



# Platform(s) and State of Simulation

S. Rockel, D. Klimentjew, J. Zhang  
{klimentjew, rockel, zhang}@informatik.uni-hamburg.de



University of Hamburg  
Faculty of Mathematics, Informatics and Natural Sciences  
Department of Informatics  
Technical Aspects of Multimodal Systems



December 1, 2011



# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

## Sensors and Simulation

Further Sensors

Gazebo



# Outline

## Platforms

### PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

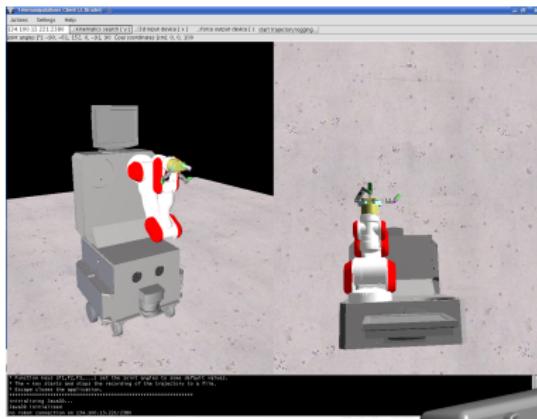
## Sensors and Simulation

Further Sensors

Gazebo



# Introduction





# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

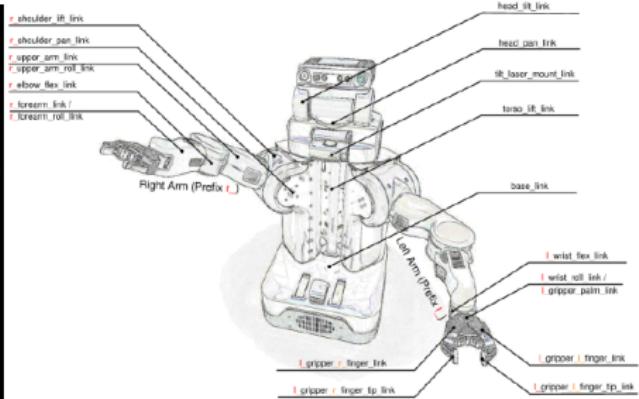
## Sensors and Simulation

Further Sensors

Gazebo



## PR2





# PR2 - Users



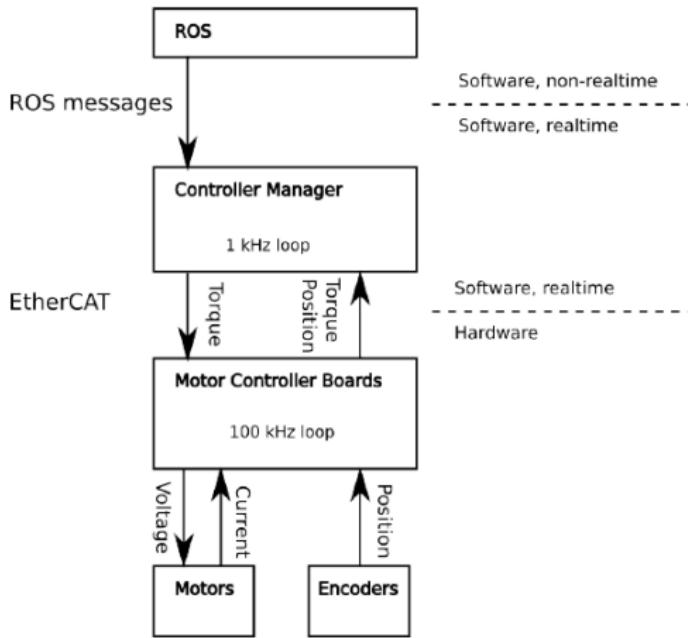


## PR2 - Hardware Specification

- ▶ 2× computers with 24 Gb RAM and quad-core Nehalem processors
- ▶ 1.3 kWh Lion Battery Pack
- ▶ 2 hrs Approximate Runtime
- ▶ Coordinate system (for all links) positive z-axis up, positive x-axis forward, and positive y-axis robot-left when PR2 in the home pose

