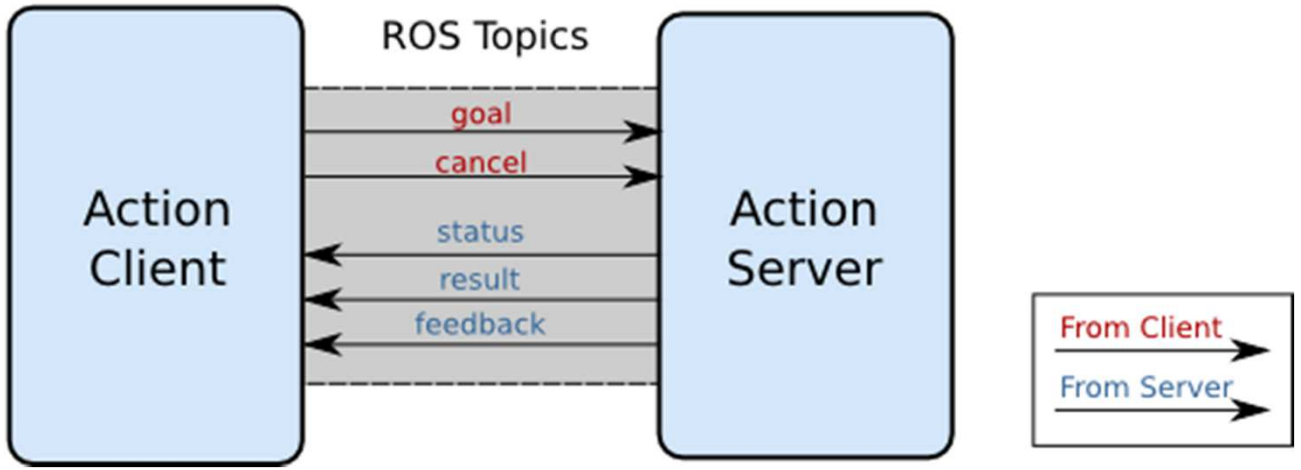
Computation Graph Level

- Actionlib
  - Goal description similar to message and service definitions
  - State-machine running on server and client

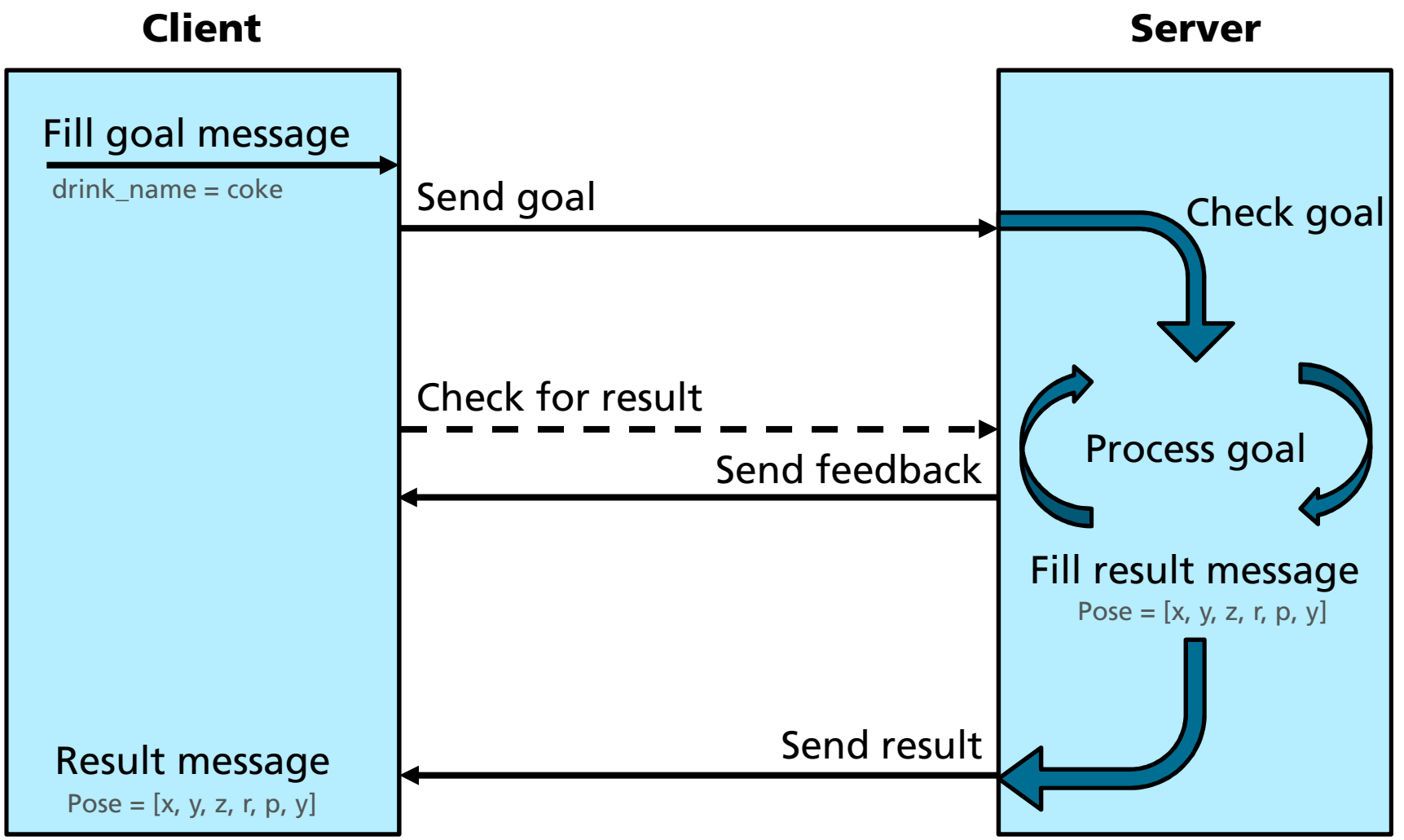
DetectBottle.action

```
#goal
std_msgs/String drink_name
---
#result
Geometry_msgs/Pose3D pose
---
#feedback
Int16 status
```

