

AI & Robotics

Introduction to ROS



Goals



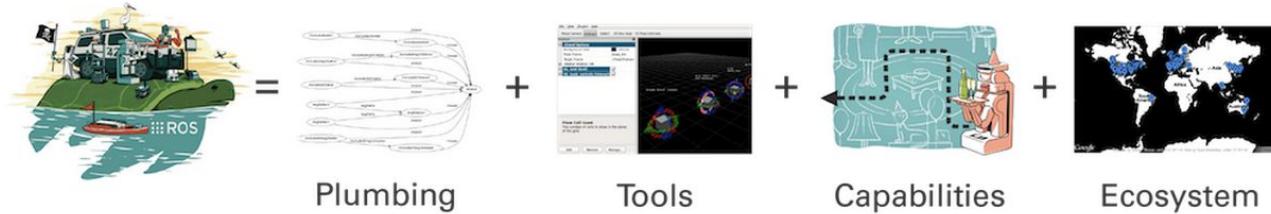
The junior-colleague

- can describe ROS in own words
- can describe flexible robotics in own words
- can explain in own words the benefits of using ROS
- can explain "structured communications layer" in context of ROS
- can explain the salient characteristics of and its consequence to ROS
- can describe in own words "distributions" in context of ROS
- can describe in own words different usages of ROS
- can explain the features of ROS
- can explain in own words distributed systems in the context of ROS
- can explain the differences between a framework and a library
- can explain interaction of nodes in context of ROS
- can describe the downsides of ROS
- can describe the difference between ROS(1) and ROS2 using the salient characteristics of ROS(1) and the new use cases.

Robot Operating System



- Open source software
- Robotics framework
- “Flexible Robotics”
- Active Community



Without ROS

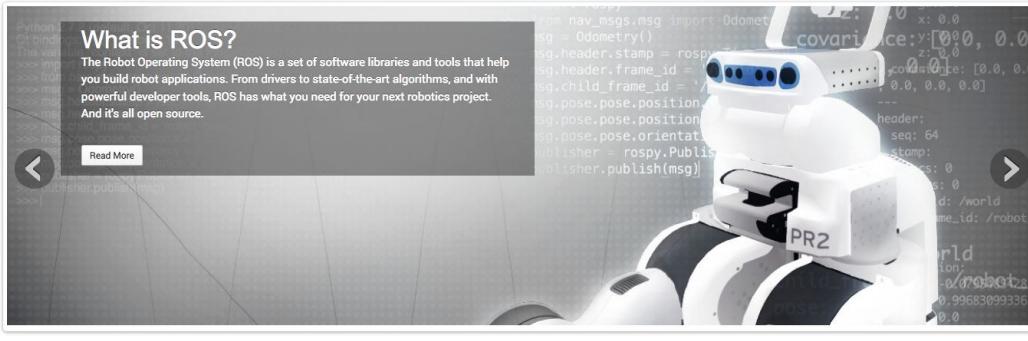


- Build device drivers
- Build communication system
- Write all algorithms for
 - perception,
 - navigation
 - motion planning
 - ...
- Create log and error handling
- ...

Website



About Why ROS? Getting Started Get Involved Blog



ROS Kinetic Kame
Kinetic Kame is the 10th official ROS release. It is supported on Ubuntu Wily and Xenial. Get Kinetic Kame now!

[Download](#)

ROS Spotlight: Celebrating 9 Years of ROS!



[Wiki](#)
Find tutorials and learn more

[ROS Answers](#)
Ask questions. Get answers

[Blog](#)
Get the latest news

[Forums](#)
Hear the latest discussions

[Press Kit](#)
[Contact Us](#)

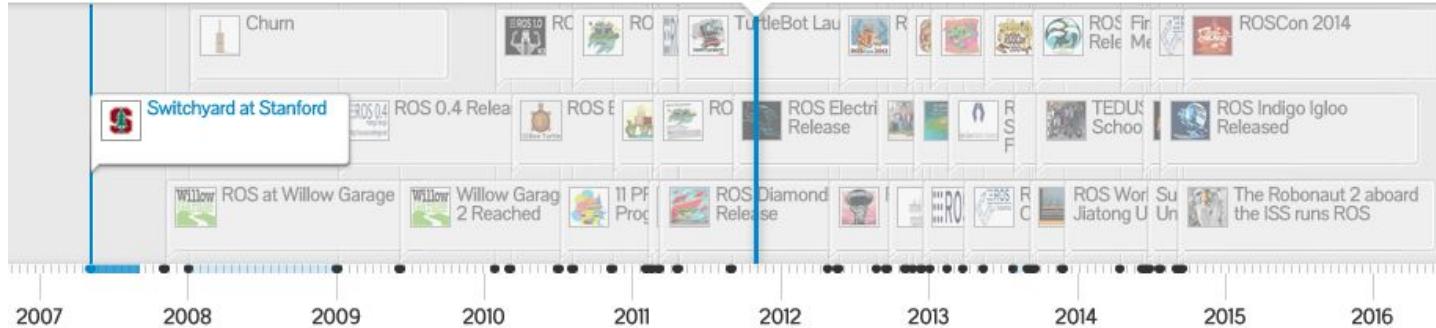


Brought to you by the Open Source Robotics Foundation. [Donate today!](#)
Except where otherwise noted, the ros.org web pages are licensed under Creative Commons Attribution 3.0

[INFO]

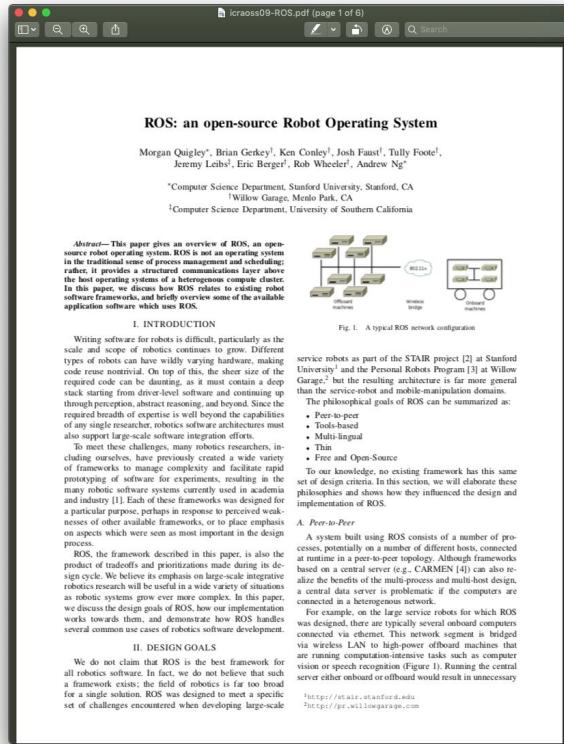
Website: <http://www.ros.org>

History



- Stanford Artificial Intelligence Laboratory (2007)
(Switchyard - STAIR: Stanford Artificial Intelligence Robot)
- Willow Garage (2008 – 2013)
(Robotics research lab and technology incubator)
- Open Source Robotics Foundation (2013 - . . .)
(Open Source Robotics Foundation, Inc.)

The conference paper (January 2009)



@ ICRA Workshop on Open Source Software

Abstract—This paper gives an overview of ROS, an open-source robot operating system. ROS is not an operating system in the traditional sense of process management and scheduling; rather, it provides a structured communications layer above the host operating systems of a heterogeneous compute cluster. In this paper, we discuss how ROS relates to existing robot software frameworks, and briefly overview some of the available application software which uses ROS.

Abstract—This paper gives an overview of ROS, an open-source robot operating system. ROS is not an operating system in the traditional sense of process management and scheduling; rather, it provides a structured communications layer above the host operating systems of a heterogeneous compute cluster. In this paper, we discuss how ROS relates to existing robot software frameworks, and briefly overview some of the available application software which uses ROS.

INTRODUCTION

Writing software for robots is difficult, particularly as the scale and scope of robotics continues to grow. Different types of robots can have wildly varying hardware, making code reuse nontrivial. On top of this, the sheer size of the development community is large, and maintaining a distributed stack working from driver-level software and continuing up through perception, abstract reasoning, and beyond requires the required breadth of expertise is well beyond the capabilities of any single researcher. Robotics software architecture must also support (e.g.,) multi-software integration effects.

To overcome these challenges, many robotics researchers, including ourselves, have previously created a wide variety of frameworks to manage complexity and facilitate rapid prototyping of software for robotics, resulting in the academic and industrial communities [1]. Each of these frameworks was designed for a particular purpose, perhaps in response to perceived weaknesses of other available frameworks, or to place emphasis on aspects which were seen as most important in the design of the framework.

ROS, the framework described in this paper, is the result of a product of tradeoffs and prioritizations made during its development. In this paper, we do not claim that ROS is the best framework for all robotics needs. In fact, we do not believe that such a framework exists, for the field of robotics is far too broad for a single solution. ROS was designed to meet a specific set of challenges encountered when developing large-scale

A. Peer-to-Peer

A system built using ROS consists of a number of processes running on a number of different hosts, connected at runtime in a peer-to-peer topology. Although frameworks based on a central server (e.g., CARMEN [4]) can also realize the benefits of the multi-process and multi-host design, a central data server is problematic if the computers are distributed over a wide geographic area.