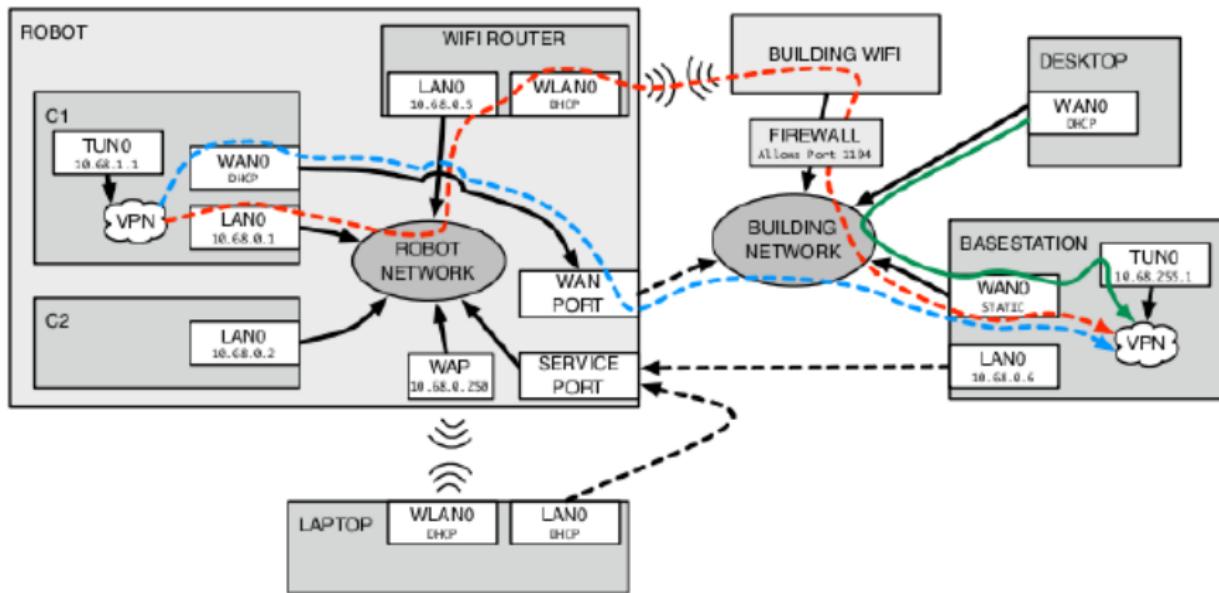


# The PR2 motion control layout





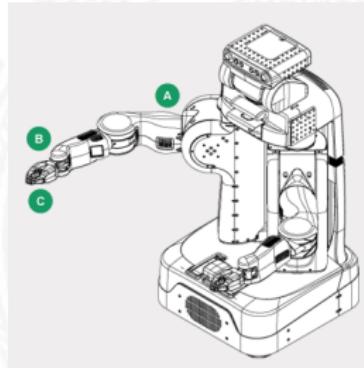
# Network explanation





## PR2 - Hardware Specification

- ▶ Arm DOFs: arm 4 (A), wrist 3 (B), gripper 1 (C)
- ▶ Link Lengths: upper arm 400 mm, forearm 321 mm, wrist to gripper surface 120 - 200 mm
- ▶ Range of motion: shoulder pan/tilt  $170^0/115^0$ , upper arm roll  $270^0$ , elbow flex  $140^0$ , forearm roll continuous, wrist pitch/roll  $130^0$ /continuous, gripper 90 mm max
- ▶ Force output: 4 DOF passive counterbalance, arm payload 1.8 Kg, wrist torque 4 Nm, grip force 80 N





## PR2 - Intrinsic sensors

- ▶ Microstrain 3DM-GX2 IMU (above the shoulders)
- ▶ Three-Axis Accelerometer (gripper)
- ▶ Calibration LED (gripper)



## PR2 - Extrinsic sensors - Head

- ▶ Microsoft Kinect (color/depth image/point cloud [ $640 \times 480 @ 30\text{ fps}$ ])
- ▶ Global shutter color gigabit ethernet camera (Prosilica GC2450C, 5 MP, [ $2448 \times 2050 @ 15\text{ fps}$ ])
- ▶ Wide stereo camera system (Aptina MT9V032C12STC, 100 Mb color ethernet, [ $752 \times 480 @ 15\text{ fps}$ ])
- ▶ Narrow stereo system (Aptina MT9V032C12STM, 100 Mb monochrome ethernet, [ $752 \times 480 @ 15\text{ fps}$ ])
- ▶ LED textured light projector (triggered with narrow-angle stereo camera)



## PR2 - Extrinsic sensors - II

- ▶ Tilting laserscanner (Hokuyo UTM-30LX,  $135^{\circ}$ ( $+90^{\circ}$  to  $-45^{\circ}$ ), above the shoulders)
- ▶ Laserscanner (Hokuyo UTM-30LX, base)
- ▶ Global shutter gigabit ethernet camera ( $2 \times$ , forearm)
- ▶ Fingertip pressure sensor arrays (gripper)
- ▶ Speaker





# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

## Sensors and Simulation

Further Sensors

Gazebo



# ROS Introduction

## ROS

- ▶ Meta Operating System for robotics
- ▶ Obtain, build, write and run code across multiple computers and robots
- ▶ Open Source
- ▶ BSD Licensed (very liberal<sup>1</sup>)
- ▶ Willow Garage
- ▶ Community

---

<sup>1</sup>[http://en.wikipedia.org/wiki/BSD\\_licenses](http://en.wikipedia.org/wiki/BSD_licenses)



## Introduction (cont.)

## Robots Using ROS > 50





# ROS Basics

- ▶ Supported Platforms
  - ▶ Linux, Mac OS, partial support for Windows
- ▶ Languages
  - ▶ C/C++, Python, Octave, Lisp, Java



# ROS Basics

- ▶ Supported Platforms
  - ▶ Linux, Mac OS, partial support for Windows
- ▶ Languages
  - ▶ C/C++, Python, Octave, Lisp, Java



# What does ROS cover?

## ROS

- ▶ Simulation
- ▶ Task execution
- ▶ Mobile manipulation
- ▶ Navigation
- ▶ Visualization
- ▶ Client libraries
- ▶ Message passing



# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

## Sensors and Simulation

Further Sensors

Gazebo



# ROS Nodes

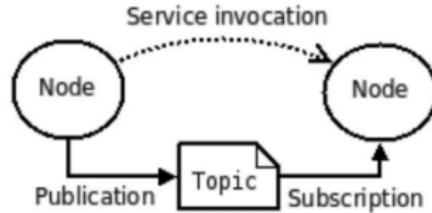
- ▶ Master (rosmaster)
  - ▶ provides naming and registration services
  - ▶ tracks topics and services
  - ▶ enables localization of nodes (nodes talk peer-to-peer)
  - ▶ XML-RPC-based API
- ▶ Generally: they are uniquely named



