CADi

ROS as a Development Platform

ROS 101

Robot
Operating
System

Rolando Bautista Montesano
rolando.bautista@tec.mx

July 21, 2020

Robot Operating System

ROS

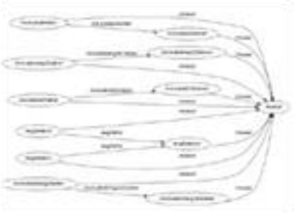
- Framework for writing robot software.
- Reuse software as possible.
- Set of tools, libraries and conventions to make our life easy.
- Simplify the implementation of robust and general-purpose software.

Mec → lab ← Power

VC

ML

=



+

+

+

Plumbing

Tools

Capabilities

Ecosystem

Pag hewr libs

RVIZ RQT ...

A Distributed, Modular Design

- Use ROS as needed!

S1
A1
dis 2

- Choose what you need.
- Developers around the globe.

- Who uses it and why should I use it??

TUM

Carnegie Mellon University

UNIVERSITA DEGLI STUDI DI TORINO

COLUMBIA UNIVERSITY

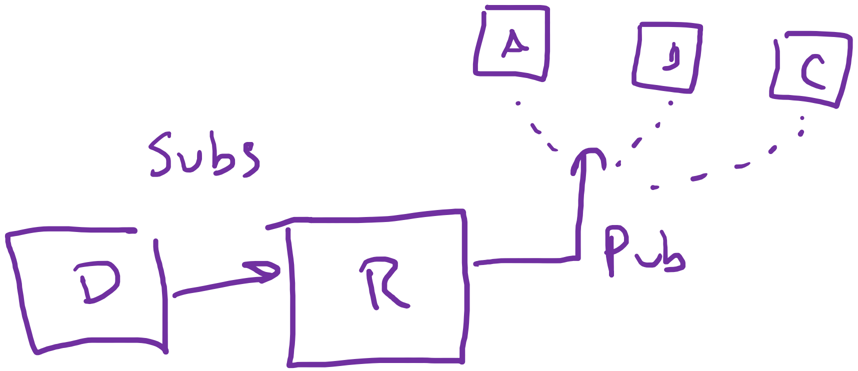
LEGES SINE MORIBUS VANAE

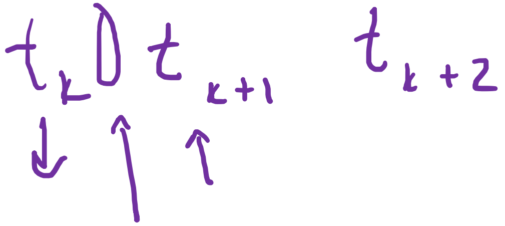
MIT

Core components

3

- Communications infrastructure
  - Support for Python and C++ code.
- Message Passing
  - Publish/subscribe mechanism
- Recording and Playback of Messages
  - Use data as needed on different systems
- Remote Procedure Calls
  - Select who should sent/receive information as well as what and when.
- Distributed Parameter System





Robot-Specific Features

Libre de plataforma

Ix,yz Gx yz raw gray grb arr

- Standard Robot Messages ✓ Q
  - Data coming from IMUs, cameras, lasers
  - Geometry concepts like poses, transforms and vectors ✓
  - Navigation data as odometry, paths and maps ✓
- Preemptable Remote Procedure Calls
  - Check the process of how an action is being performed
- Pose Estimation, Localization, and Navigation ✓
  - Integrated algorithms.

Tools

- Command-Line Tools
  - Powerful set of instructions that can launch nodes, check topics or services.
- RVIZ
  - Most popular tool
  - General purpose 3D visualization of sensor data

imus
pc
lidar
cámaras
paths

- RQT
  - QT based framework to create GUIs
  - rqt_graph - Check connection between nodes.
  - rqt_plot – Monitor data coming from diverse sensors.