For example, on the large service robots for which ROS was designed, there are typically several onboard computers connected via ethernet. This network segment is bridged via wireless to a high-power offboard machine that performs computation tasks such as computer vision or speech recognition (Figure 1). Running the central server either onboard or offboard would result in unnecessary latency.

II. DESIGN GOALS

We do not claim that ROS is the best framework for all robotics needs. In fact, we do not believe that such a framework exists, for the field of robotics is far too broad for a single solution. ROS was designed to meet a specific set of challenges encountered when developing large-scale

<http://lair.stanford.edu>

<http://www.willowgarage.com>

it provides a structured communications layer above the host operating systems of a heterogeneous compute cluster. In this paper, we discuss how ROS relates to existing robot software frameworks, and briefly overview some of the available application software which uses ROS.

PR2 Robot



[INFO]

Video: <https://www.youtube.com/watch?v=HMx1xW2E4Gg>

Salient characteristics thanks to the PR2 Robot

- **A single robot**
- **Workstation-class computational resources on board**
- **No real-time requirements**
(or, any real-time requirements would be met in a special-purpose manner)
- **Excellent network connectivity**
(either wired or close-proximity high-bandwidth wireless);
- **Applications in research, mostly academia**
- **Maximum flexibility, with nothing prescribed or proscribed**
(e.g., “we don’t wrap your main()”)

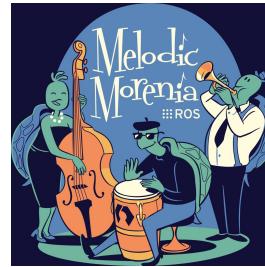
Distribution (= Version)



Kinetic
May 23,
2016



Lunar
May 23,
2017



Melodic
May 23,
2018

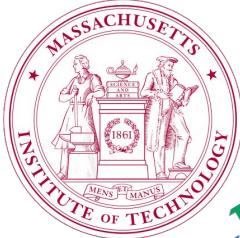
- +/- Ubuntu release schedule (Primarily targeted OS)
- Previous **LTS**: Kinetic Kame (May 2016 – May 2021)
- Lunar Loggerhead (May 2017 – May 2019)
- Current **LTS**: Melodic Morenia (May 2018 – May 2023)
- Platforms: <http://www.ros.org/reps/rep-0003.html>

Used in . . .



- Robotics research
- Competitions
- Robot prototyping
- Robot specific development
- Industry

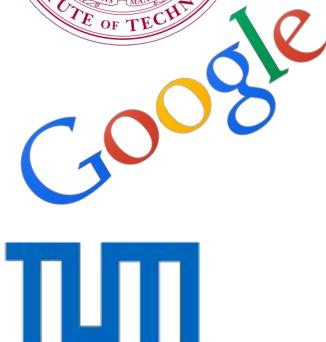
Robotics Research



PXL AI & ROBOTICS LAB



Technische Universität München

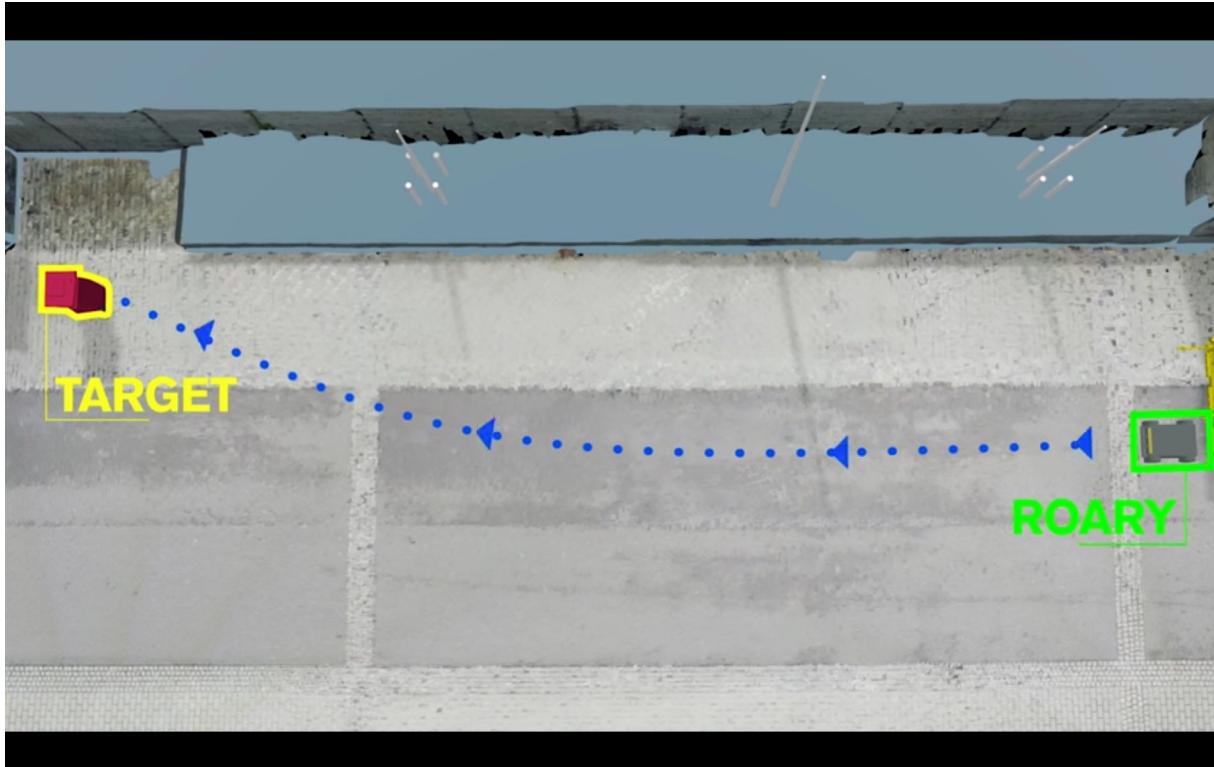


- Grasping/Manipulation
- Human-Robot Interaction
- Motion Planning
- Robot Perception
- Task Planning
- Integration with other technologies
- ...



FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

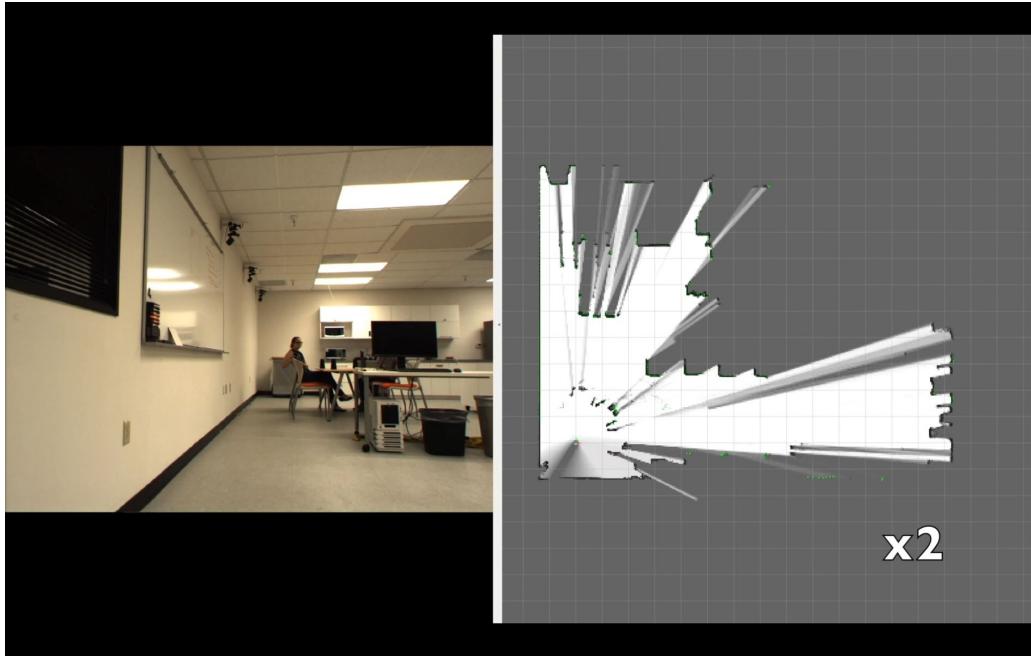
Research: Volvo ROAR Project



[INFO]

Video: <https://www.youtube.com/watch?v=fNIV6Dcj29E> (Volvo Groups)

Research: Toyota Research Institute



[INFO]