# ROS Communication

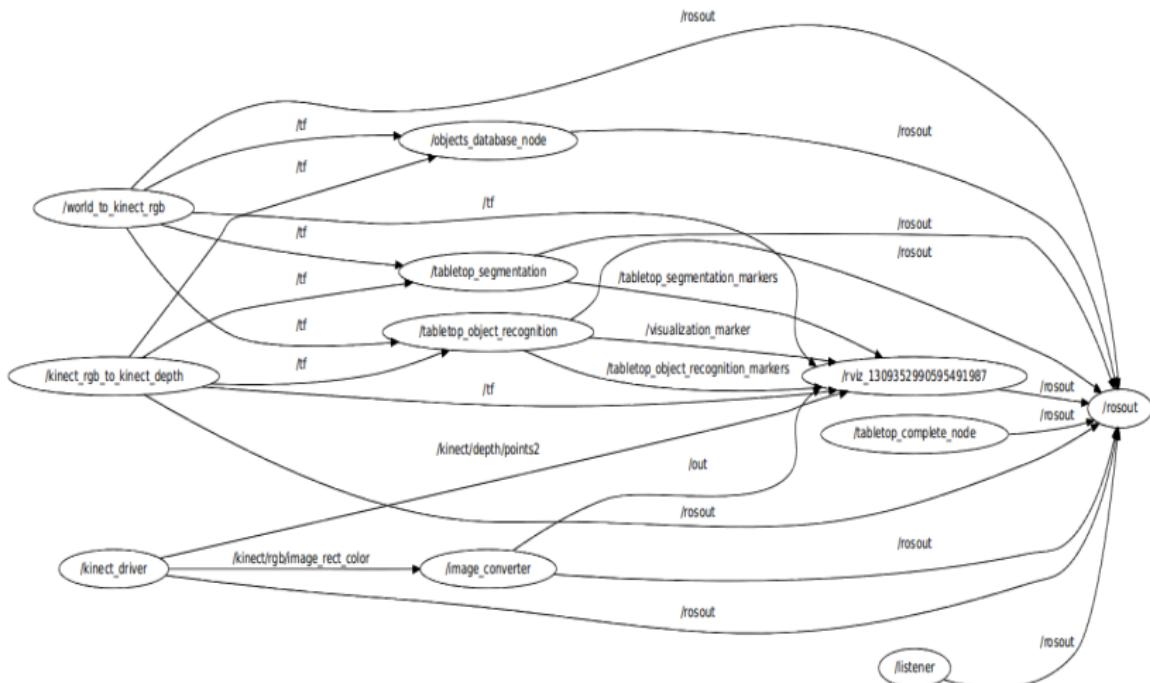
- ▶ Publisher sends message to subscribers
  - ▶ Usually TCP/IP transport
  - ▶ XML-RPC is only used to negotiate transport (no messages via XML-RPC)