Action Interface



Actionlib workflow example: detect bottle



Fraunhofer IPA

ROS environment settings

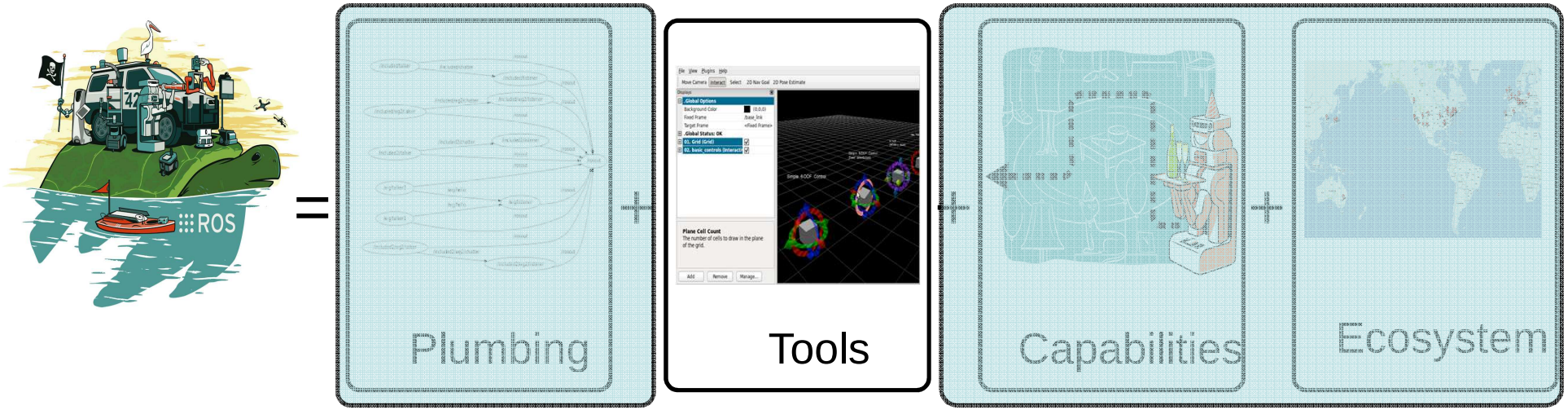
- ROS_PACKAGE_PATH
  - Search path to find packages
  - E.g.
    ROS_PACKAGE_PATH=<Path_to_your_overlays>:/opt/ros/groovy/stacks:/opt/ros/groovy/share
- ROS_MASTER_URI
  - Defines host and port where the roscore is running
  - All ROS nodes need to be able to connect to this URL
  - E.g.
    http://localhost:11311
- ROS cheat sheet
  - Lists lot of useful terminal commands
  - http://download.ros.org/downloads/ROScheatsheet.pdf

ROS deployment

- Starting a single node with rosrun
  - rosrun <package_name> <executable_name> [arguments]

- Starting multiple nodes with roslaunch
  - roslaunch <package_name> <launchfile_name> [arguments]
  - Can be grouped and hierarchically included
  - Support use of arguments and renaming of topics
  - Support uploading of parameters through yaml-files

Tools

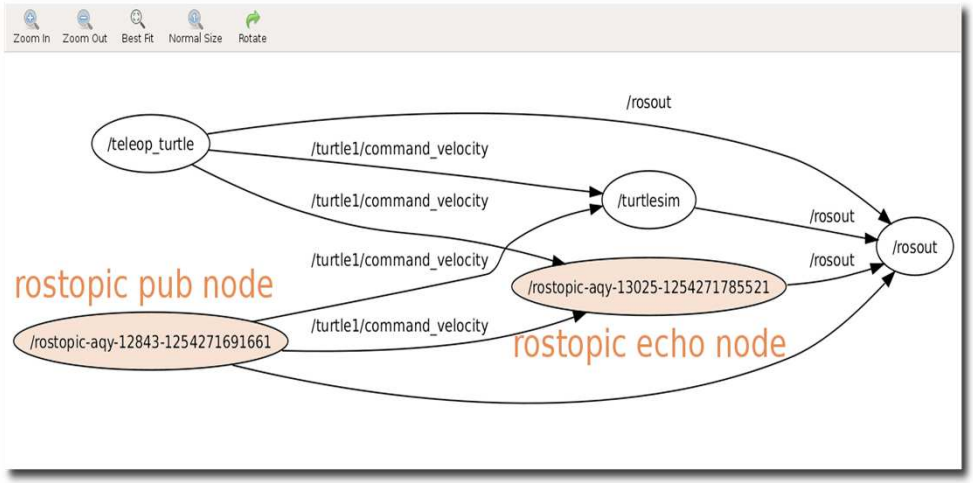


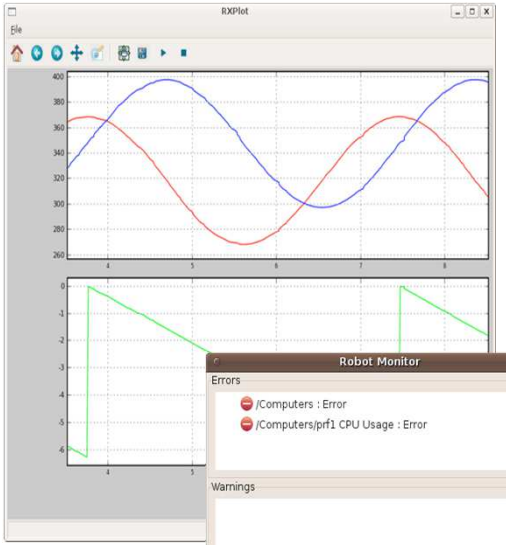
Fraunhofer IPA

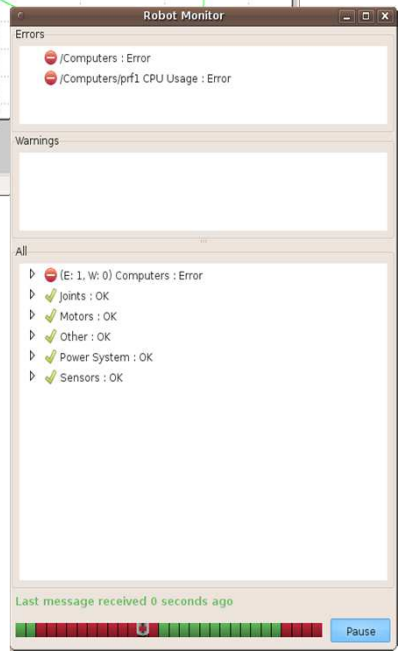
Tools

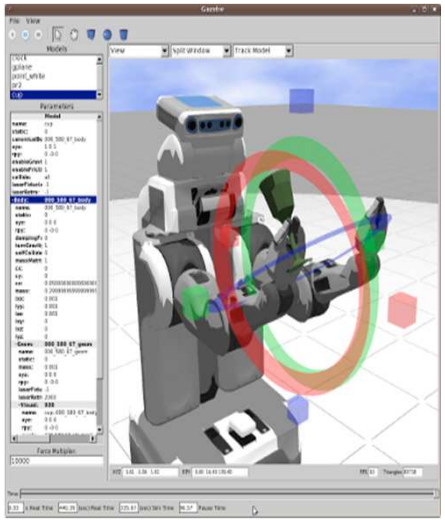
- Standard Linux tools
  - Compilers, Debuggers, Loggers, IDEs
- Multiple language support
  - C/C++, Java, Python, Lisp
- Standard libraries
  - Boost, MySQL, XML (whatever you can imagine)
- Modern GUI tools