Video: <https://www.youtube.com/watch?v=cK6s7soVwws>

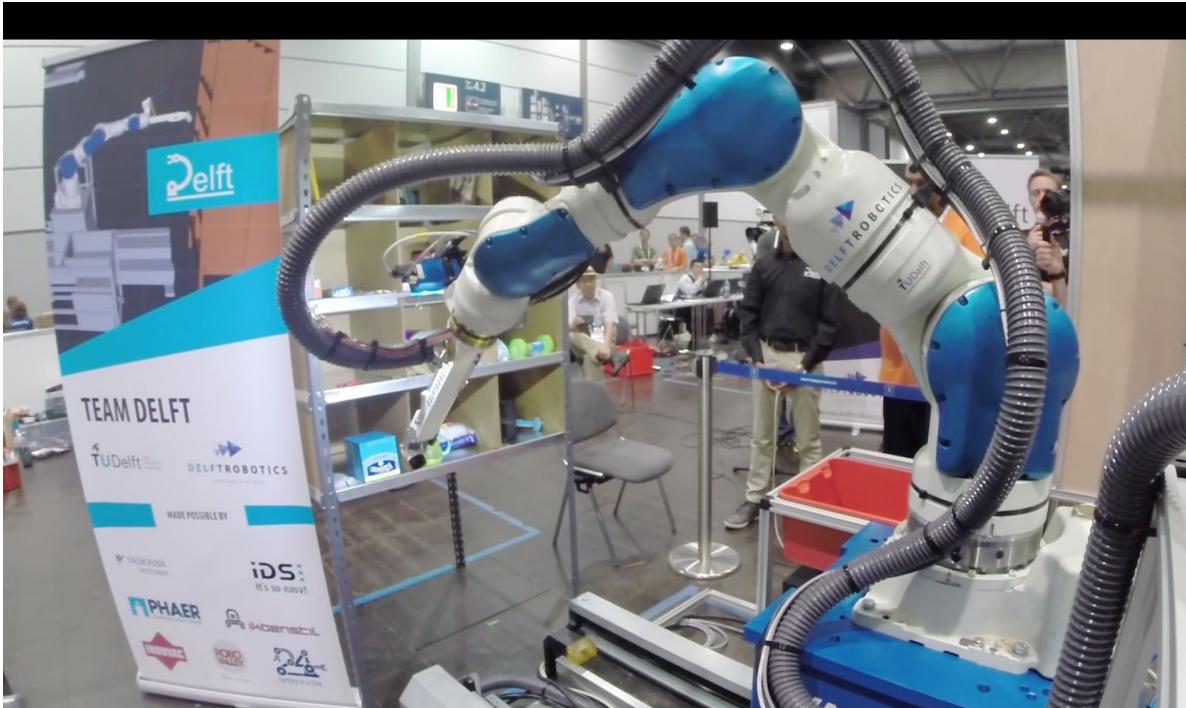
Competition: MASCOR



[INFO]

Video: <https://www.youtube.com/watch?v=D7ZLLalPoAY> (MASCOR FH Aachen)

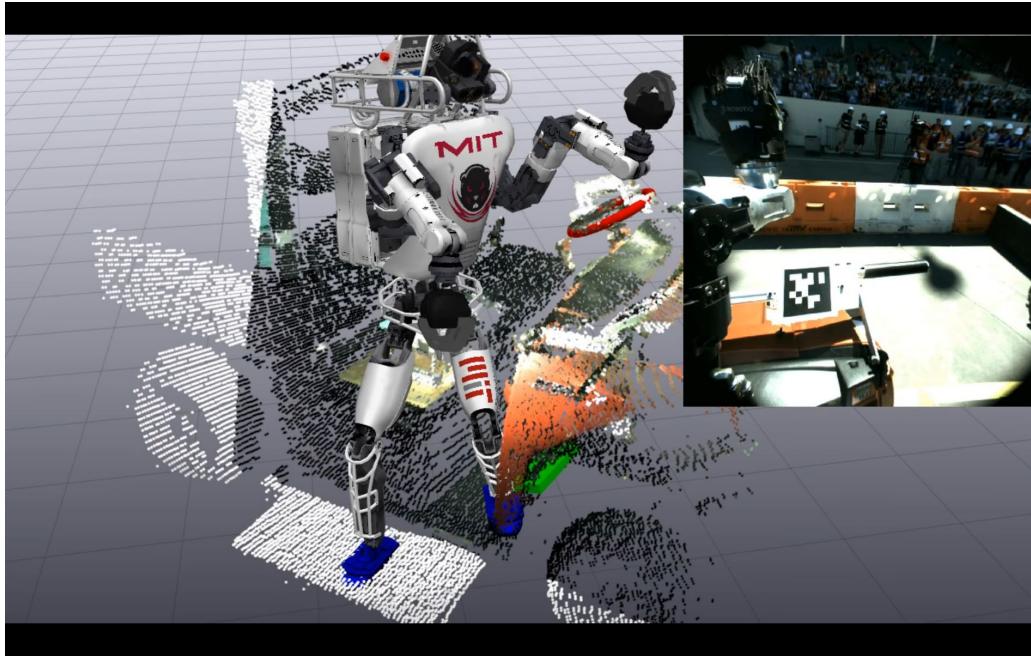
Competition: Amazon Picking Challenge 2016



[INFO]

Video: <https://www.youtube.com/watch?v=3KlzVWxomqs>

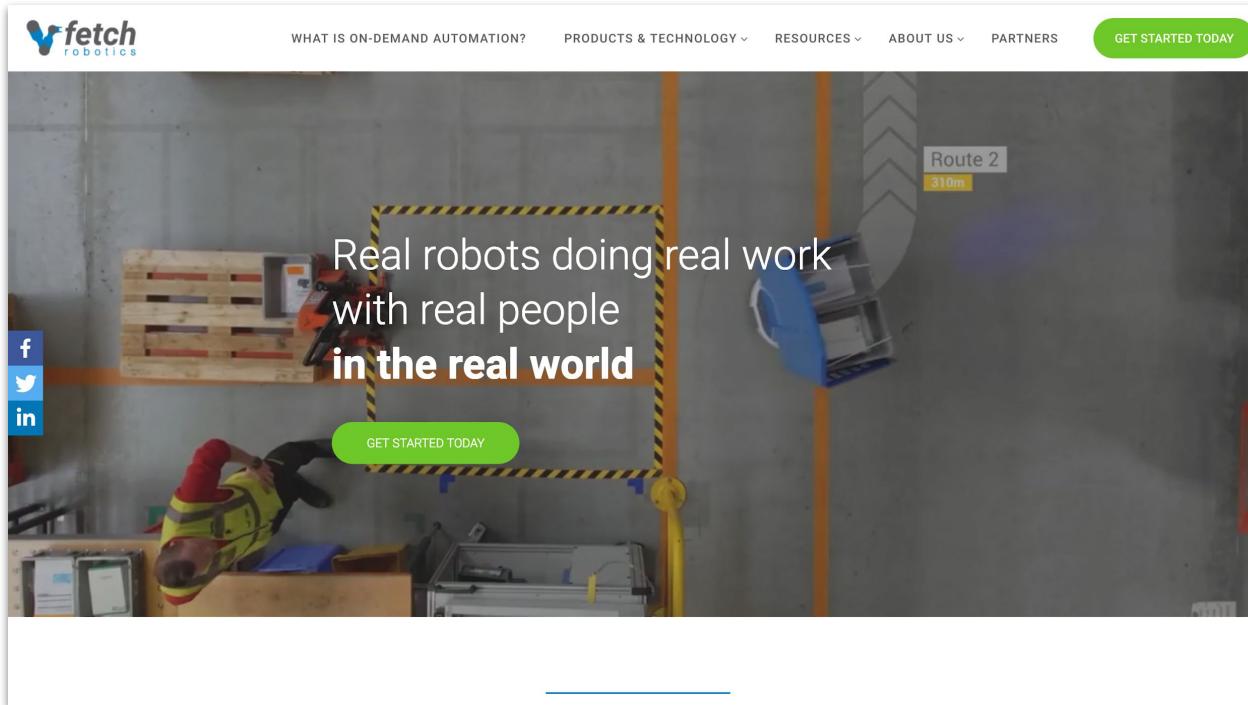
Competition: MIT Atlas in DRC Finals



[INFO]