## Service Invocation





# ROS Communication (cont.)





# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

## Sensors and Simulation

Further Sensors

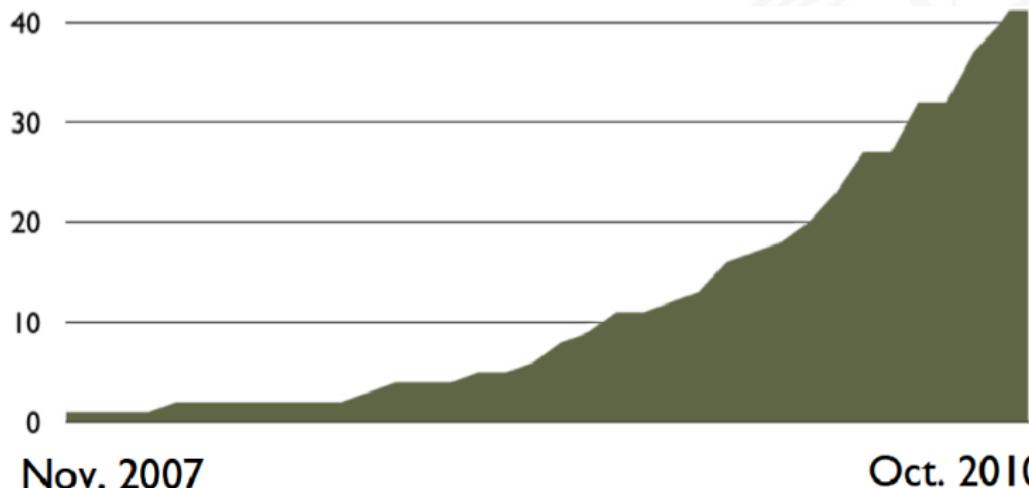
Gazebo



# ROS Repositories

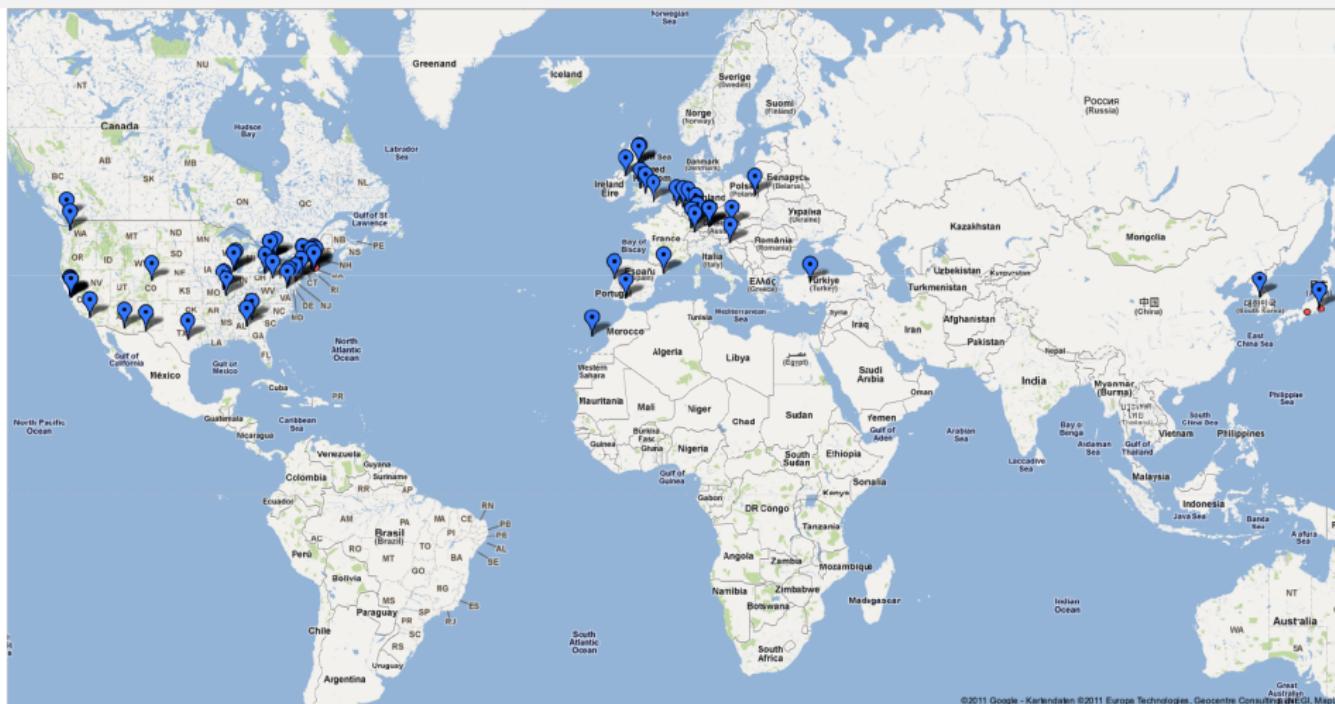
Repositories world-wide

- ▶ Collection of packages and stacks, hosted online
- ▶ many repositories (>50): Stanford, CMU, TUM ..





# ROS Repositories (cont.)

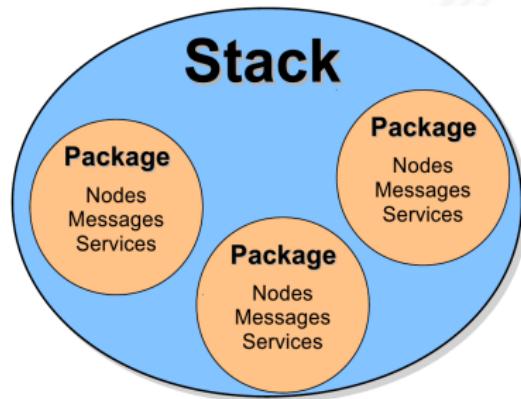


©2011 Google - Kartendaten ©2011 Europe Technologies, Geocentre Consulting/G.I. Map



# ROS Stacks

- ▶ Collect similar packages that work together to achieve e.g.:
  - ▶ 2D Navigation
  - ▶ Manipulation
  - ▶ SLAM

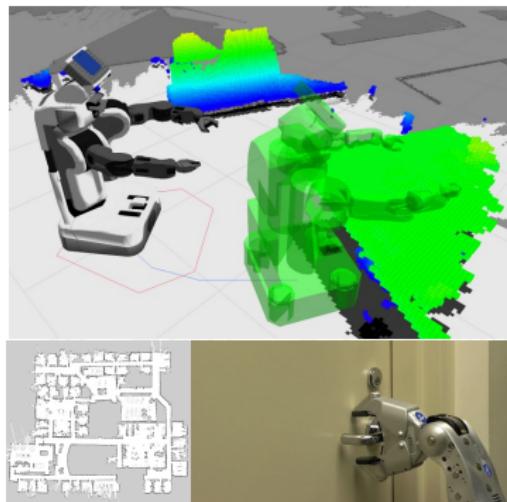




# ROS Stacks Overview<sup>2</sup>

Currently > 400 Stacks available

- ▶ (2D/3D) Navigation
- ▶ PR2 arm navigation
- ▶ PR2 opening doors
- ▶ Exploration
- ▶ GUI for PR2 robot
- ▶ PR2 object manipulation
- ▶ PR2 simulator