Tools









plotting

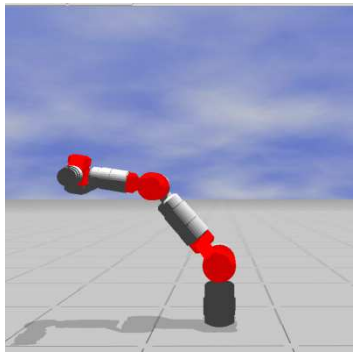
graph visualization

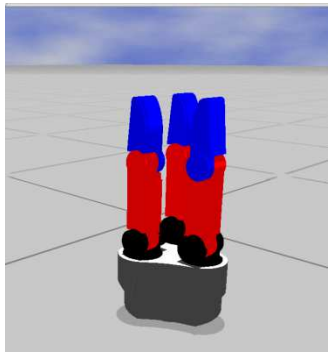
diagnostics

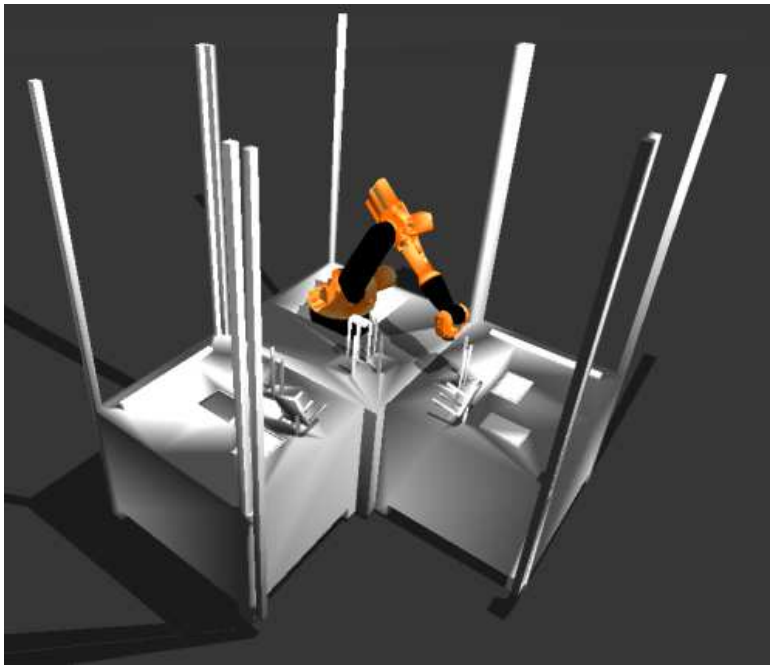
Simulation/visualization

Fraunhofer IPA

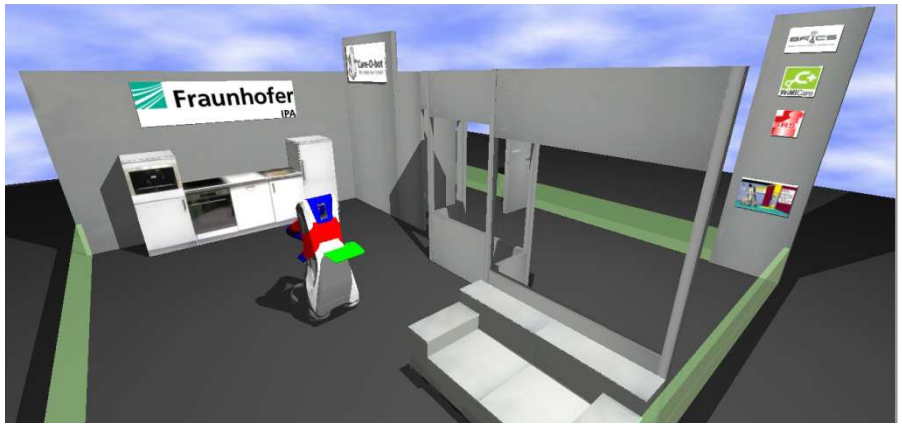
Tools: Simulation – Gazebo







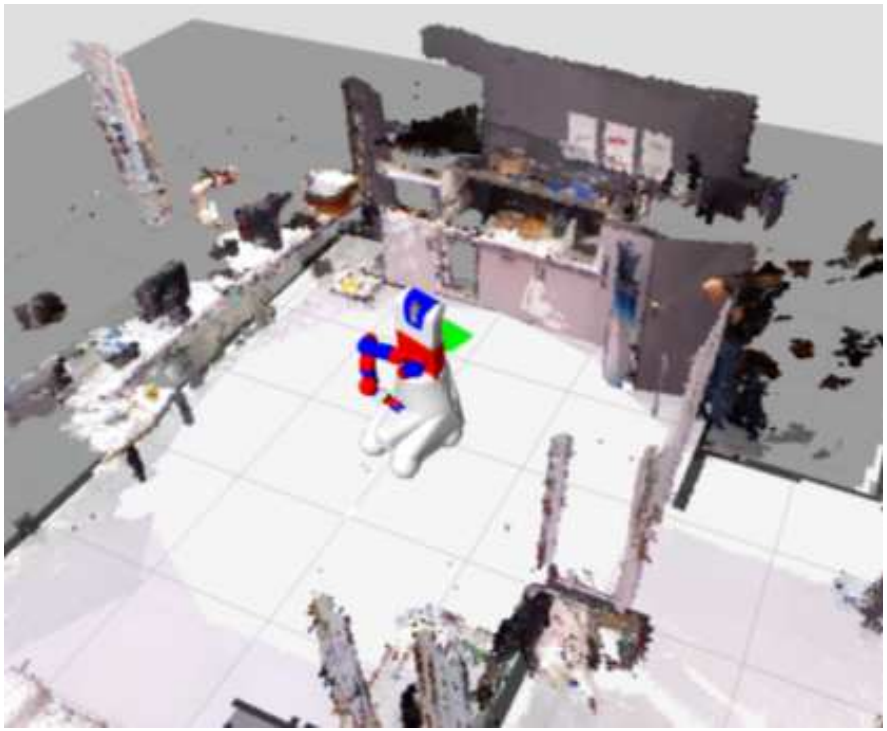
- Single simulated components
- Simulated sensors and actors, e.g.
  - Arm joints, hand
  - Cameras, laser scanners
- Model of the whole robot
- Kinematic and dynamic models of hardware components
- Environment model