Video: <https://www.youtube.com/watch?v=em69XtIEEAg>

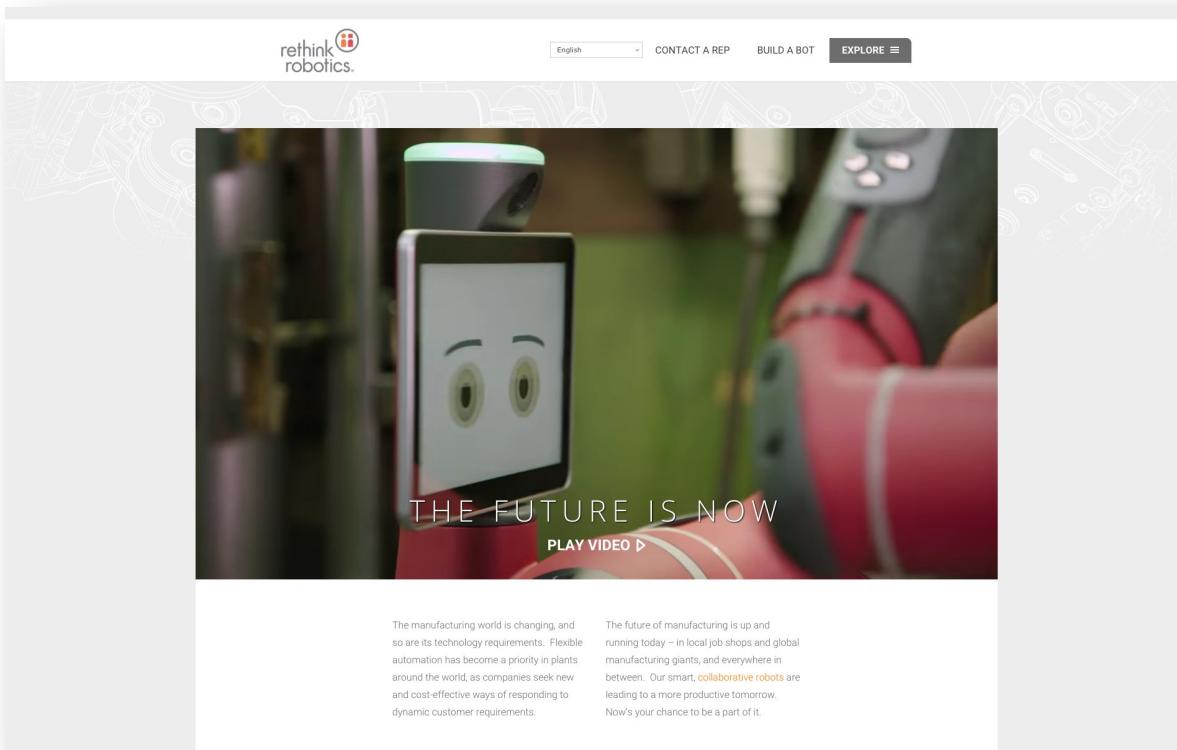
Robot Specific Development



[INFO]

Website: <https://fetchrobotics.com>

Robot Specific Development



The screenshot shows the Rethink Robotics website. At the top, there's a navigation bar with links for "English", "CONTACT A REP", "BUILD A BOT", and "EXPLORE". The main visual is a video thumbnail featuring a small white robot head with a screen displaying cartoon eyes, set against a background of industrial machinery. The text "THE FUTURE IS NOW" is overlaid on the video, along with a "PLAY VIDEO" button. Below the video, two columns of text provide information about the manufacturing industry and Rethink Robotics' role in it.

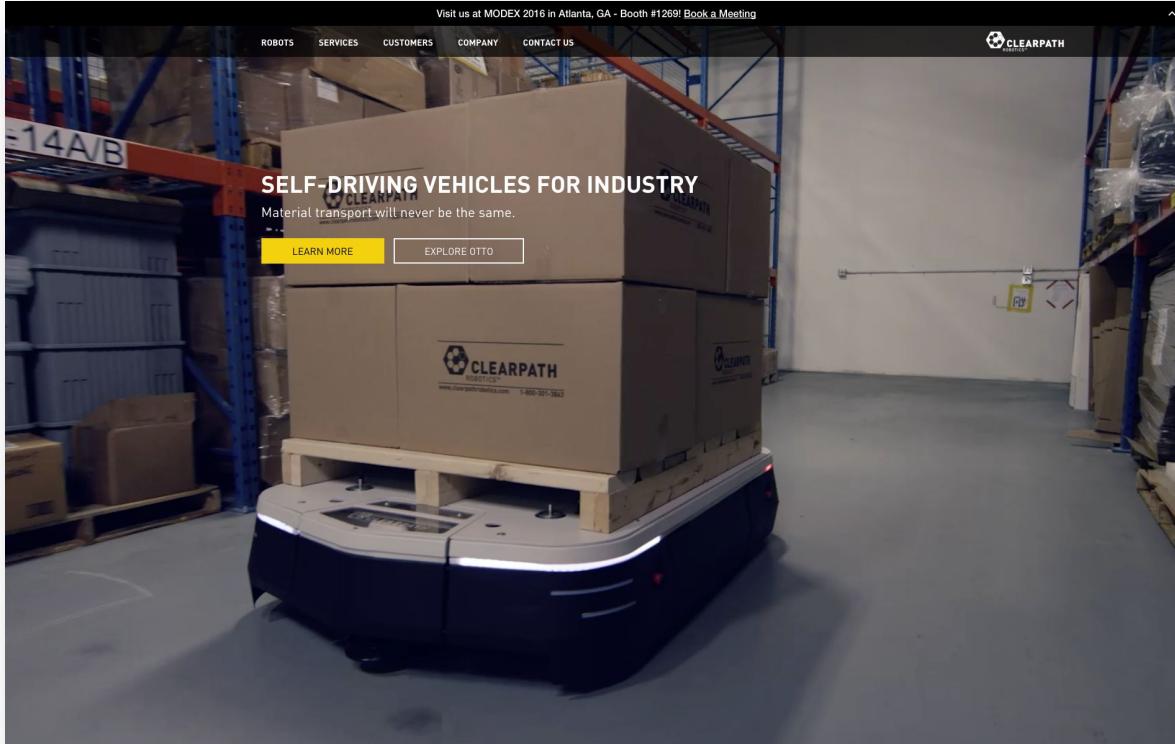
The manufacturing world is changing, and so are its technology requirements. Flexible automation has become a priority in plants around the world, as companies seek new and cost-effective ways of responding to dynamic customer requirements.

The future of manufacturing is up and running today – in local job shops and global manufacturing giants, and everywhere in between. Our smart, **collaborative robots** are leading to a more productive tomorrow. Now's your chance to be a part of it.

[INFO]

Website: <http://www.rethinkrobotics.com>.

Robot Specific Development



[INFO]

Website: <http://www.clearpathrobotics.com>.

Cool ROS projects



[INFO]

Source: <https://www.youtube.com/watch?v=C2OCmsdcZTg>



ROS Industrial



[SOURCE]

<https://rosindustrial.org/ric/current-members>



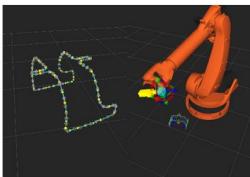
THE ROSIN OPEN CALL:
3+ MILLION EUR AVAILABLE
TO THIRD PARTIES FOR
ROS-INDUSTRIAL
DEVELOPMENT

ROSin

ROS-Industrial
Quality-Assured
Robot Software Components



Focused Technical Projects
(FTPs)



Software Quality Assurance

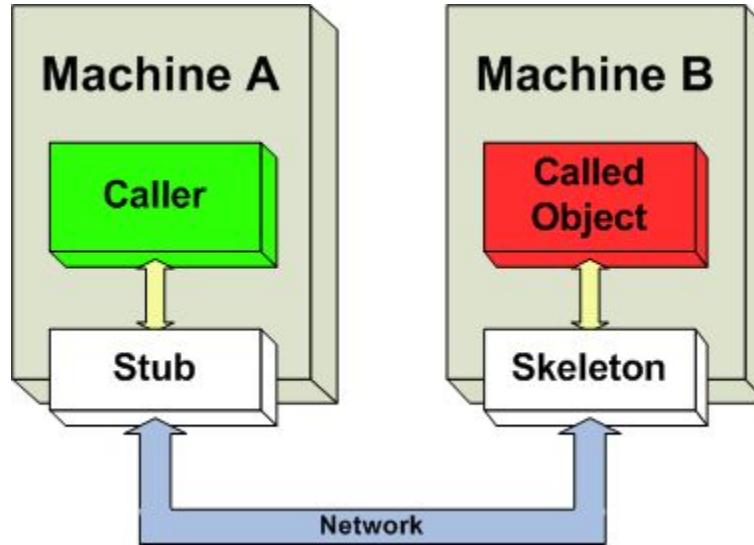


Education

Features

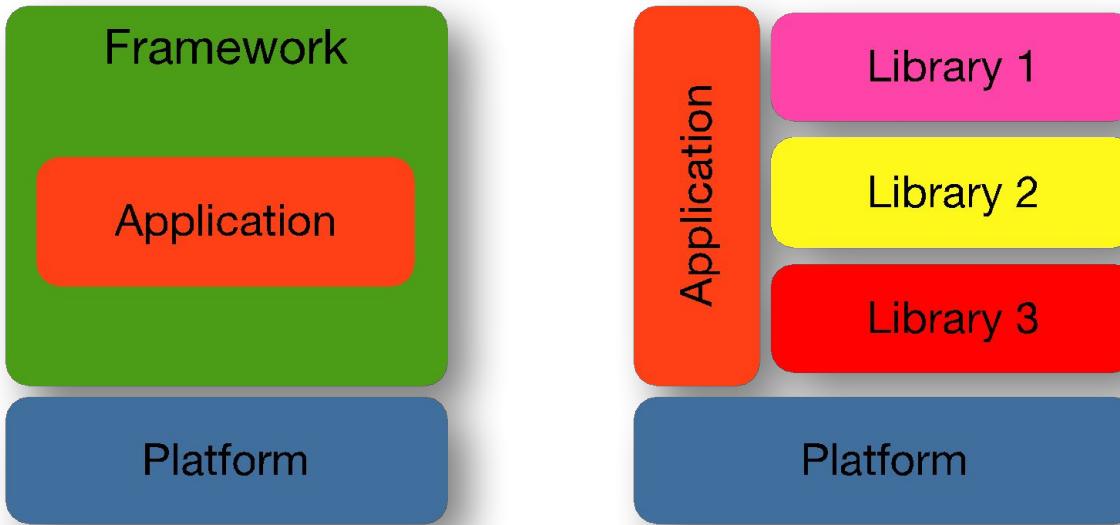
- **Distributed framework**
- Interaction of nodes
- **Robot platform or type independent**
- **Sensor independent**
- Easy testing (Real Robot Simulation)
- **Programming language independent**
(Standard language support: C++/Python/LISP)
- Huge amount of code sharing

