Course of

Robot Programming with ROS 2

Day 1

1. Introduction

ikerlan

Universidad Rey Juan Carlos

Intelligent Robotics Lab

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

Overview Repositories 136 Projects Packages Stars 17

Francisco Martín Rico
fmrico

Doctor Engineer on Robotics. Full Professor at Rey Juan Carlos University. Leader of the Intelligent Robotics Lab

Edit profile

343 followers · **14** following

Rey Juan Carlos University
Madrid
fmrico@gmail.com
www.gsyc.es/~fmartin
@fmrico

Achievements

x2 x3

Beta Send feedback

Highlights

PRO

Organizations

Pinned Customize your pins

PlanSys2/ros2_planning_system Public
This repo contains a PDDL-based planning system for ROS2.
C++ 300 66

ros2_dotnet Public
Forked from ros2-dotnet/ros2_dotnet
.NET bindings for ROS2
C# 1

cascade_lifecycle Public
Managed nodes (or lifecycle nodes, LN) package provides a mechanism to create LifecycleNode activation trees
C++ 47 5

navigation2 Public
Forked from ros-planning/navigation2
ROS2 Navigation
C++ 1 1

rclcpp Public
Forked from ros2/rclcpp
rclcpp (ROS Client Library for C++)
C++

popf Public
The POPF planner from KCL planning group with some modifications to make it work with "modern" compilers...
C++ 13 7

697 contributions in the last year Contribution settings

Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May
Mon
Wed
Fri

Learn how we count contributions Less More

Contribution activity

June 2023

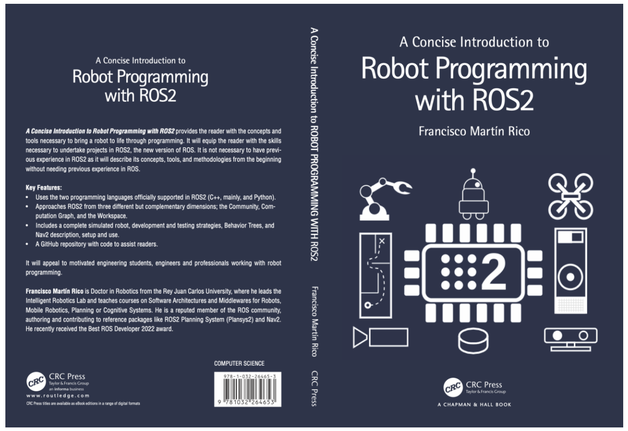
fmrico has no activity yet for this period.

Show more activity

Seeing something unexpected? Take a look at the GitHub profile guide.