<sup>2</sup><http://www.ros.org/browse>



# ROS based Navigation

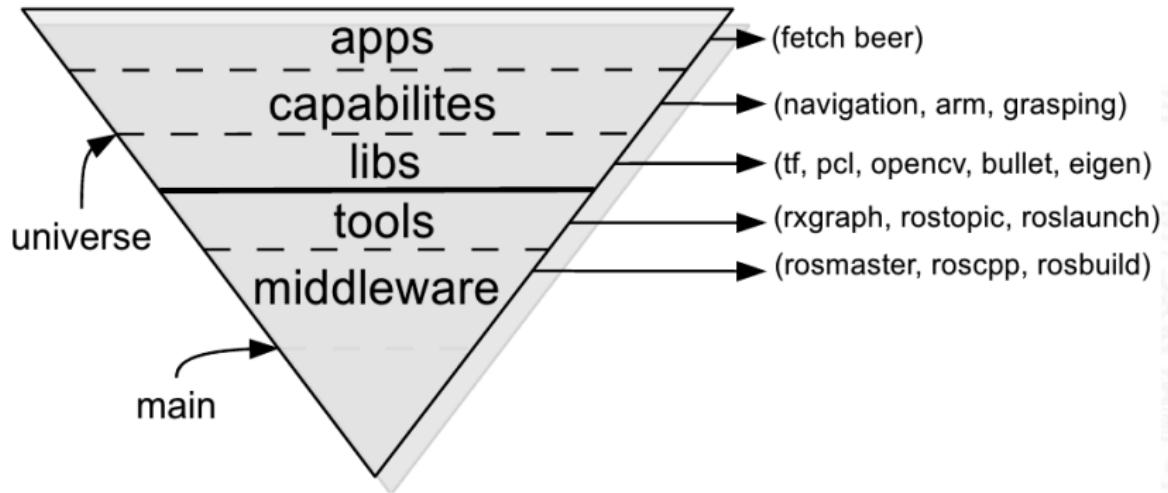
## Includes

- ▶ Path planning, Obstacle avoidance, Automatic map making



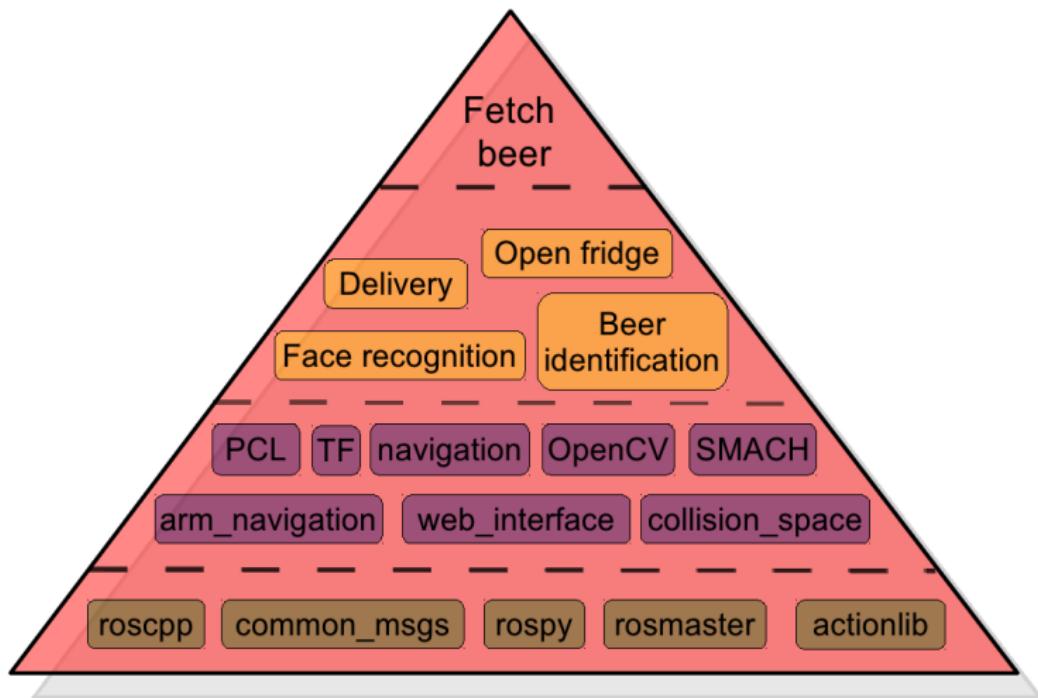


# Example Application





## Example Application (cont.)





# ROS Strengths for RACE

- ▶ Visualization
- ▶ Object recognition
- ▶ Navigation
- ▶ Manipulation/Grasping
- ▶ Plugging in Sensors
  - ▶ already integrated
  - ▶ RACE specific



# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

## Sensors and Simulation

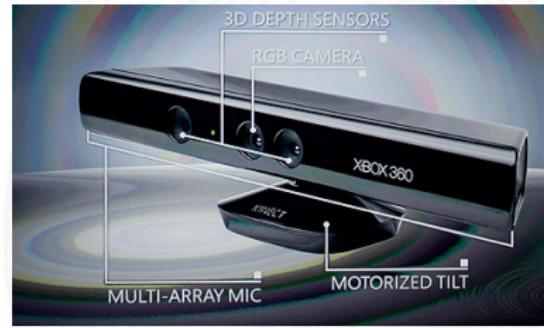
Further Sensors

Gazebo



# Kinect

- ▶ Motion sensing input device by Microsoft for the Xbox 360 video game console
- ▶ Range camera technology by Israeli developer PrimeSense
- ▶ 3D scene information from a continuously-projected infrared structured light





## Kinect operations - I

- ▶ The IR camera and the IR projector form a stereo pair with a baseline of approximately 7.5 cm
- ▶ The IR projector sends out a fixed pattern of light and dark speckles
- ▶ The pattern is generated from a set of diffraction gratings, with special care to lessening the effect of zero-order propagation of a center bright dot



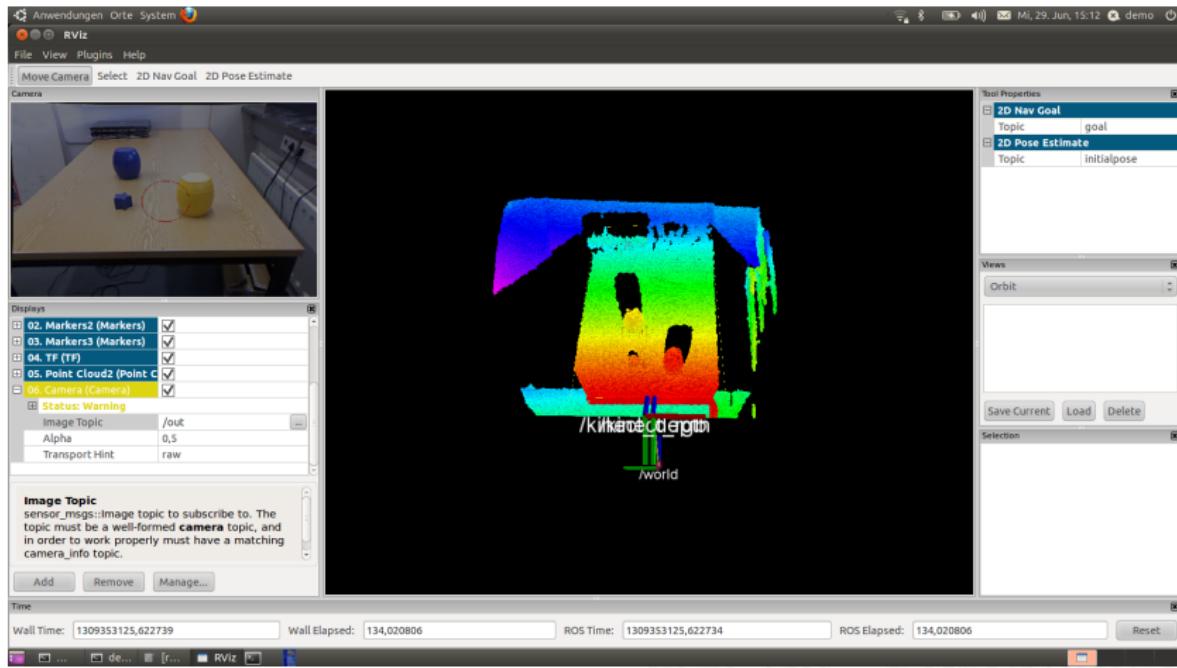


## Kinect - technical data

- ▶ Resolution of  $(640 \times 480)$  @ 30 Hz (color) and  $(320 \times 240)$  @ 30 Hz (depth)
- ▶ Angular field of view of  $57^0$  horizontally and  $43^0$  vertically
- ▶ Range of approximately 0.7 - 6 m (practical 0.7 - 3.5 m)
- ▶ Physical tilt range ( $-31^0$  to  $+31^0$ )
- ▶ Voice microphone and array supporting single speaker voice recognition (16-bit audio @ 16 kHz)
- ▶ OpenNI and Freenect drivers