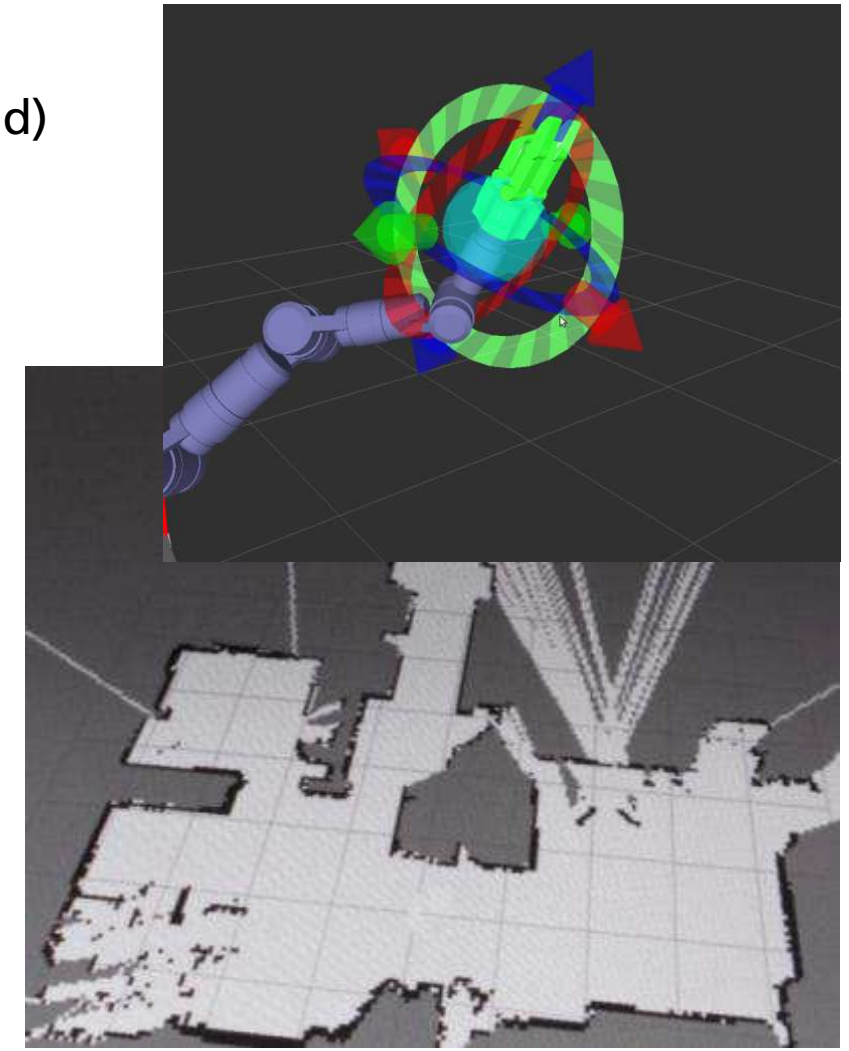


http://www.ros.org/wiki/simulator_gazebo

Tools: Visualization – RVIZ

- Robot Model
- Sensor data (laser scanner, point cloud)
- 2D and 3D maps
- (Interactive-) markers

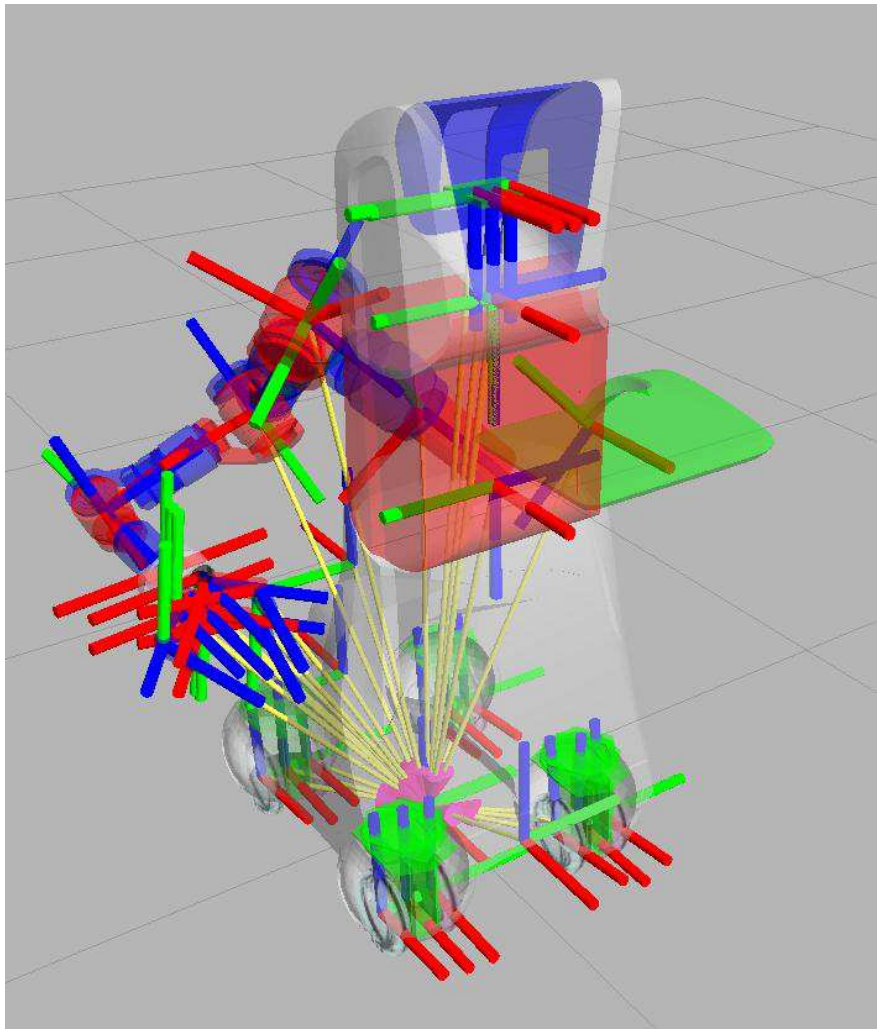




http://www.ros.org/wiki/rviz

Tools: transformations library (tf)