2023 2022 2021 2020 2019 2018 2017 2016 2015

Universidad Rey Juan Carlos



TSC

Intelligent Robotics Lab

CC BY NC ND

José Miguel Guerrero
Profesor CD
Visión Artificial / Virtualización

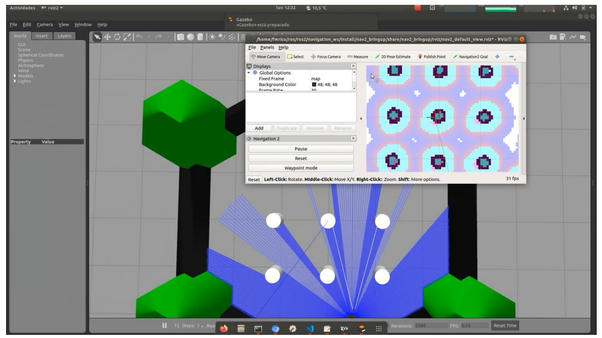
Intelligent Robotics Lab

Rodrigo Pérez
Profesor AyD
Rob. Asistencial / Behavior Trees

Juan Diego Peña
PhD Std / Inv. Contratdo
Mobile Robots / Network

Alberto García
PhD Std / Inv. Contratdo
Mobile Robots / Navigation

Juan Carlos Manzanares
Técnico de Laboratorio
Mobile Robots / Navigation



F. MARTIN, J. GINES, F. J. LERA, A.M. GUERRERO, V. MATELLAN

MANAGING VISUAL ATTENTION IN SOCIAL ROBOTS

FRONTIERS IN NEUROSCIENCE

Intelligent Robotics *Lab*

PlanSys2

IROS 2021

A PLANNING SYSTEM FRAMEWORK

FRANCISCO MARTÍN, JONATAN GINÉS, VICENTE MATELLÁN, AND FRANCISCO J. RODRÍGUEZ

NAV2

IROS 2021

REGULATED PURE PURSUIT FOR ROBOT PATH TRACKING

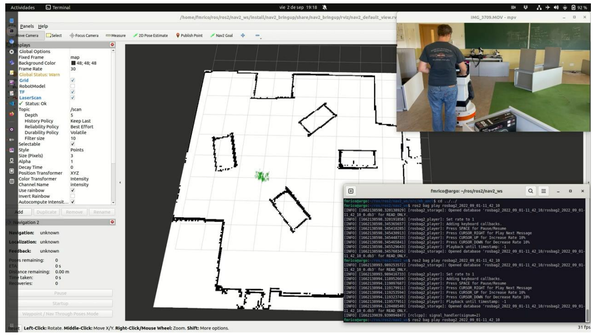
STEVE MACENSKI, SHRIJIT SINGH, FRANCISCO MARTÍN AND JONATAN GINES

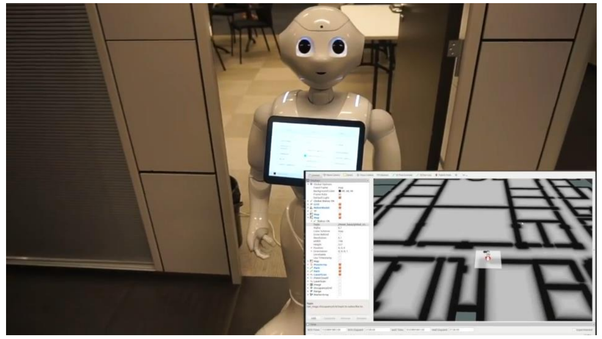
THE MARATHON 2

A NAVIGATION SYSTEM

STEVE MACENSKI, FRANCISCO MARTIN, RUFFIN WHITE, JONATHAN GINES

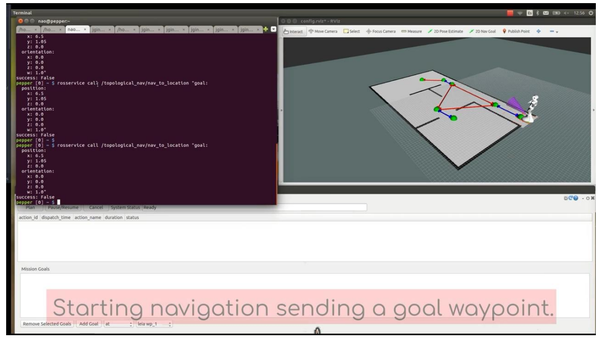
IROS 2020





Switch!

Dynamic Role Exchange



DMARCE/CASCAR PROJECT

NAVIGATING WITH KEEPOUT ZONES

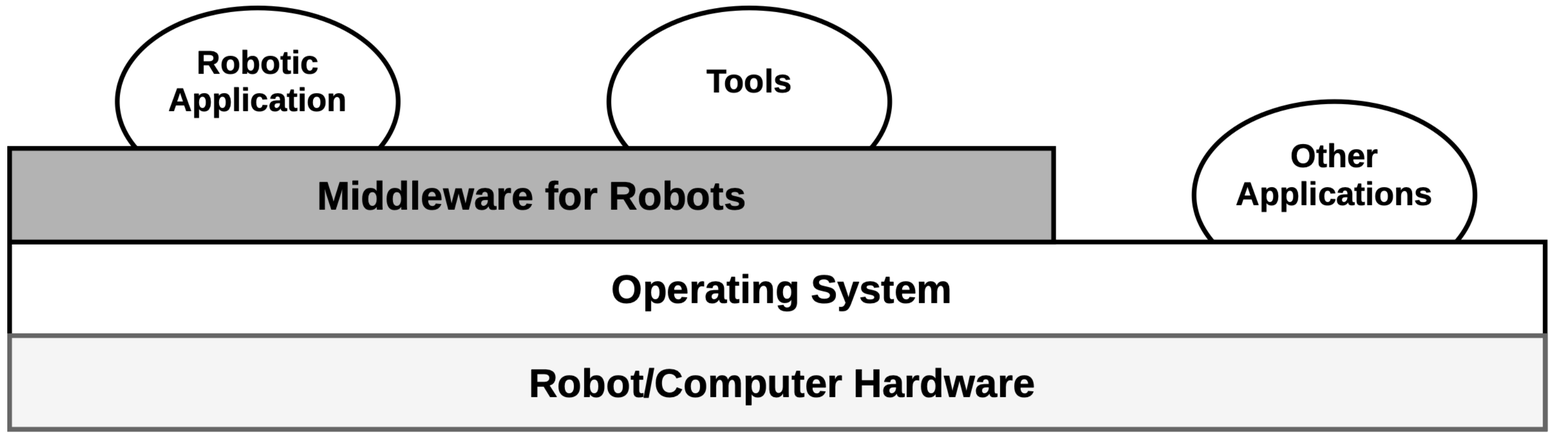
SEPTIEMBER 2022

Intelligent Robotics *Lab*

CC BY NC ND

Programming Robots

- Robots must be programmed to be useful
- We need Middlewares
- Robot programming middlewares provide drivers, libraries, and methodologies
- Few of them have survived the robot for which they were designed or have expanded from the laboratories where they were implemented
- The big difference is the ROS developers community around the world.



ROS (Robot Operating System)

“The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications. From drivers to state-of-the-art algorithms, and with powerful developer tools, ROS has what you need for your next robotics project. And it's all open source.”

http://www.ros.org/

Intelligent Robotics Lab

CC BY NC ND

- ROS and ROS2
- Lot of tutorials and documentation
- We will use Ubuntu 22.04 + Humble