2

Integration with other libraries

GAZEBO

cam
lidar
1D+

pointcloudlibrary

ROS industrial

Py C++

OpenCV

MoveIt

ROS Versions

ROS 1

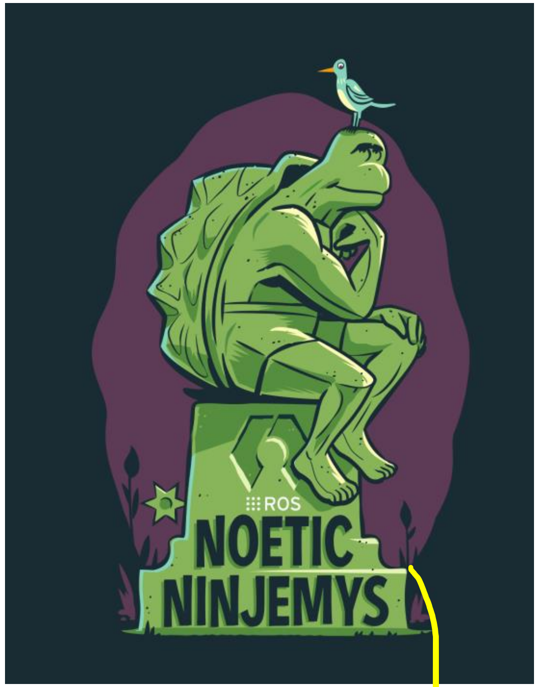
v 2014

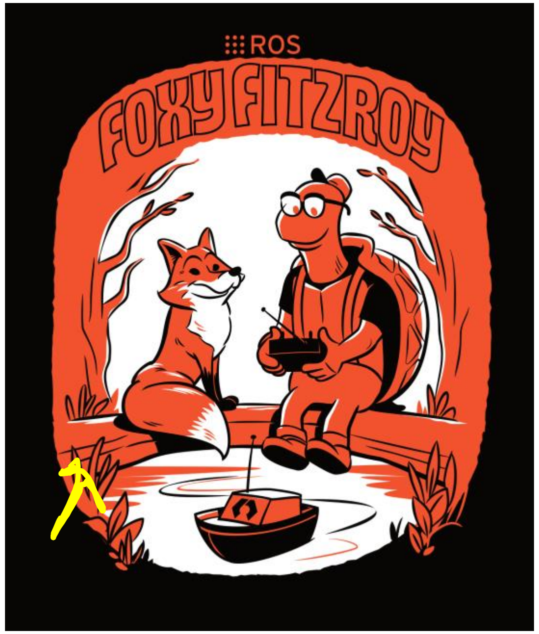
2018

2020

ROS 2

2020





EOL

ubuntu 16.04.2
Xenial Xerus

Ubuntu 18.04 LTS
(Bionic Beaver)

What's new in
UBUNTU 20.04 LTS
(FOCAL FOSSA)

Linux

Windows 10

Version comparison

ROS 1

- Linux
- Python 2.7-based (EOL) with Python 3.0 capabilities
- C++ and LISP enabled

2016

ROS 2

3,8

- Linux, Windows, macOS
- Python 3 and C++ ?
- Industry oriented

$ $

ROS
RL + FIS

amazon
Apex.AI
BOSCH Invented for life
ubuntu
EPROSIMA The Middleware Experts
intel
LG Electronics
Microsoft
open robotics
PICKNIK
ROBOTIS
ROVER ROBOTICS
SAMSUNG
DEVCOM GROUND VEHICLE SYSTEMS CENTER
Tier IV Intelligent Vehicle
TOYOTA RESEARCH INSTITUTE

How does ROS work?