Distributed



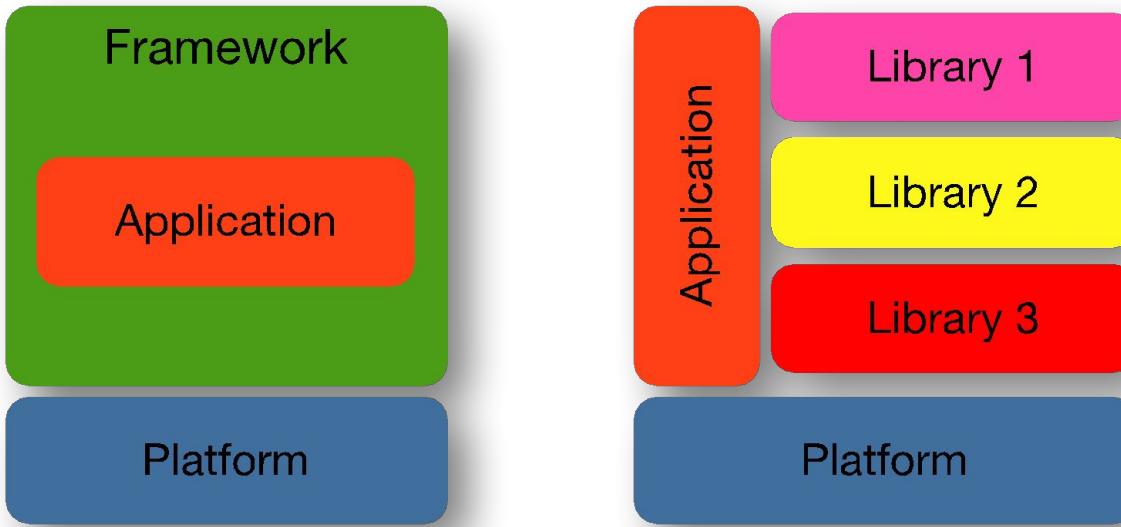
*Multiple computers connected via a network or in different processes inside the same computer.
(Inter-process & Inter-machine communication)*

Framework (vs Libraries)



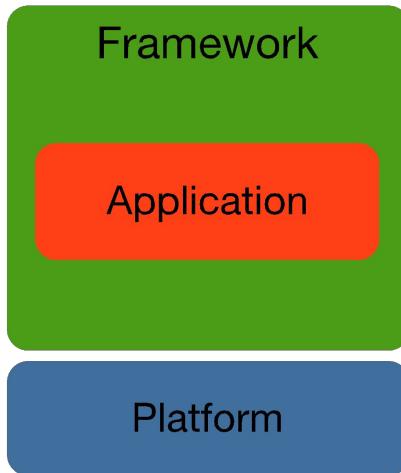
- Platform: Windows, Linux, OS X, Java, .NET, ...
- Framework: Java Spring, ASP MVC, ...
- Libraries: JUnit, Log4j, joda-time, ...

Framework (vs Libraries)



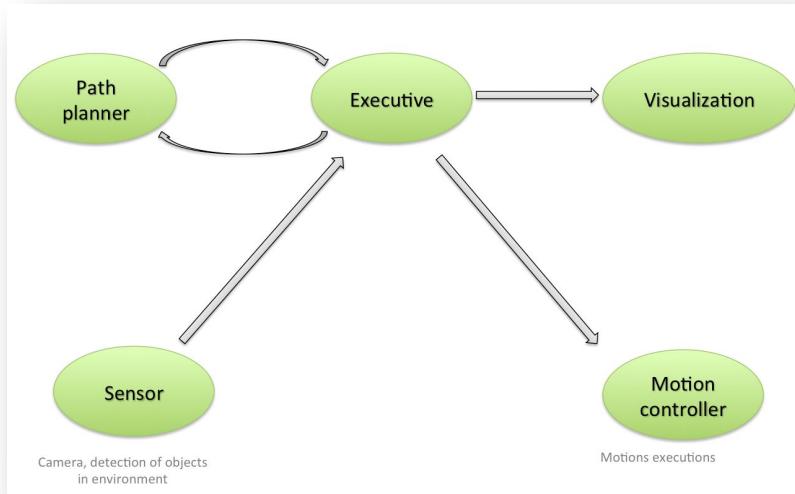
- Platform: Base layer. (OS, VM)
- Framework: Application lives inside the framework.
- Libraries: Application lives beside the libraries.

Distributed Framework



- Controls the application lifecycle
- Regulates (all) communication
- Multi process/machine
- Facilitates abstractions
- Provides standardization
- (Partially) Extensible
- Can contain multiple libraries

Interaction of Nodes



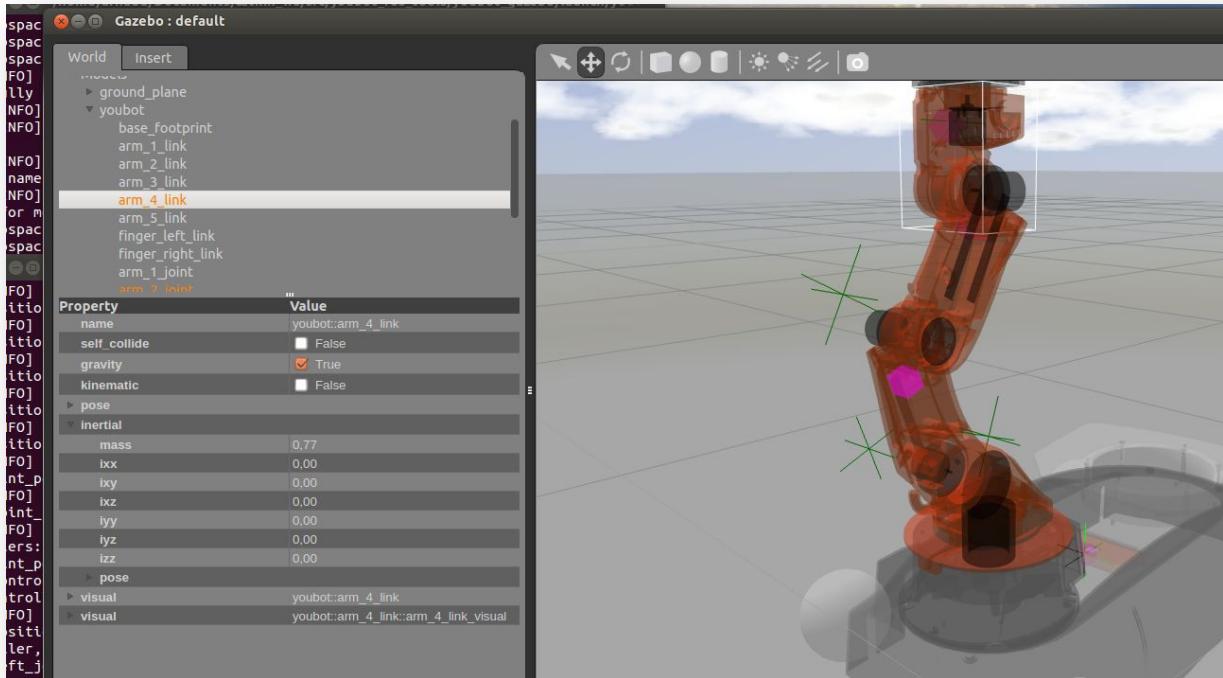
- Node based (Node graph)
- 1 project = many nodes
- Distributed
- Message passing
- Robust

Robot & Sensor Independent



- Message format standardization where possible
- Interchangeable nodes / Node reuse

Easy Testing



Gazebo
Simulator

Programming Language Independent

- Via client libraries
- 3 main client libraries
(C++: roscpp, Python: rospy, LISP: rosłisp)
- Experimental client libraries
 - Java: rosjava
 - .NET: roscs
 - Pharo: PhaROS
 - Ruby: rosruby
 - Lua: roslua
 - EusLisp: roseus
 - R: rosR
 - Julia: RobotOS.jl (wraps rospy)
- Create your own
<http://wiki.ros.org/Implementing%20Client%20Libraries>

Code Sharing (Active Community)

STANFORD
UNIVERSITY

Colorado
University of Colorado at Boulder



Carnegie Mellon

BOSCH



Washington
University
in St. Louis

THE UNIVERSITY OF
TEXAS
AT AUSTIN™



TUM
TECHNISCHE
UNIVERSITÄT
MÜNCHEN

PXL AI & ROBOTICS LAB

UNIVERSITY OF
MARYLAND

Georgia Tech

Penn

COLUMBIA
UNIVERSITY

KATHOLIEKE UNIVERSITEIT
LEUVEN

BROWN

MIT

USC
UNIVERSITY
OF SOUTHERN
CALIFORNIA

A

WPI

ROS is NOT

- A programming language
- (Only) A library
- An integrated development environment
- A hard real-time architecture

Why ROS?