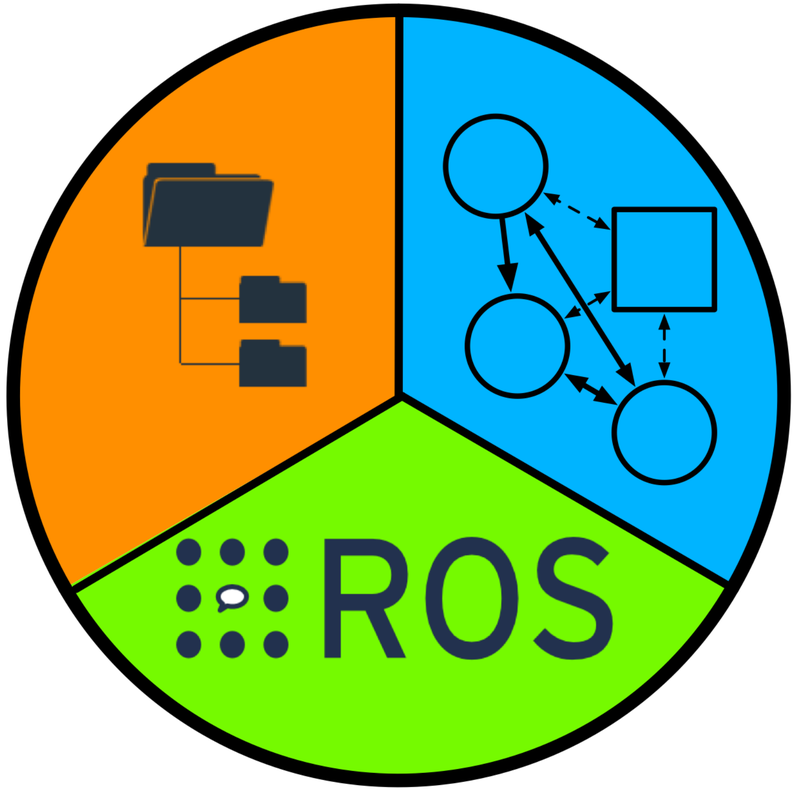
Intelligent Robotics Lab
CC BY SA

History

- ROS was born in 2006 at Stanford, within STAIR project
- Willow Garage in 2007
- Open Source Robotics Foundation in 2013
- Since 2013 is considered the standard in Robotics

ROS Dimensions

Workspace: the set of software installed on the robot or computer, the programs that the user develops, and tools to build

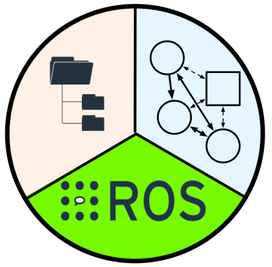


Computation Graph: a running ROS2 application

Community: vast community of developers who con- tribute with their own applications and utilities through public repositories, to which other developers can contribute

Intelligent Robotics Lab

CC BY SA



The Community

- Open Source and Licenses
- ROS2 organizes software development in federal model
- Packages and distributions
- Online resources