①

- Filesystem level – resources on disk

  instalas   Odom
  String

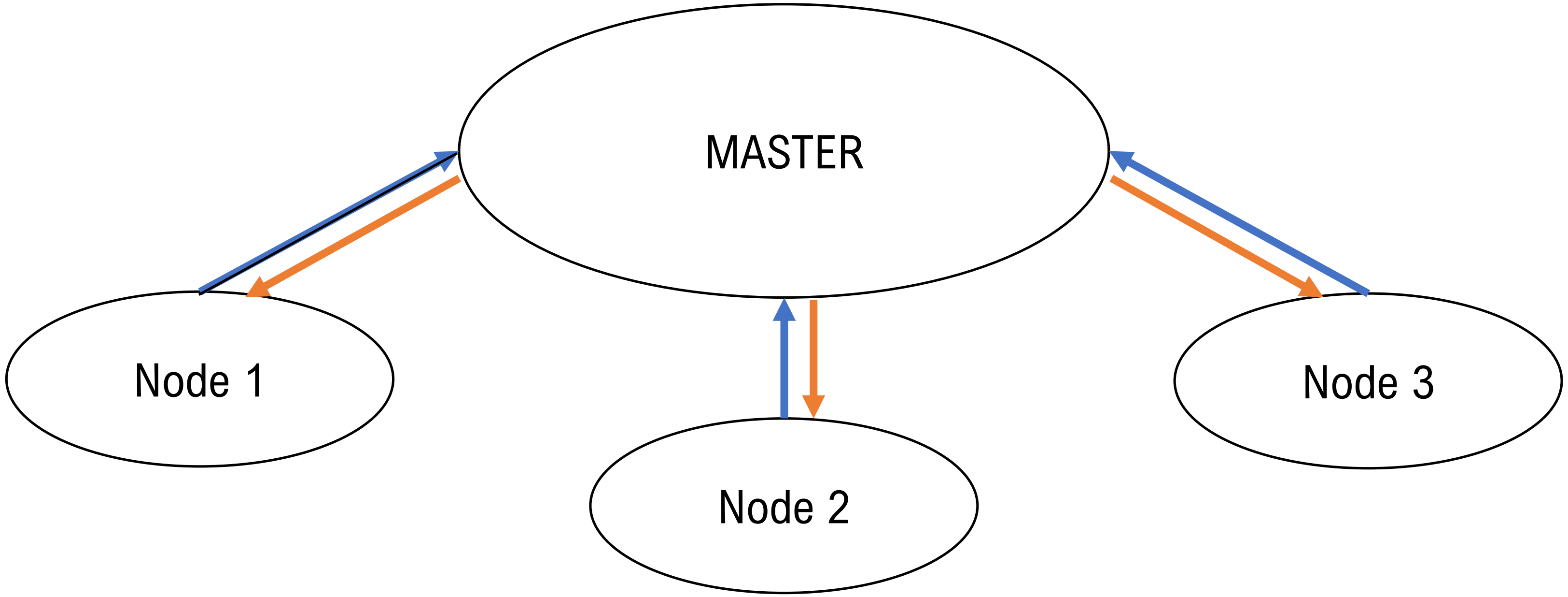
  - Packages, metapackages, manifests, repositories, message types
- Computation Graph level – P2P network that process data together
  - Nodes, master, parameter server, messages, topics, services and bags
- Community level – resources that enable community exchange
  - Distributions, repositories, Wiki, …