- Software sharing
- Fast development
- Fast results
- Free & Open source (BSD license)
- Suitable for large scale projects/research

Why ROS?

Fast development for new robotic
systems without reinventing the wheel.

(By using standard tools & interfaces.)

ROS Examples (<http://robots.ros.org>)

ROS Robots

≡ 

Ox Series



Category: ground
Resources: [Wiki](#)

Ox series robots (pronounced as Ox) are the most versatile, accurate and durable mobile robotics platforms by Nex Robotics.

diff-drive ground mobile base

ABB Manipulators



Category: component
Resources: [Website](#) [Wiki](#)

ROS-Industrial support for ABB manipulators

ROS-Industrial abb arm groovy hydro indigo

ADAS Development Vehicle Kit



Category: ground
Resources: [Wiki](#)

A complete hardware and software solution that allows seamless control of throttle, brake, steering, and shifting.

ackermann ground

AGVS



Category: ground
Resources: [Website](#) [Wiki](#)

AGVS robot is a robot for logistics transport. It is motorized by two motors controlling direction and traction. The robot has a low profile chassis, making it suitable for movement under hospital roller containers.

ground indoor mobile robot robotnik autonomous logistics robot

AMIGO



Category: ground
Resources: [Wiki](#)

AR10



Category: component
Resources: [Website](#) [Wiki](#)

ROS Examples



Operating System ROS 8 years: https://www.youtube.com/watch?v=z70_3wMFO24

ROS Examples



Toyota HSR:

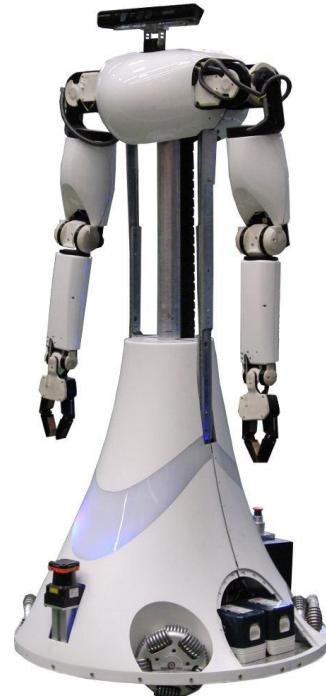
<https://www.youtube.com/watch?v=d29ylyjRr08>

ROS Examples



PAL Robotics: Reem-C

ROS Examples



AMIGO

ROS Examples



Otto Bock SensorHand Speed

ROS Examples



Grizzly

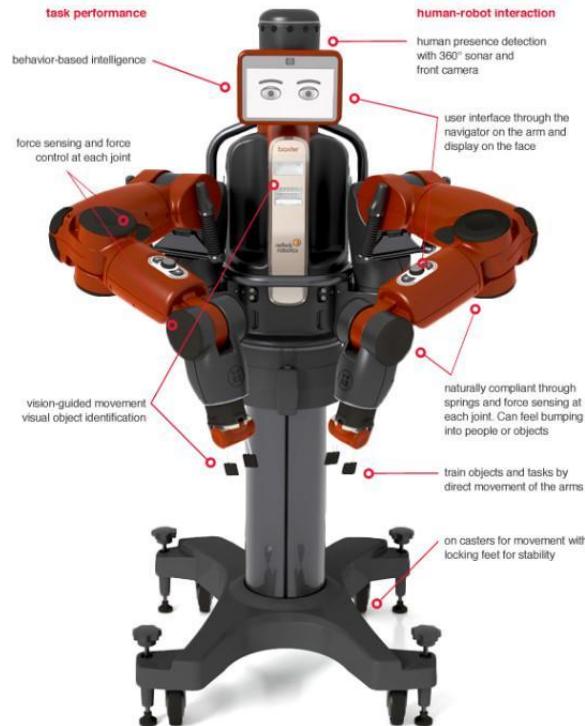
ROS Examples



ROS Examples



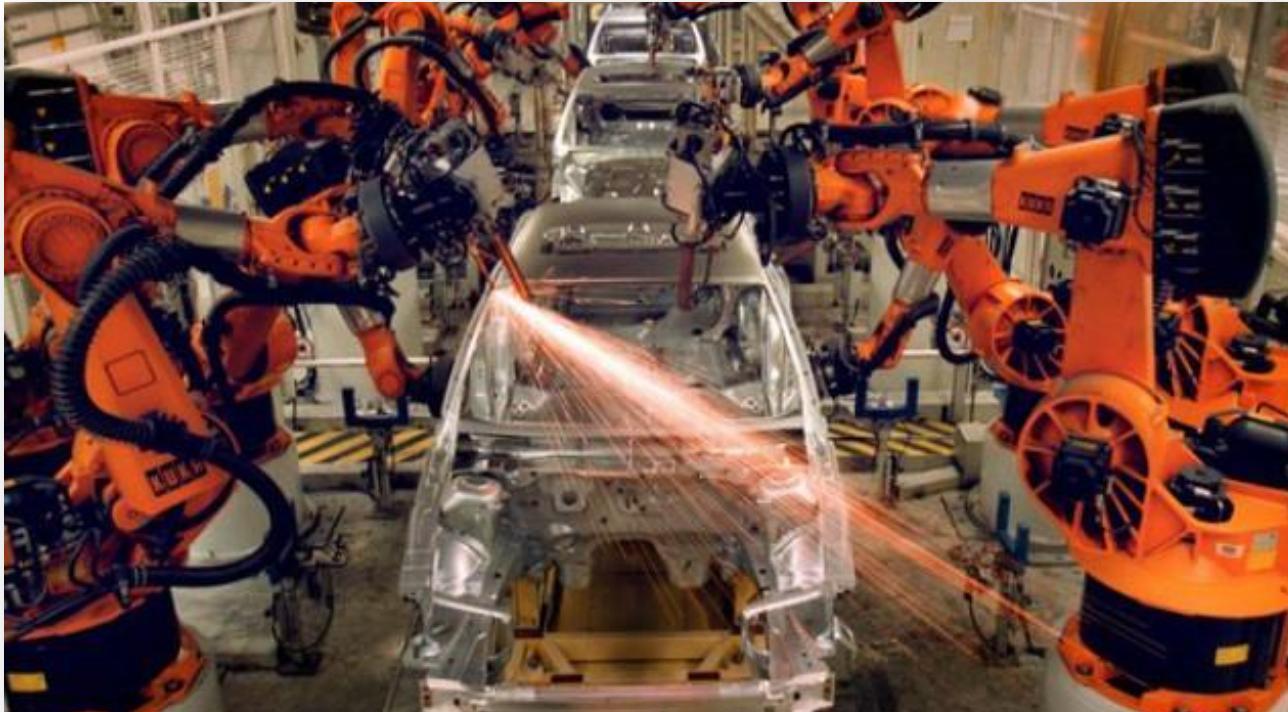
ROS Examples



ROS Examples



ROS Examples



ROS Examples



ROS Examples



ROS Examples



ROS Examples



ROS Examples



ROS Examples

The screenshot shows the GitHub repository page for `dji-sdk / Onboard-SDK-ROS`. The repository has 90 issues, 14 pull requests, 0 projects, and 176 insights. It features 252 commits, 8 branches, 25 releases, and 15 contributors. The latest commit was on 8 Jan. The repository contains files like `ISSUE_REPLY_TEMPLATE.md`, `README.md`, and `ReadMe.md`. A section titled "DJI Onboard SDK ROS 3.7" discusses the latest update, which was released on 14 Aug 2018, adding new telemetry topics, an emergency stop API, and support for the A3, N3, and M210/M210RTK platforms.

Official ROS packages for DJI onboard SDK.

Branch: 3.7 ▾ New pull request Create new file Upload files Find file Clone or download ▾

dji-dev Update ISSUE_REPLY_TEMPLATE.md file. Latest commit 5325b65 on 8 Jan

.github Update ISSUE_REPLY_TEMPLATE.md file. 2 months ago

dji_sdk Added hardware check for RC topics. 6 months ago

dji_sdk_demo OSDK 3.6 Release: New camera demos and several fixes and improvements... a year ago

.gitignore OSDK ROS 3.3 2 years ago

ReadMe.md OSDK-ROS 3.7: Support for many new telemetry topics, a few new APIs a... 7 months ago

README.md

DJI Onboard SDK ROS 3.7

Latest Update

OSDK-ROS 3.7 was released on 14 Aug 2018. This release adds many new telemetry topics, an emergency stop API and more for the A3, N3 and M210/M210RTK platforms. Please see the [release notes](#) for more information.