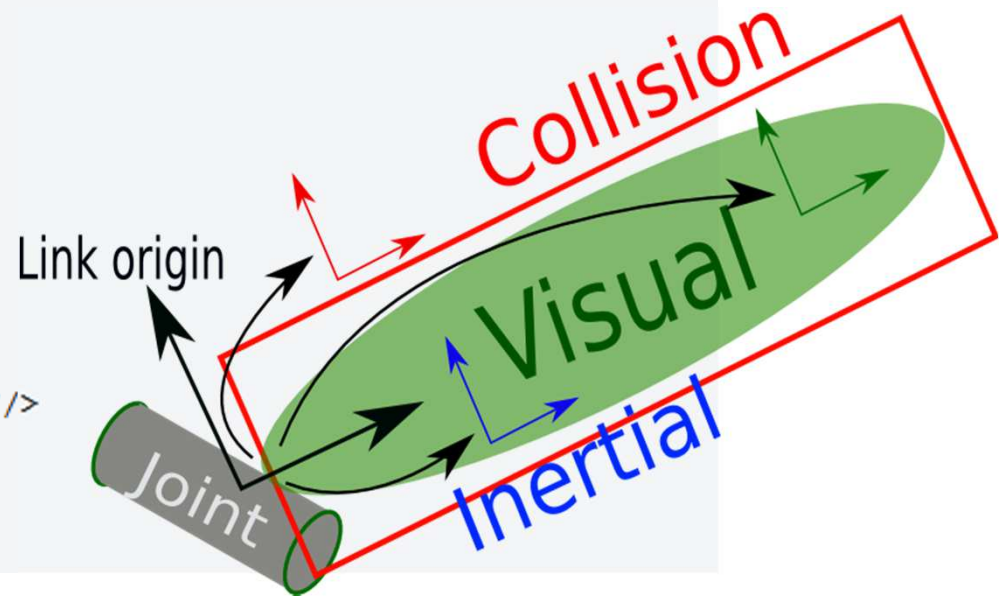
- Tree of coordinate systems
- Defined by urdf (Robot Description Language)
- Generated automatically out of /joint_states topic
- Transformations between all coordinate systems available
- API for geometic transformations and convertion (C++ and Python)



http://www.ros.org/wiki/tf

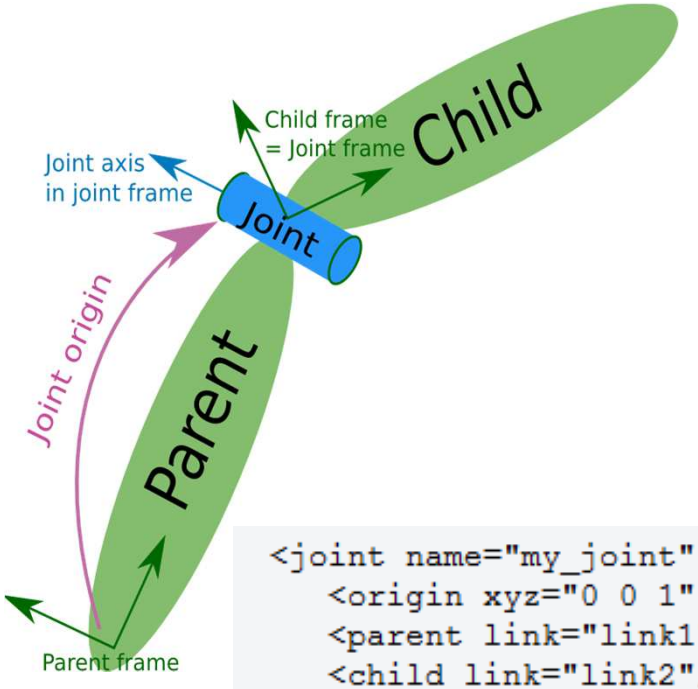
Tools URDF (link)

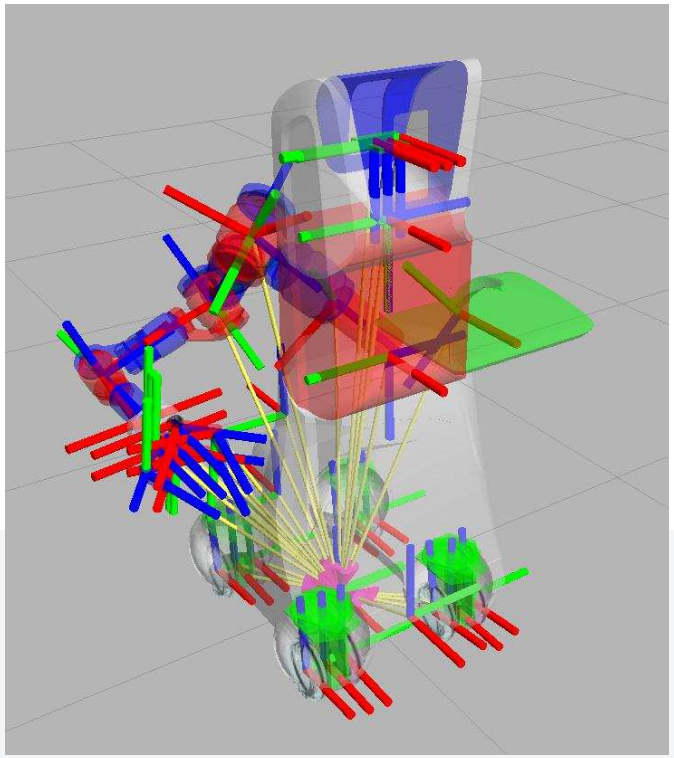
```
<link name="my_link">
  <inertial>
    <origin xyz="0 0 0.5" rpy="0 0 0"/>
    <mass value="1"/>
    <inertia ixx="100"  ixy="0"  ixz="0" iyy="100" iyz="0" izz="100" />
  </inertial>

  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <box size="1 1 1" />
    </geometry>
    <material name="Cyan">
      <color rgba="0 255 255 1.0"/>
    </material>
  </visual>

  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <cylinder radius="1" length="0.5"/>
    </geometry>
  </collision>
</link>
```