Computation Graph level

C struct {
double temp
int id
bool Fault
arr[] sensp

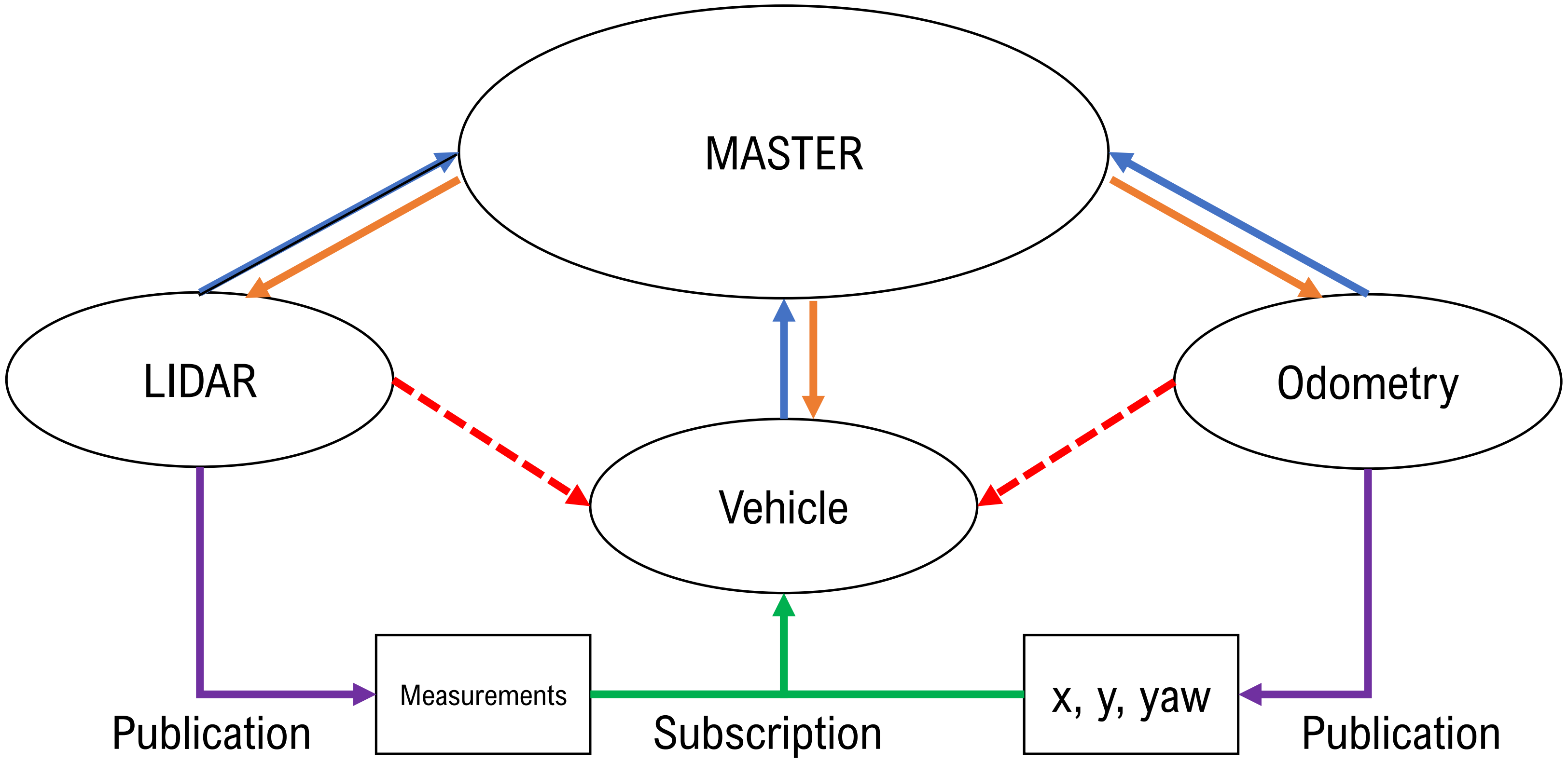
- Nodes – Processes that perform computation, i.e. LRF, motors, localization. Written on rospy or roscpp.
- Master – Provides name registration and lookup for the graph.
- Parameter server – Assigns labels and stores data. Part of the master.
- Messages – Data structure comprised by fields. ~ C struct
- Topics – Mean of transport of the messages. Data can be acquired or sent through it. One node can publish/subscribe to several topics.