ROS

Open Source Robotics Foundation

MIT LICENSE GNU LICENSE BSD-2

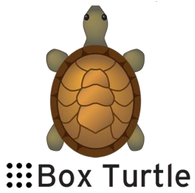
OPEN SOURCE LICENSE

APACHE LICENSE 2.0 BSD-3

Intelligent Robotics Lab

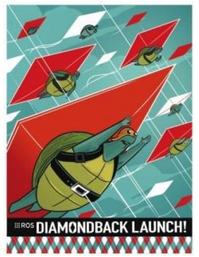
CC BY SA

ROS Distributions

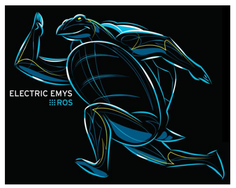


Mar 2, 2010

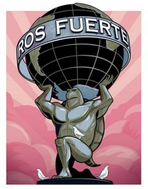
Aug 2, 2010



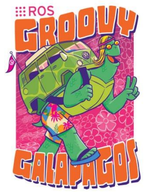
Mar 2, 2011



Aug 30, 2011



Apr 23, 2012



Dec 31, 2012

Sept 4th, 2013

Jul 22nd, 2014

May 23rd, 2015

May 23rd, 2016

May 23rd, 2017

May 23rd, 2018

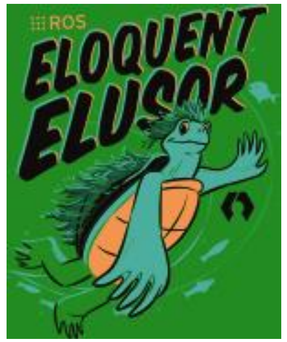
May 23rd, 2020

Intelligent Robotics Lab

ROS 2 Distributions

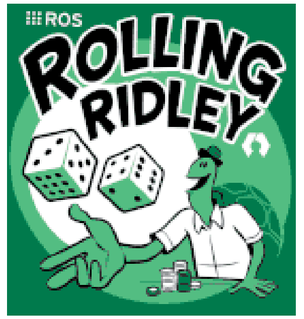
Dec 8th, 2017

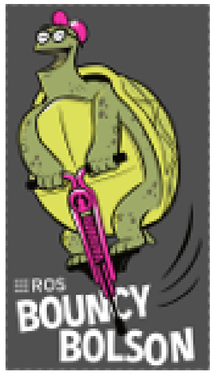
Dec 14th, 2018



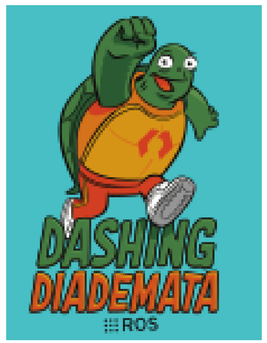
Nov 22nd, 2019

May 23rd, 2021

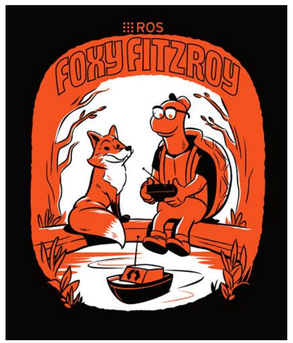




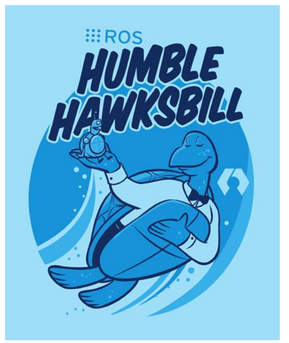
July 2nd, 2018



May 31st, 2019



Jun 5th, 2020

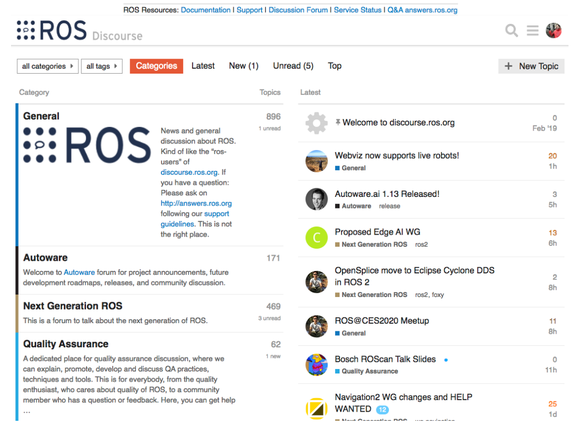


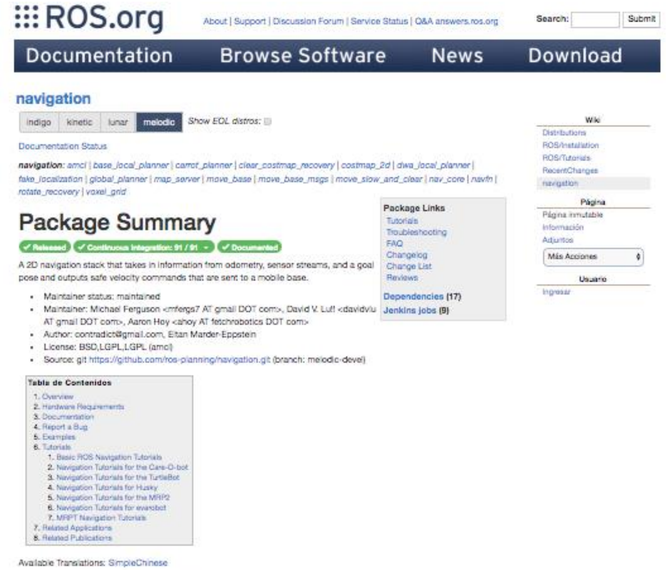
May 23rd, 2022

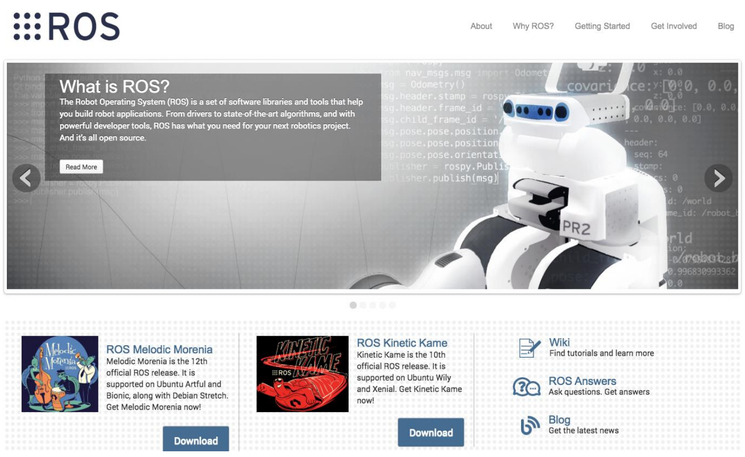
Intelligent Robotics Lab

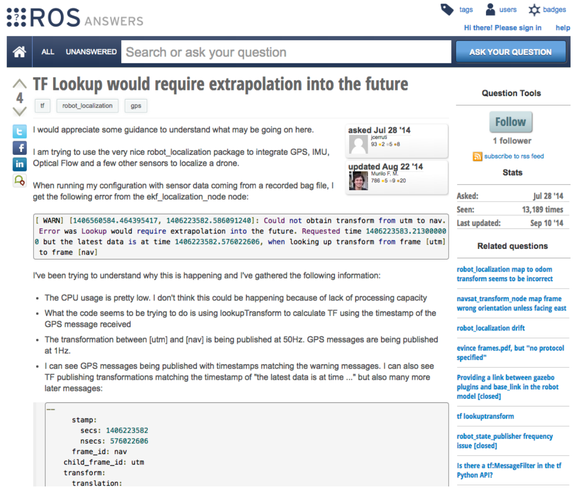
ROS Resources

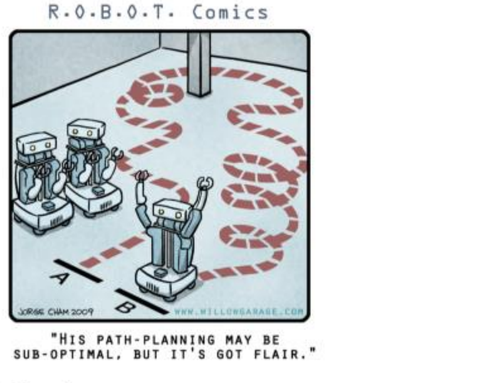
- ROS.org [link]
- Distributions [link]
- Package documentation [link][link]
- ROS Wiki [link]
- ROS Answers [link]
- ROS Discurse [link]
- Blog [link]