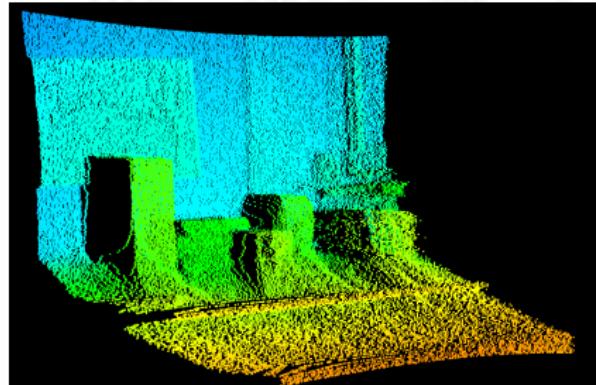
# Kinect - first results





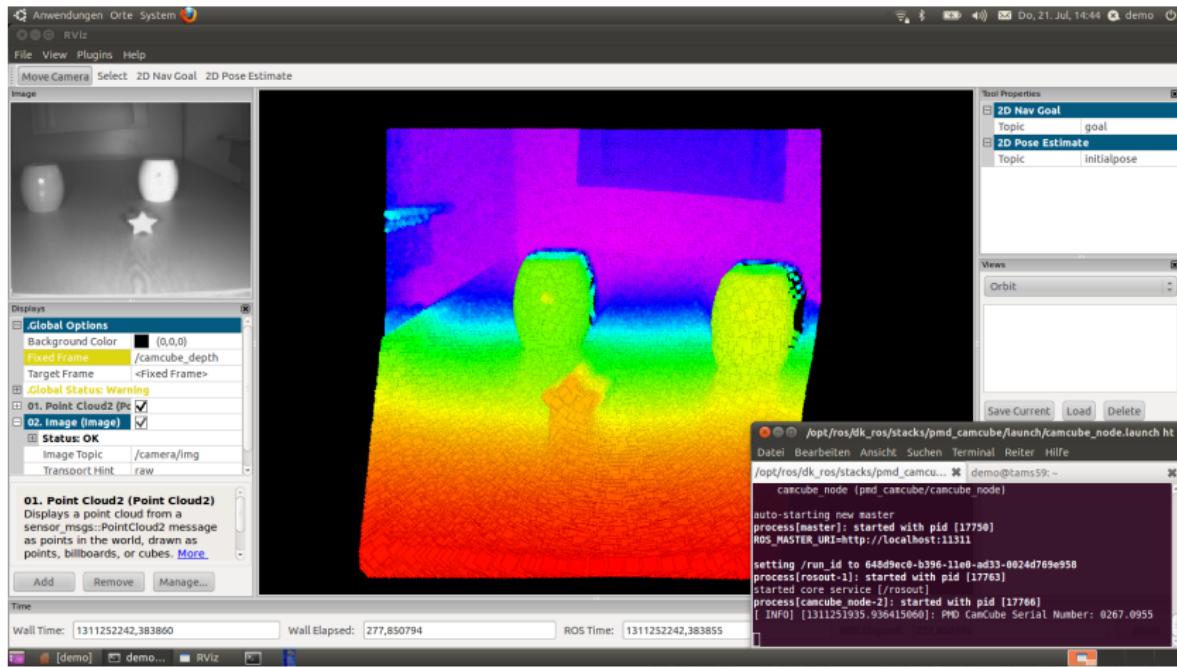
## Kinect - summary

- ▶ Fast sensor with large resolution
- ▶ Stable, repeatable results
- ▶ Ideal for surfaces, inaccuracy on edges (remedy through fusion with the tof-camera ( $200 \times 200 @ 40$  fps) or other sensors)