Process – 1 Registration information

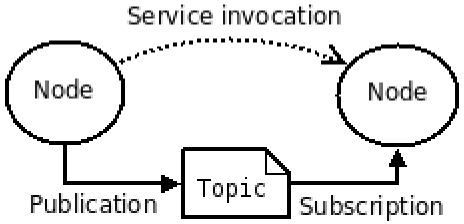


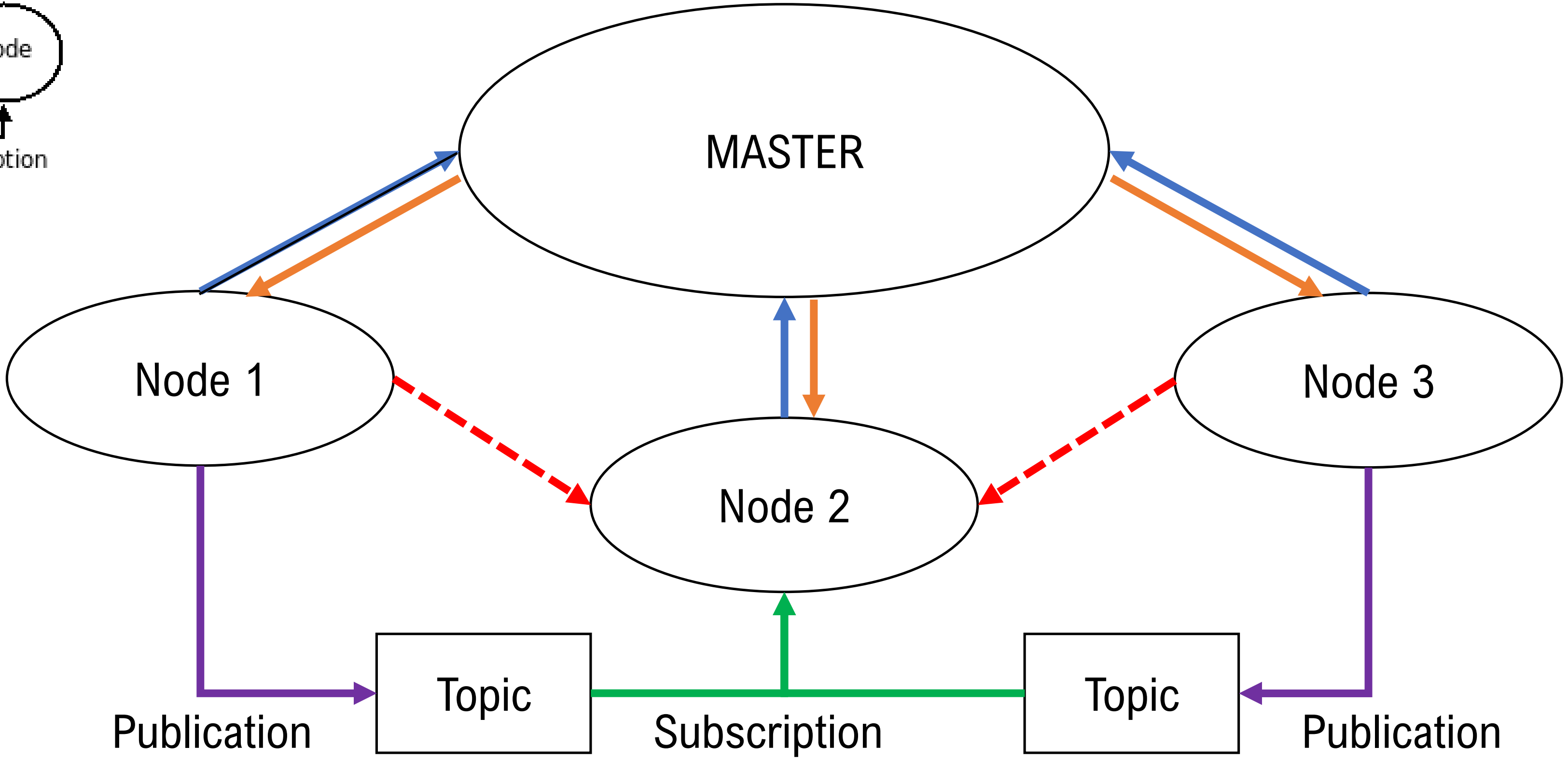
- Information from other nodes →
- Topics and services registration | callbacks →

Process – 2 Example



Process – 2 Data transmission





Practical example

- Create a publisher
- Subscribe to it

Workspace

- Catkin – build system for ROS
- CMake + Python scripts

- It generates targets from raw source to be used by an end user
- Portable