- http://www.ros.org/wiki/urdf

Tools URDF (joint)

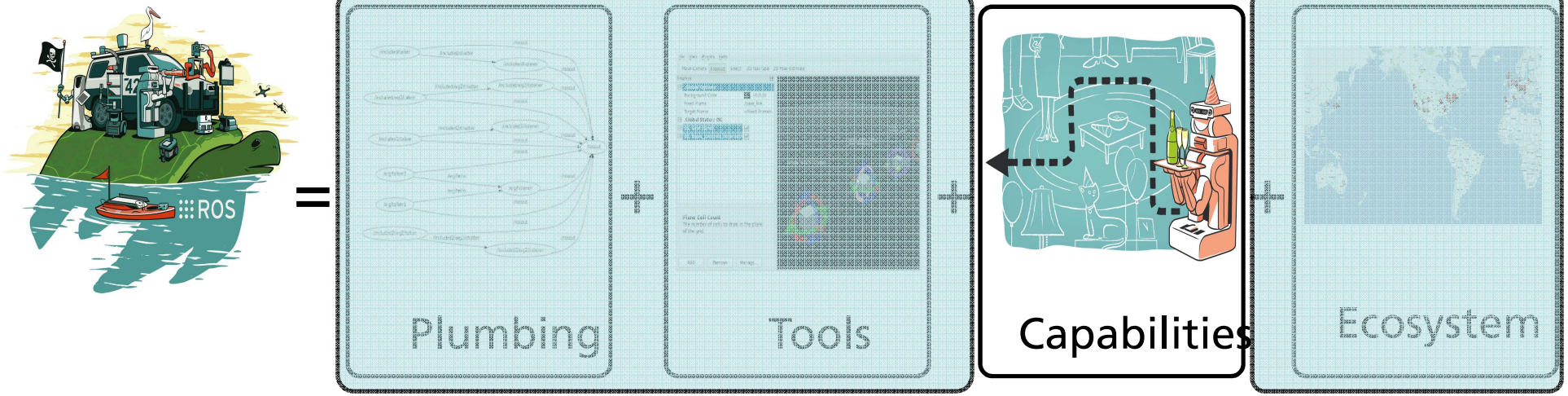




```
<joint name="my_joint" type="floating">
   <origin xyz="0 0 1" rpy="0 0 3.1416"/>
   <parent link="link1"/>
   <child link="link2"/>

   <calibration rising="0.0"/>
   <dynamics damping="0.0" friction="0.0"/>
   <limit effort="30" velocity="1.0" lower="-2.2" upper="0.7" />
   <safety_controller k_velocity="10" k_position="15" soft_lower_limit="-2.0"
soft_upper_limit="0.5" />
 </joint>
```