# PMD Camcube in ROS





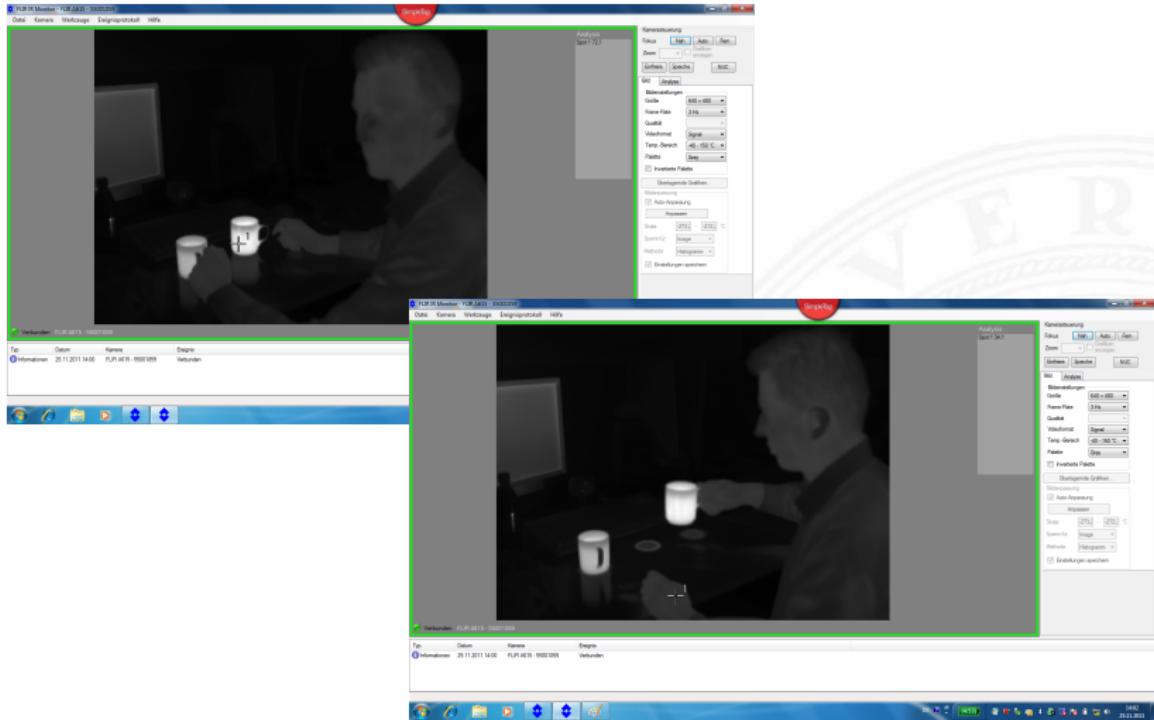
## FLIR A615

- ▶ FOV  $25^0 \times 19^0$
- ▶ IR resolution  $[640 \times 480]@50$  Hz
- ▶ Accuracy  $\pm 2^0 C$  or  $\pm 2\%$  of reading
- ▶ Image and control over Gigabit Ethernet





# FLIR A615





# Outline

## Platforms

PR2 Introduction

PR2 Overview

## ROS

ROS Introduction

ROS Concepts

ROS Repositories

## Sensors and Simulation

Further Sensors

Gazebo



## Gazebo

- ▶ Developed to be fully compatible with the Player<sup>3</sup> device server
- ▶ Full integrated in ROS (version 1.0)
- ▶ 3D dynamic multi-robot simulation
- ▶ Open source
- ▶ ODE - Open Dynamics Engine (Bullet)

---

<sup>3</sup>The Player/Stage Projekt, <http://playerstage.sourceforge.net/>

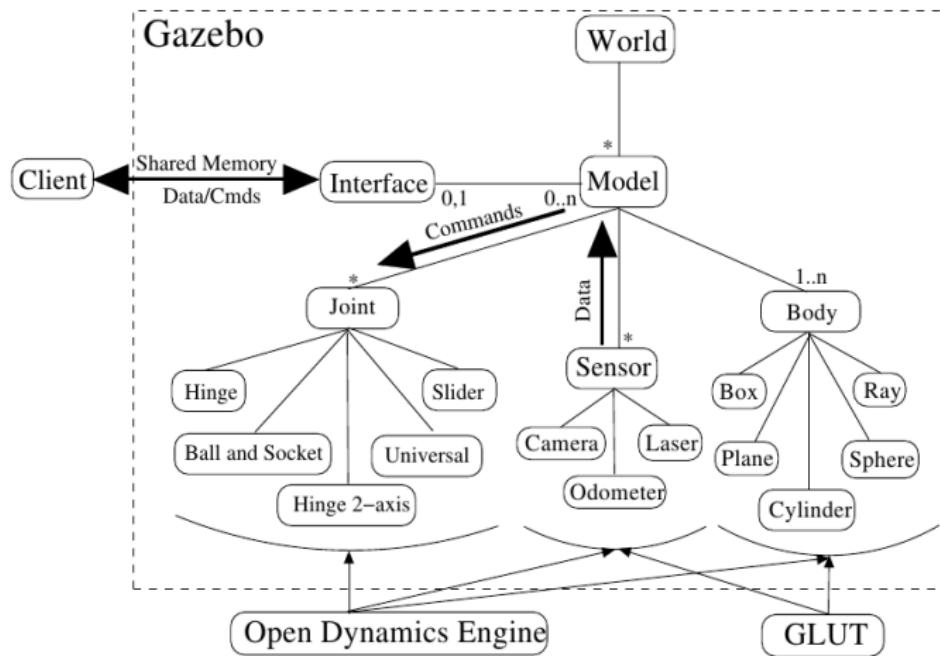


# Gazebo

- ▶ The World represents the set of all models and environmental factors such as gravity and lighting
- ▶ Accuracy in terms of robot sensors and actuators
- ▶ API for easily create new robots, actuators, sensors, and arbitrary objects
- ▶ Possibilities to create complex worlds
- ▶ Simulation of remote environments
- ▶ GUI for data visualization
- ▶ Visualization with help of OpenGL and GLUT