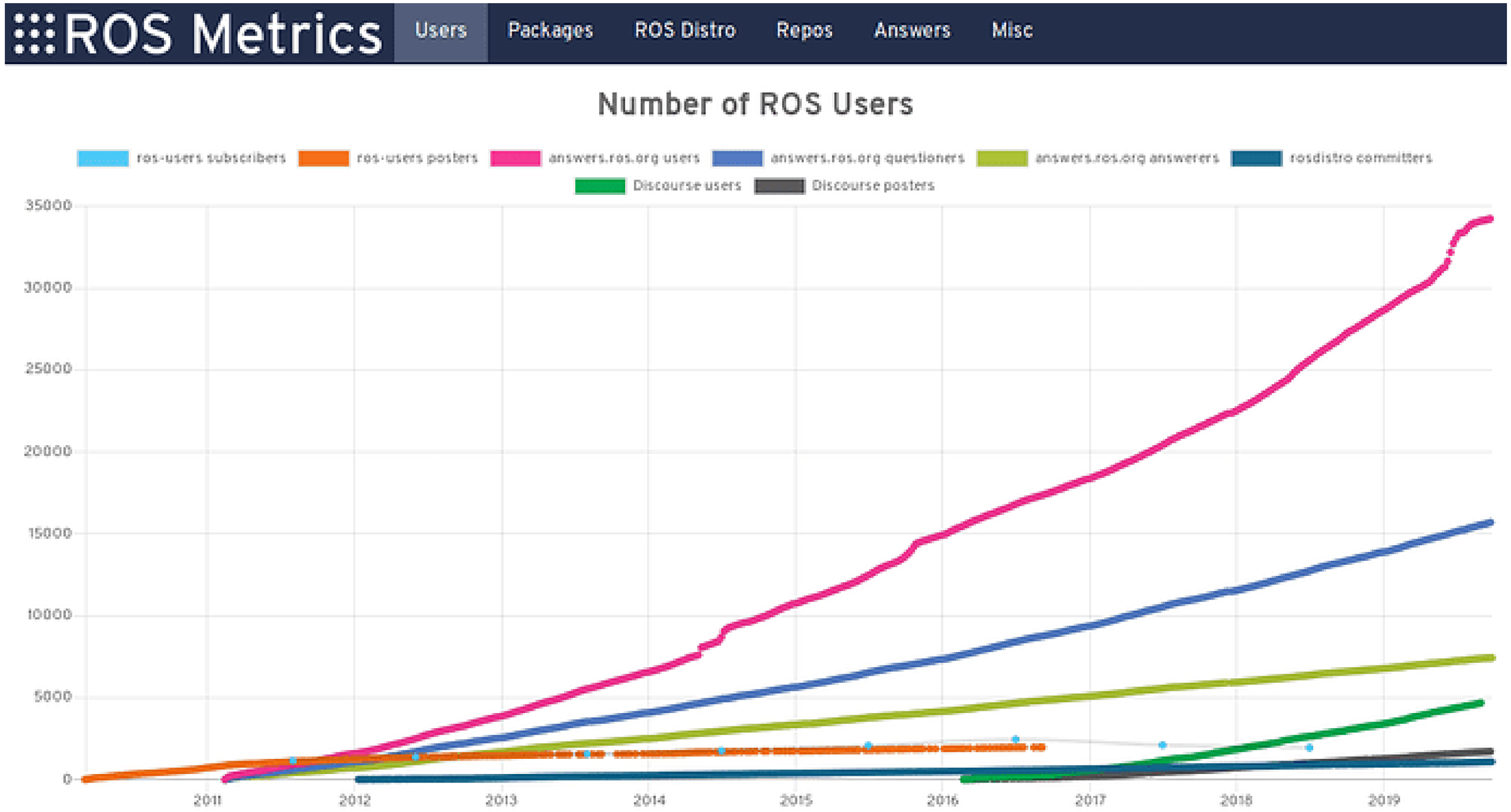






ROS Metrics

https://metrics.ros.org/



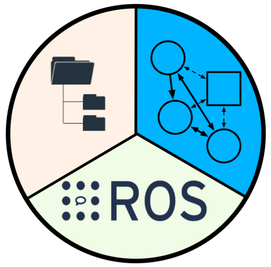
ROS Robots

https://robots.ros.org/

ROS

CELEBRATING TEN YEARS

The Computation Graph



- A robot's software looks like during its execution.
- A Computation Graph contains ROS2 nodes that communicate with each other so that the robot can carry out some tasks.
- The logic of the application is in the nodes, as the primary elements of execution in ROS2.
- Communication mechanisms:
  - **Publication/Subscription**: Asynchronous N:M
  - **Services**: Synchronous 1:1
  - **Actions**: Asynchronous 1:1