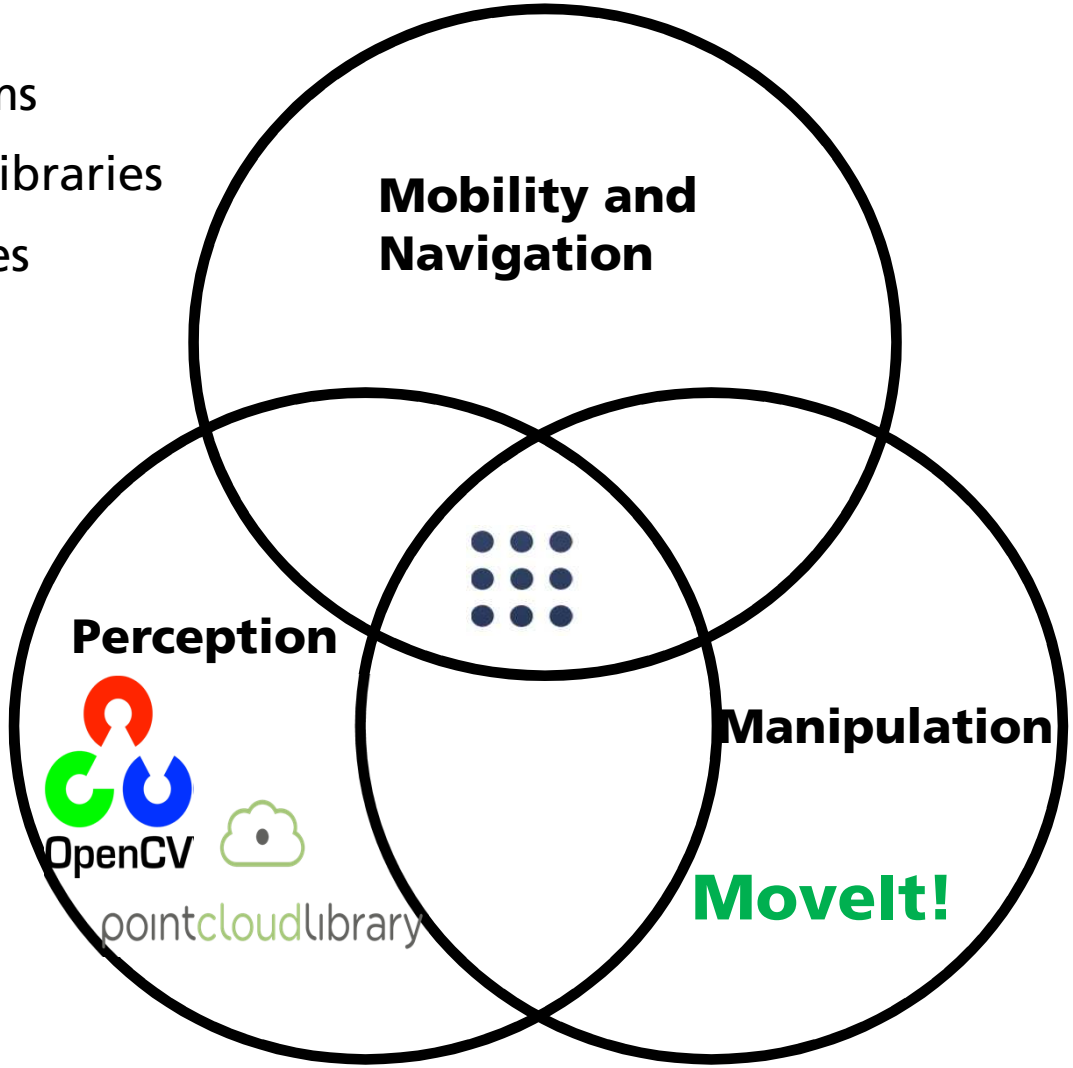
- http://www.ros.org/wiki/urdf

Capabilities

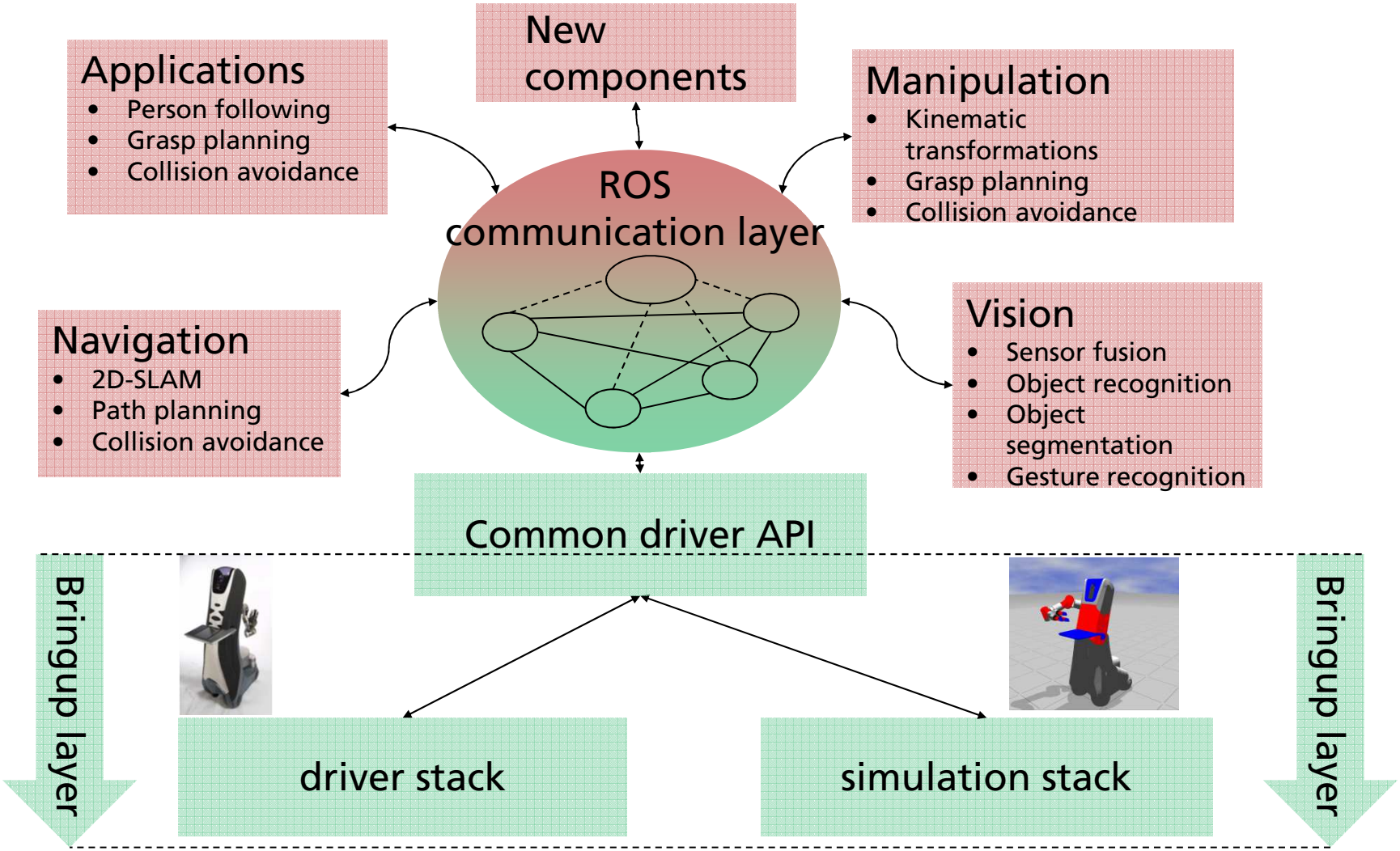


Capabilities

- State of the art algorithms
- Integration of available libraries
- Wide range of capabilities
  - Navigation
  - Perception
  - manipulation