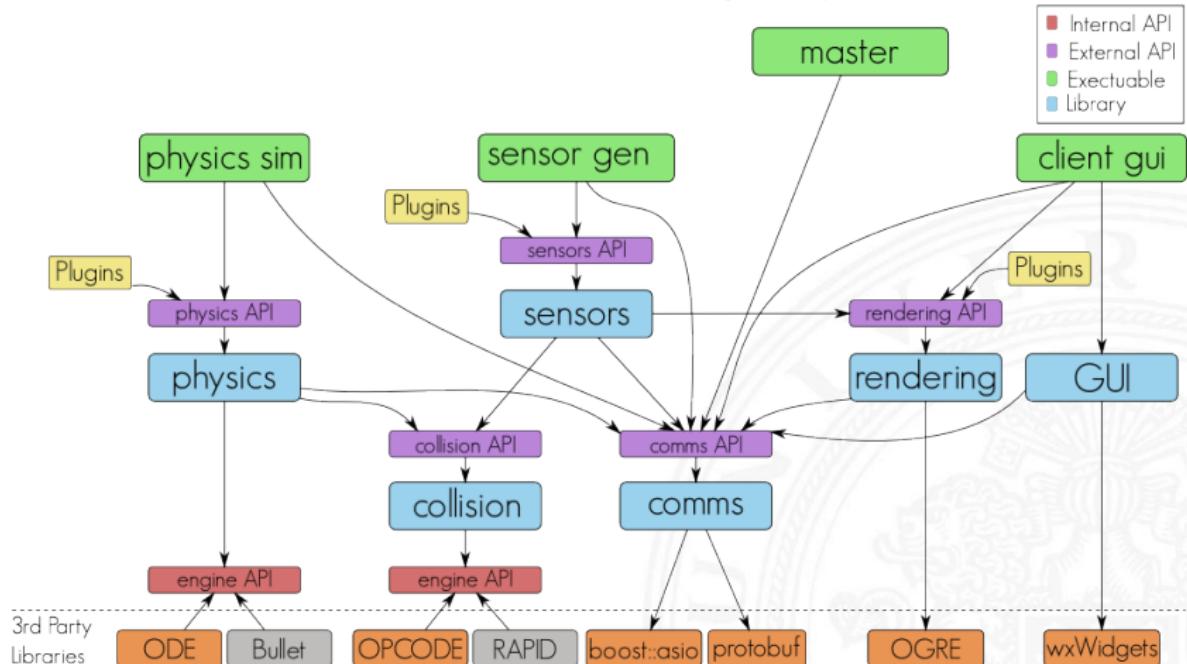


# General Structure of Gazebo components



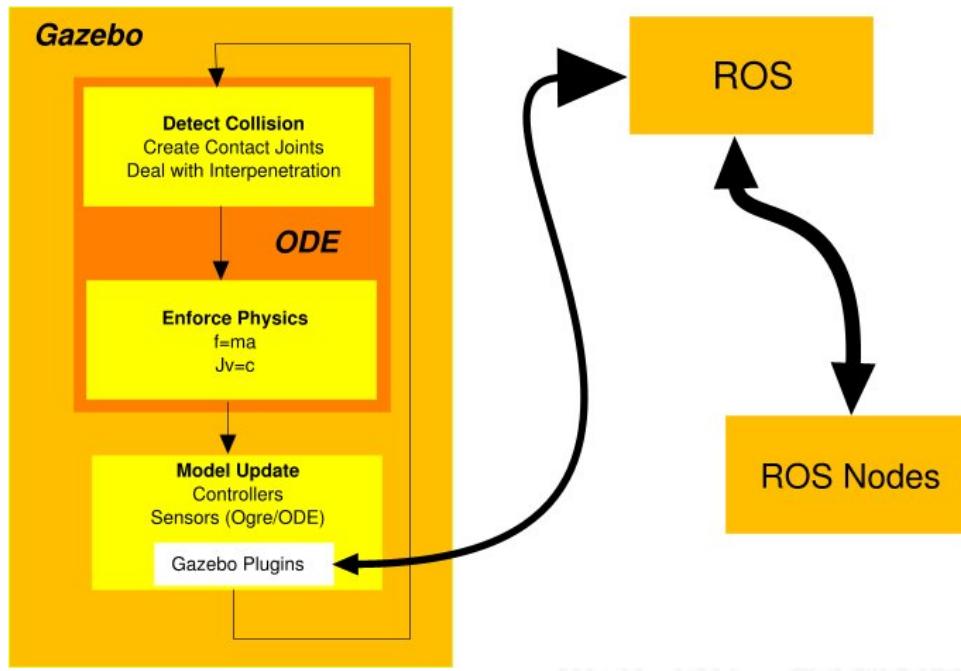


## Gazebo Dependency Graph



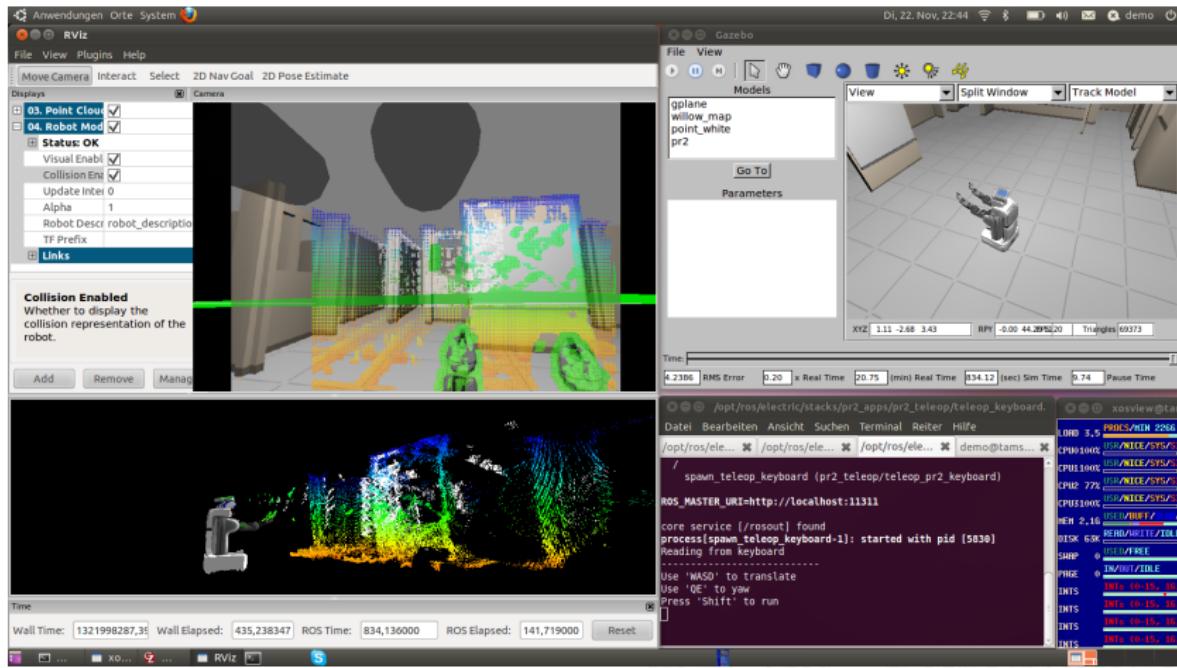


# ROS Simulation Interface





# ROS, Stage, PR2





# Thank You!

## Any questions?





# Further Reading