ROS Examples



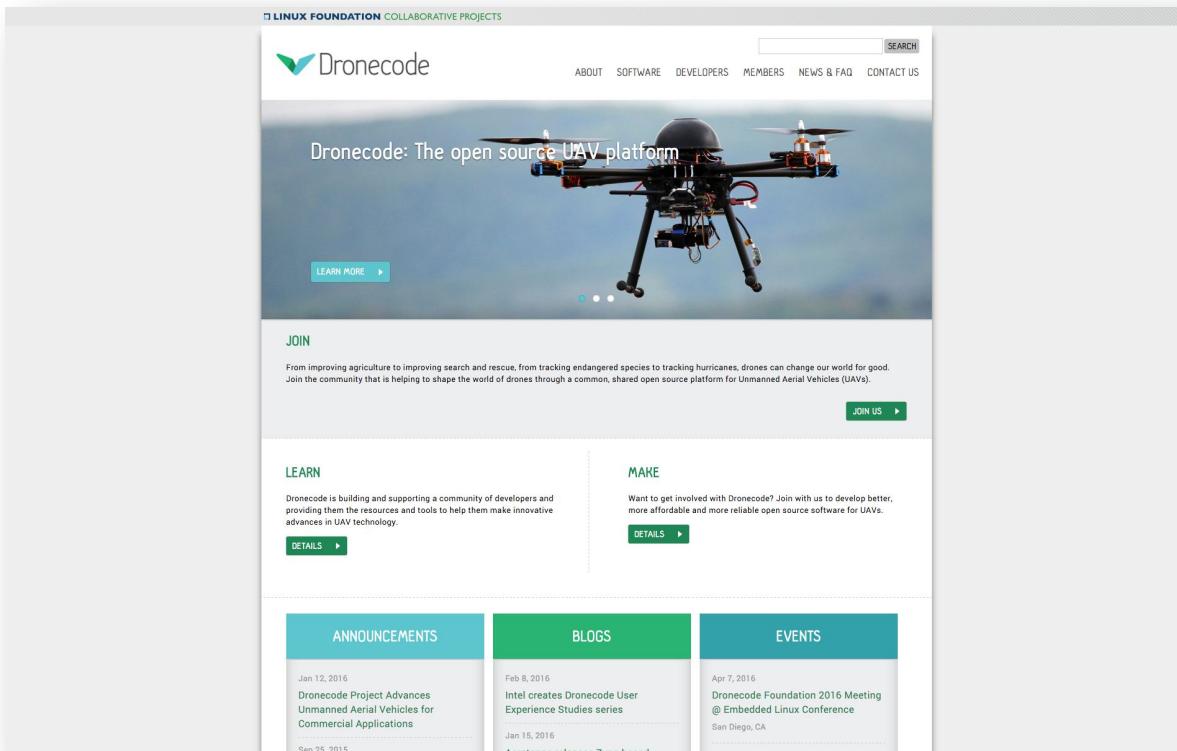
ROS Examples



ROS Examples



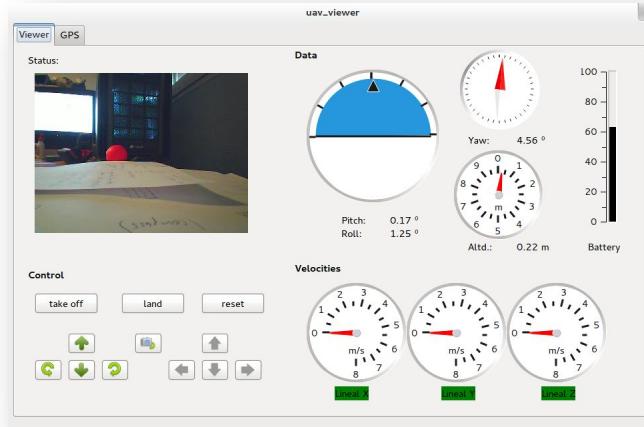
ROS Examples



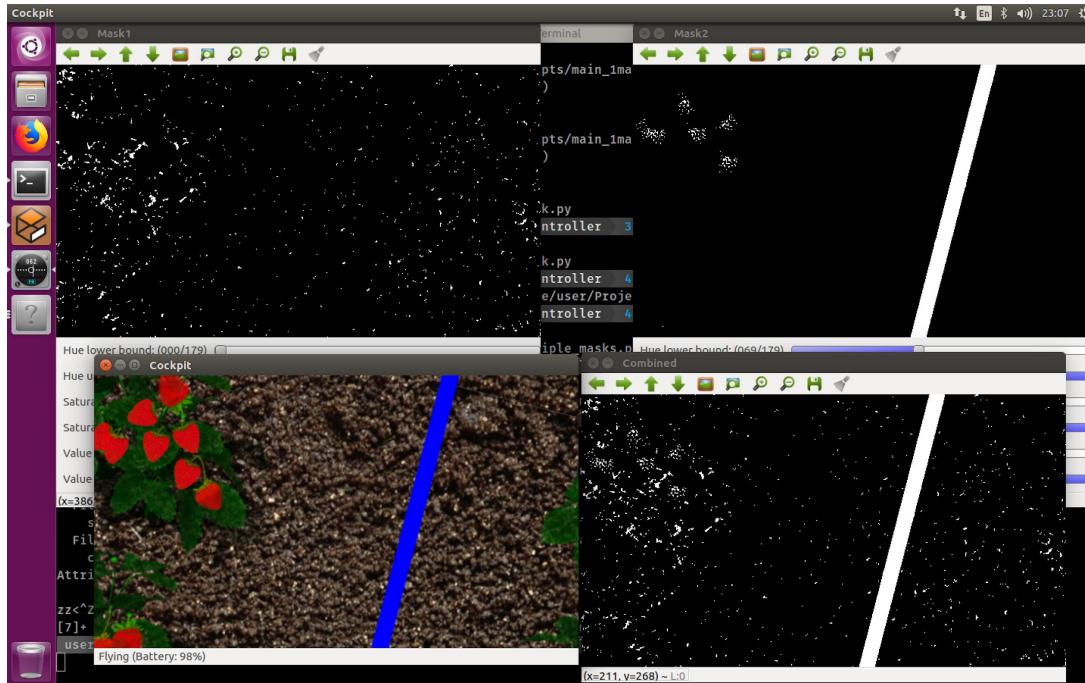
[INFO]

Website: <https://www.dronecode.org>.

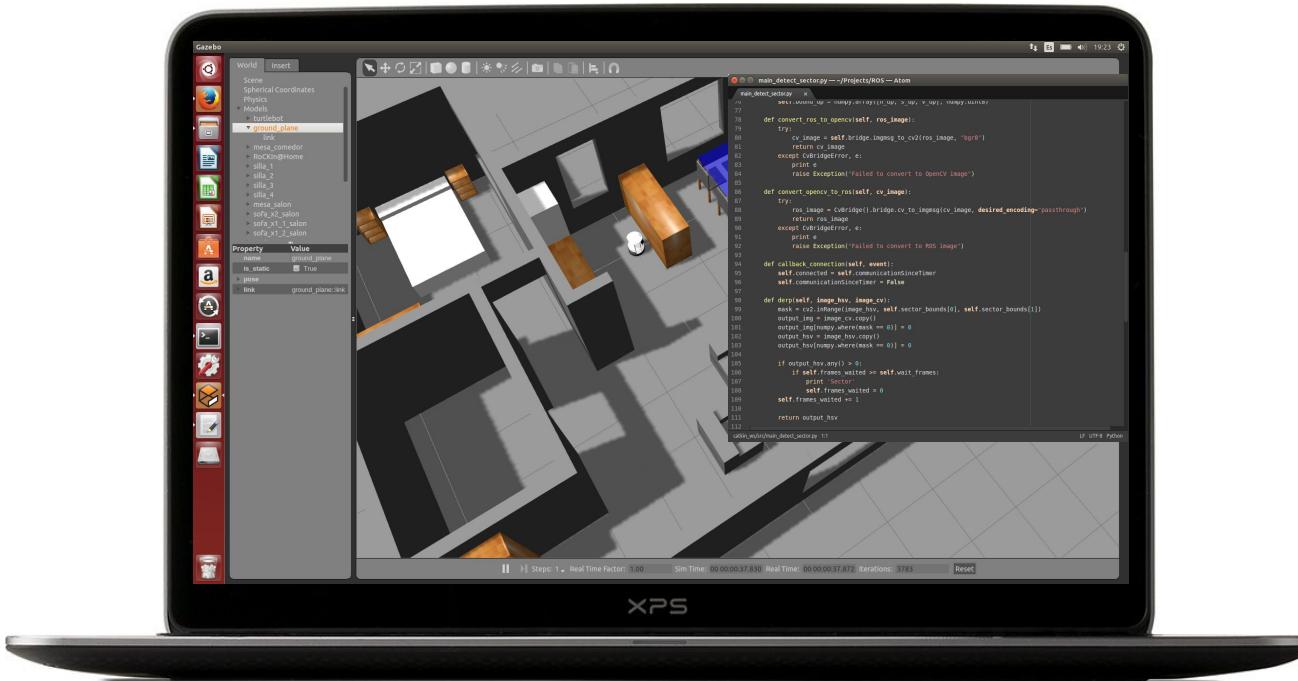
ROS Examples



Downside: slow progression to new versions



Downside: slow progression to new versions



Downside: slow progression to Python 3

M

Become a member [Sign in](#) [Get started](#)

How to setup ROS with Python 3

 Beta [Follow](#)
Feb 27 · 3 min read

While working on [ROS package for Anki Vector](#) I had no choice but to use Python 3.6 as Anki's Python SDK was specifically built for that version. Doing so I found that setting up ROS with Python 3 isn't such a trivial task. It's also not extremely complex, but there is no single guide that says exactly how to do it properly—so I decided to write one. This guide was written for ROS Melodic but a lot of it applies also for Kinetic.