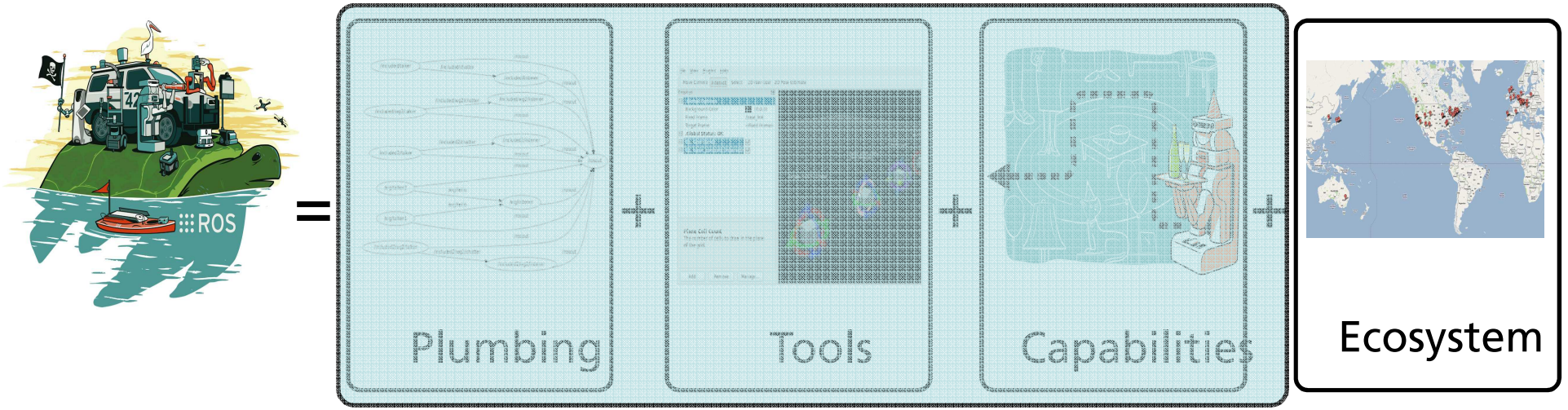


Capabilities



Fraunhofer IPA

Community/Ecosystem

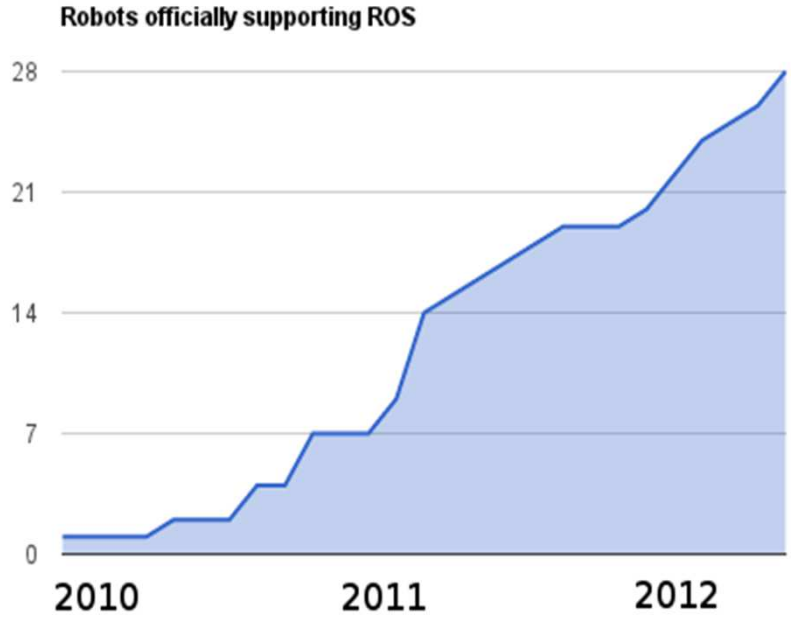


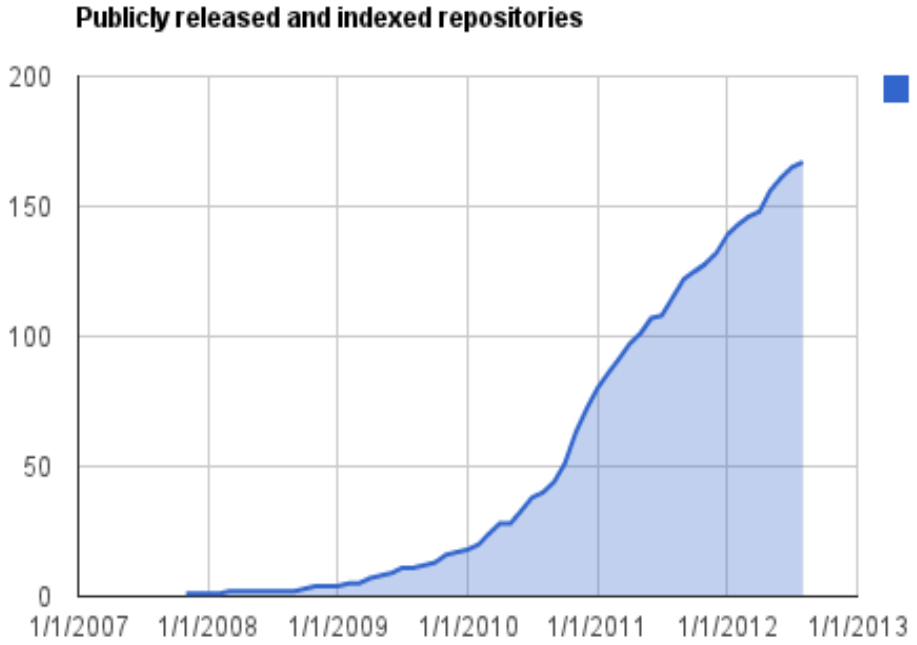
Community/Ecosystem

- Repositories
  - Federated network of code repositories
  - Different institutions can develop and release robot software components
- ROS Wiki (www.ros.org)
  - Main forum for information and documentation
  - Tutorials
- Bug Ticket System “trac” (https://code.ros.org/trac/ros) and github issues
  - Bugs can be reported and processed
- Mailing Lists
  - Review information and discussion about packages and interfaces
  - Information about new packages
- OSRF consultant network
  - http://osrfoundation.org/consultants-network/

Community/Ecosystem

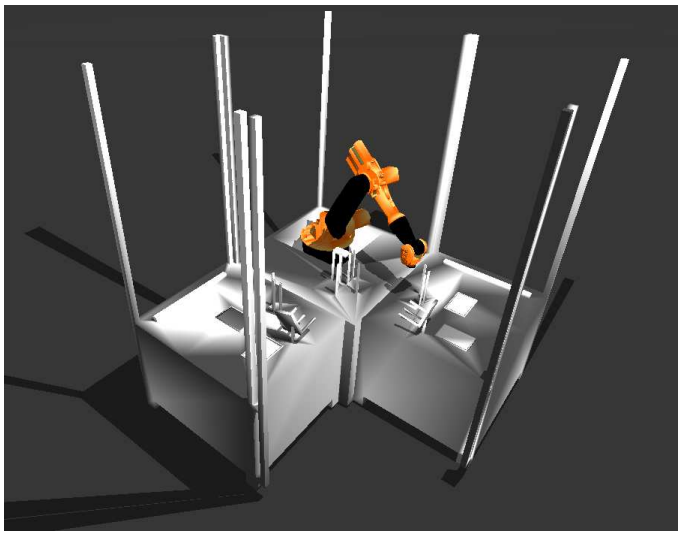
- Fast growing community
- De facto standard for service robotics





Fraunhofer IPA

Next step: from ROS to ROS Industrial



How can the power of ROS be used in the industrial domain?

ROS → ROS industrial

www.rosindustrial.org
www.ros.org/wiki/Industrial

Station 1:
Perception

Joshua Hampp

HANDS-ON SESSIONS

- Please form groups of max 4 persons
- After 1.5h switch to next station
- At each station
  - You will get an introduction to the topic
  - You will find tutorial information

Session 1: 10:00 – 11:30
Session 2: 11:30 – 13:00
Session 3: 13:45 – 15:15
Session 4: 15:15 – 16:45

Station 2:
Navigation

Alexander Bubeck

Station 3:
Manipulation

Felix Meßmer

Station 4:
Application

Florian Weißhardt

Fraunhofer IPA