Intelligent Robotics Lab

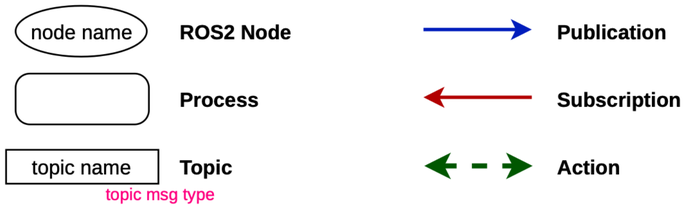
CC BY SA

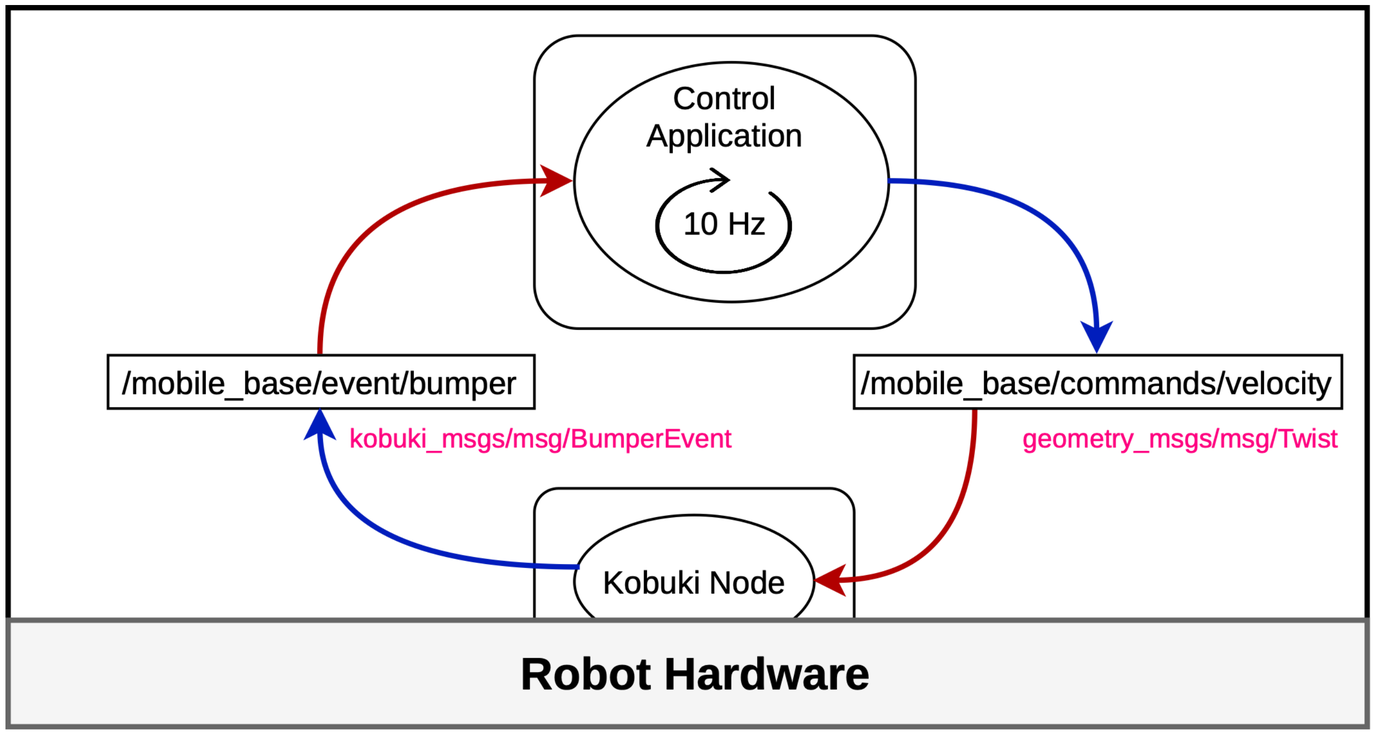
The Computation Graph

- A robot's software looks like during its execution.
- A Computation Graph contains ROS2 nodes that communicate with each other so that the robot can carry out some tasks.
- The logic of the application is in the nodes, as the primary elements of execution in ROS2.

  - Communication mechanisms:
    - **Publication/Subscription**: Asynchronous N:M
    - **Services**: Synchronous 1:1
    - **Actions**: Asynchronous 1:1

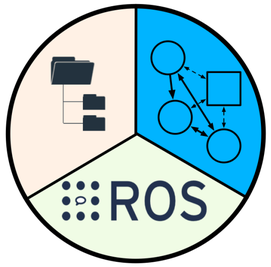
  - Execution model
    - **Iterative**
    - **Event-Oriented**





The Computation Graph

About names



- Resources in ROS2 follow a name convention
- The name of a resource depends on:
  - The type on name: relative, absolute or private
  - The node name
  - The namespace

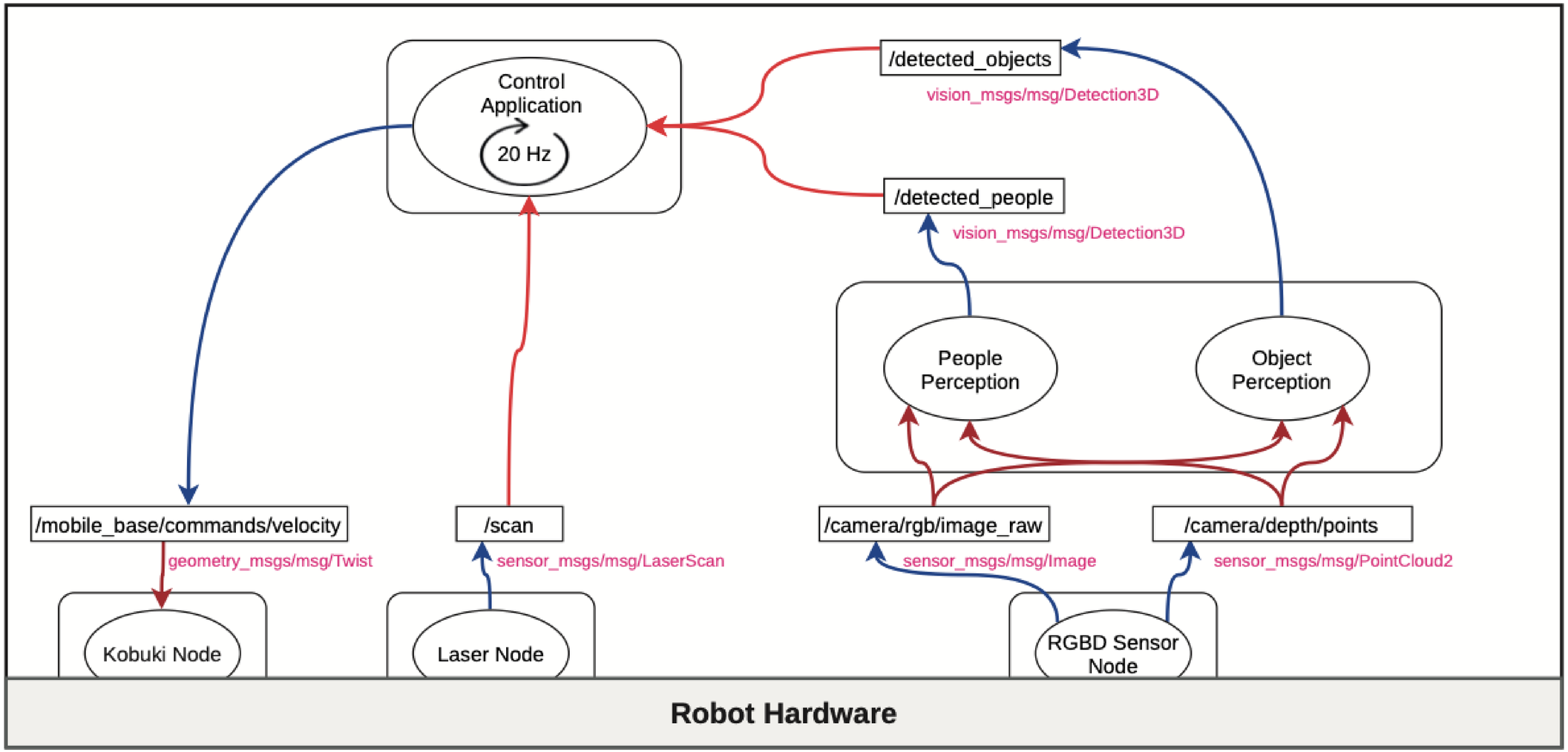
name	Result: (node: my_node / ns: none)	Result: (node: my_node / ns: my_ns)
my_topic	/my_topic	/my_ns/my_topic
/my_topic	/my_topic	/my_topic
~my_topic	/my_node/my_topic	/my_ns/my_node/my_topic