In case you just started to build your Python 3 package and you want to make sure you've fixed all of the issues we're going to discuss beforehand, you can use a [small ROS Package I wrote](#) to reproduce them and see.

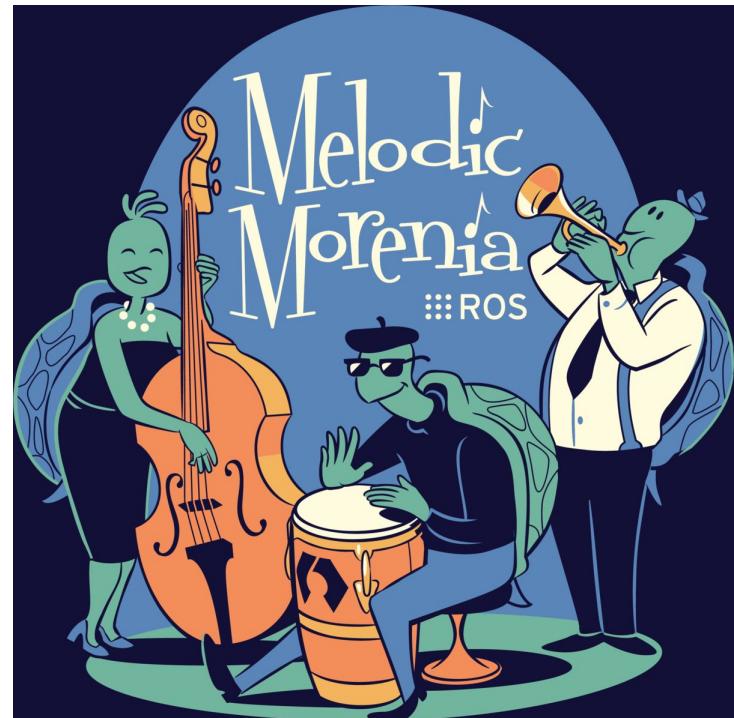
Also— ROS works best with Python 2 (as you probably know if you're reading this), ROS2, on the other hand, was built with Python 3 in mind. So if you don't need ROS(1) specifically but rather simply interested in developing for ROS using Python 3 you should consider moving to ROS2. Otherwise, follow these steps:

 Never miss a story from **Beta**, when you sign up for Medium. [Learn more](#) [GET UPDATES](#)

[INFO]

Website:

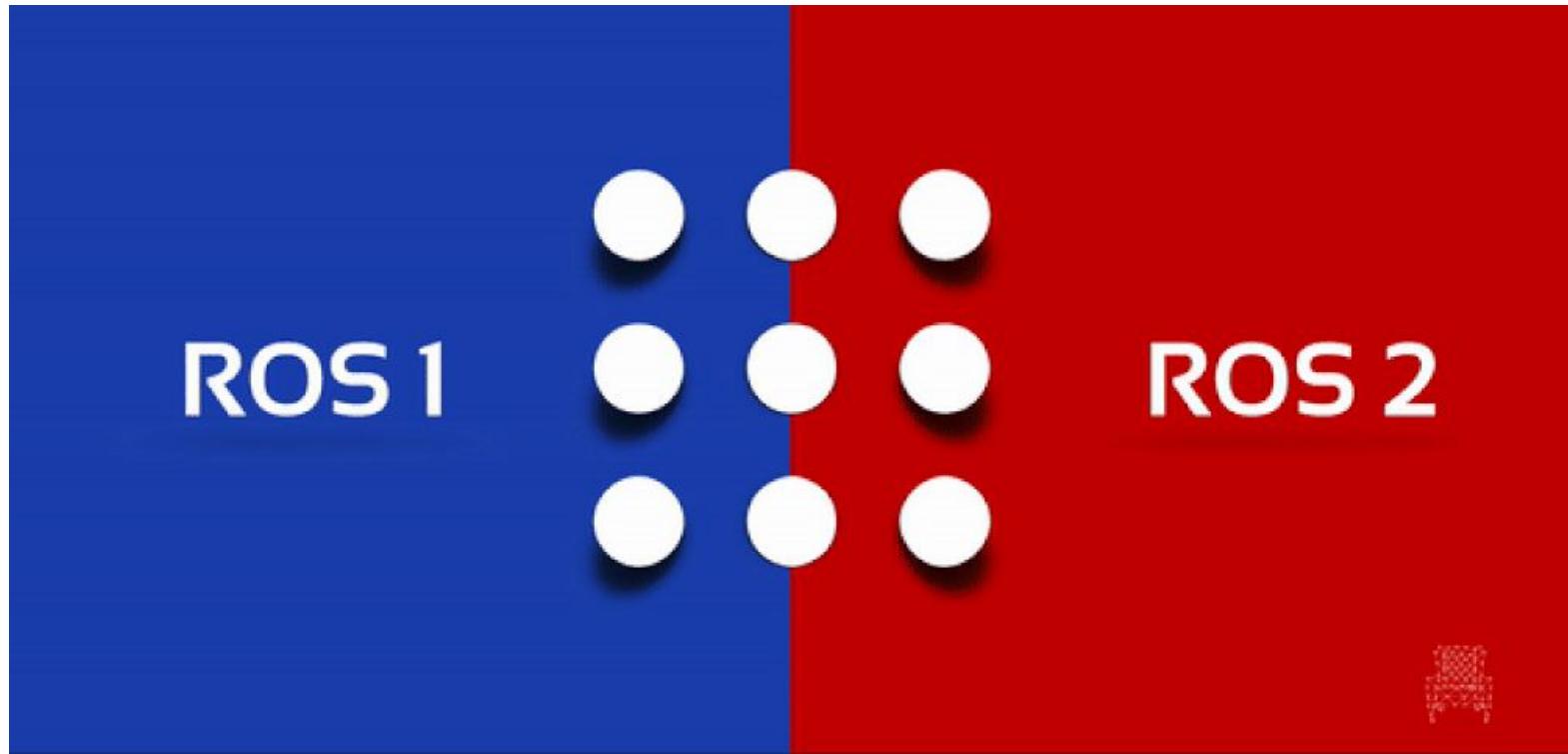
ROS 1



For this course: Melodic on Ubuntu 18.04



ROS 2: The next gen (for next year!)



ROS 2: New use cases



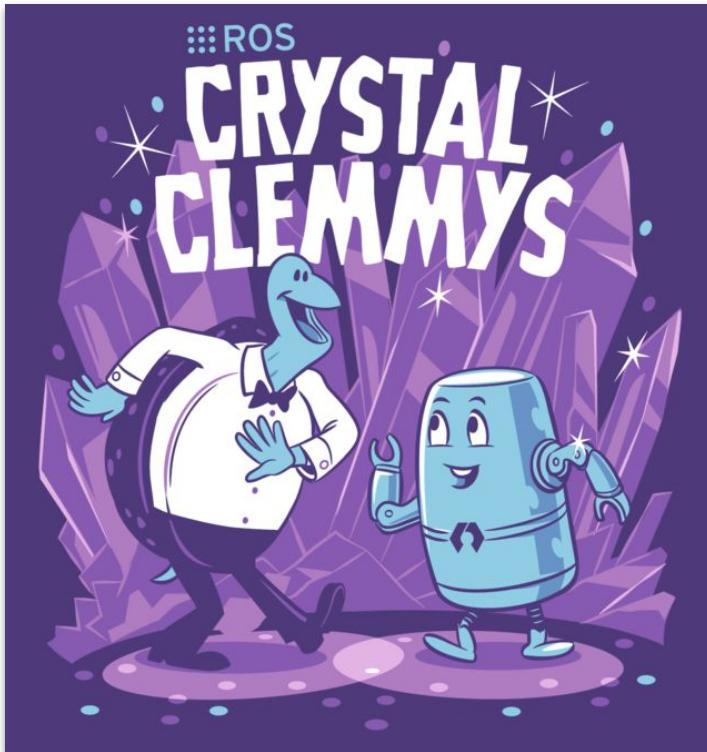
- Teams of multiple robots
- Small embedded platforms
- Real-time systems
- Non-ideal networks
- Production environments
- Prescribed patterns for building and structuring systems

ROS 2: New technologies



- Zeroconf
- Protocol Buffers
- ZeroMQ (and the other MQs)
- Redis
- WebSockets
- DDS (Data Distribution Service).

ROS 2



Crystal Clemmys is the third ROS 2 release and will be supported with bug fixes and platform updates (particularly on rolling dependencies like Windows and MacOS) for one year with **support ending in December 2019**. While we do aim to keep the API as stable as possible, **we can't guarantee 100% API compatibility between releases.**