Intelligent Robotics Lab

CC BY SA

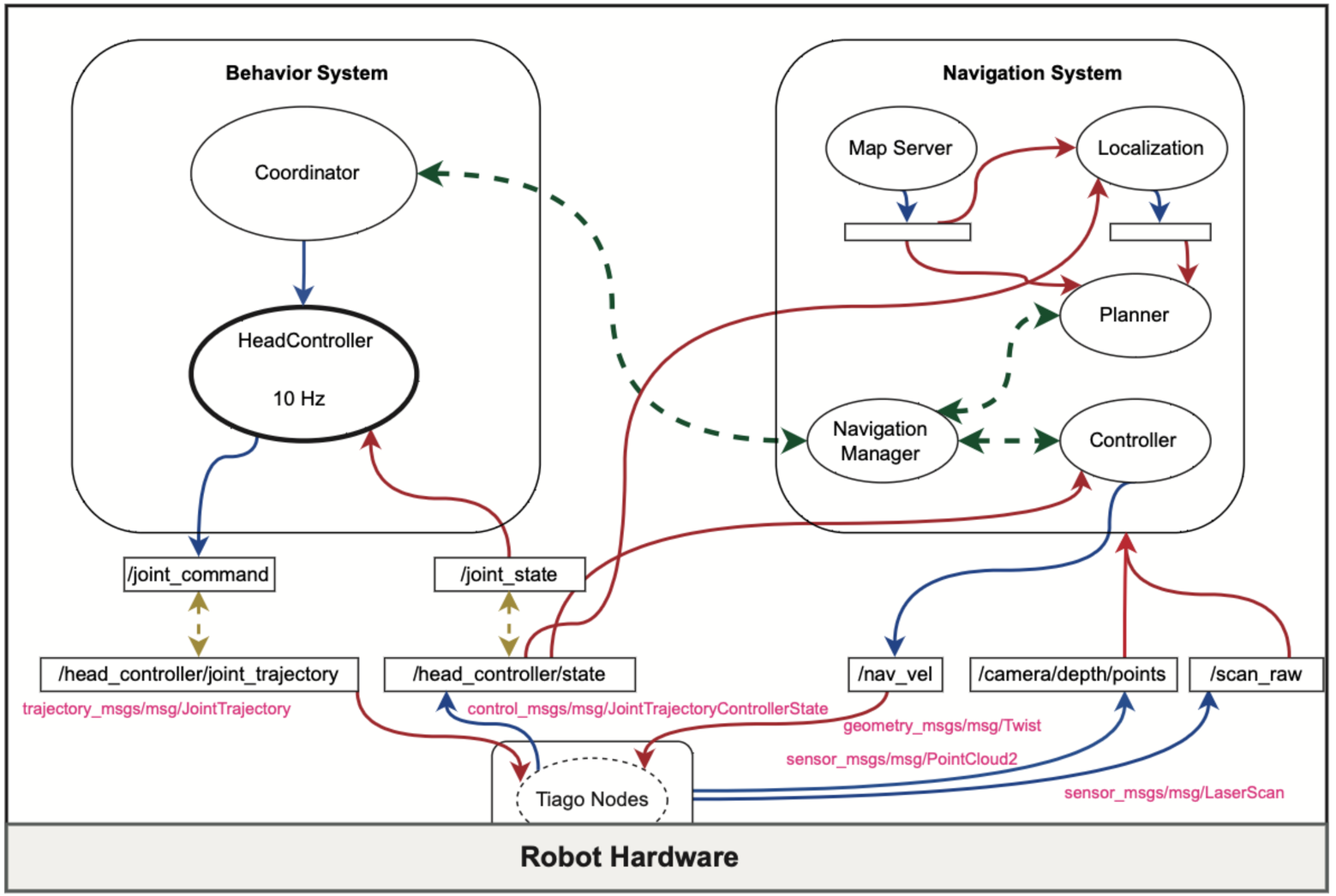
The Computation Graph

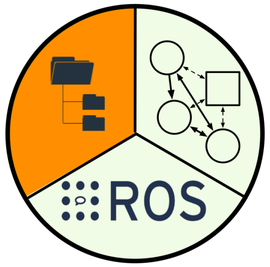
Examples



The Computation Graph

Examples



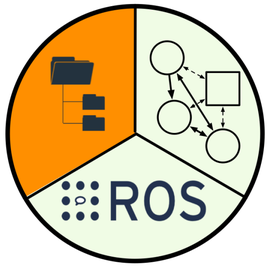


The Workspace

- Approaches ROS2 software from a static point of view.
- Where the ROS2 software is installed, organized, and all the tools and processes that allow us to launch a computing graph.
- This includes the build system and node startup tools.
- Elements:
  - **Package**:
    - It is the minimum functional set of software.
    - Contains executables, libraries, or message definitions with a common purpose.
  - **Workspace**:
    - A directory that contains packages.
    - Activable to be available to use.
- ***Underlay*** y ***overlay***

Intelligent Robotics Lab

CC BY SA



The Workspace

Set up the user workspace

- Build the workspace

```
$ cd ~/ikerlan_ws
$ colcon build –simlink-install
```

- `Build`, `install`, and `log`
- Activate the user workspace

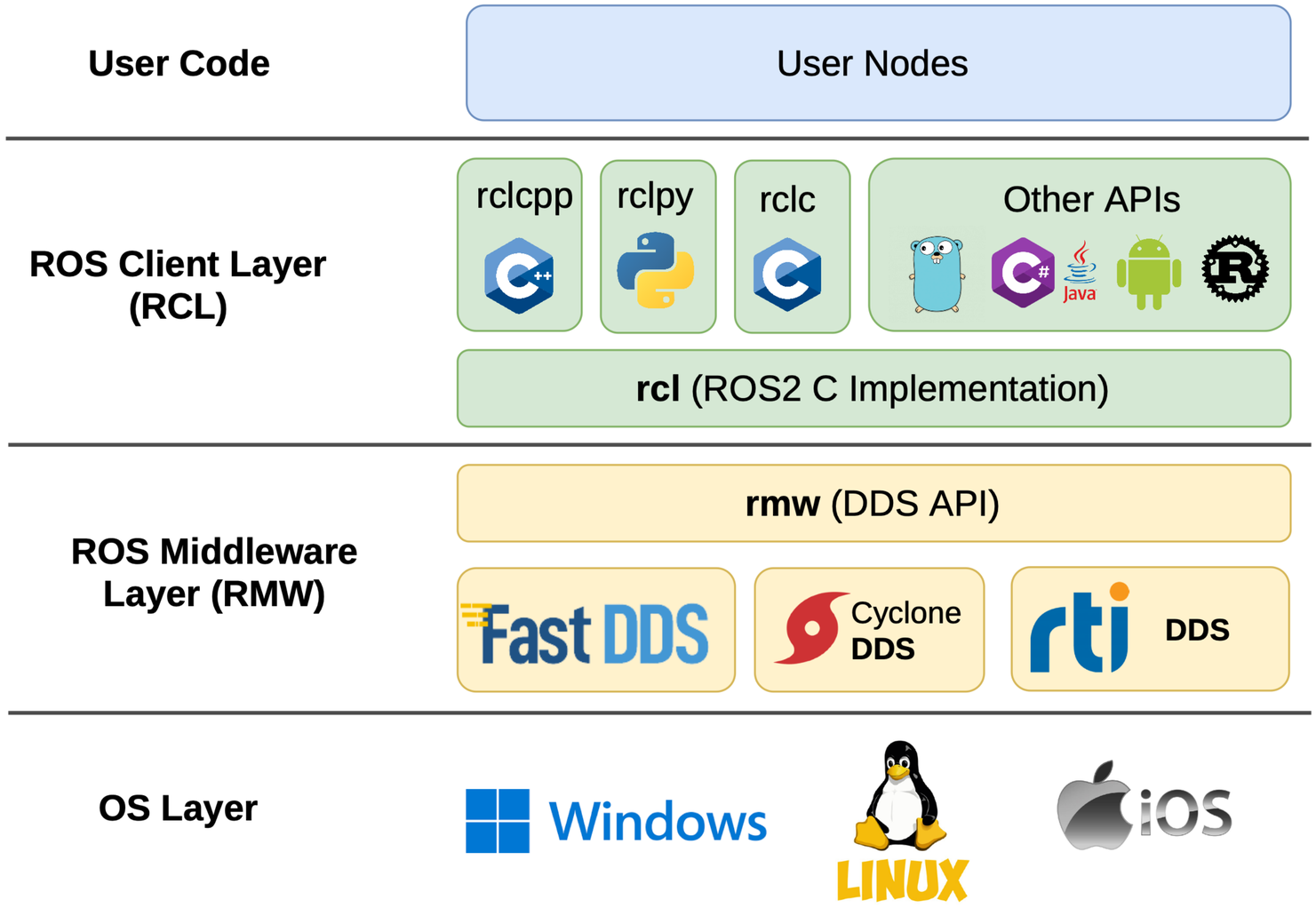
```
$ source ~/ikerlan_ws/install/setup.bash
```

```
$ echo “source ~/ikerlan_ws/install/setup.bash” >> ~/.bashrc
```

Intelligent Robotics Lab

CC BY SA

ROS2 Design


User Code
User Nodes
ROS Client Layer (RCL)
rclcpp
rclpy
rclc
Other APIs
Java
rcl (ROS2 C Implementation)
ROS Middleware Layer (RMW)
rmw (DDS API)
Fast DDS
Cyclone DDS
rti DDS
OS Layer
Windows
LINUX
iOS