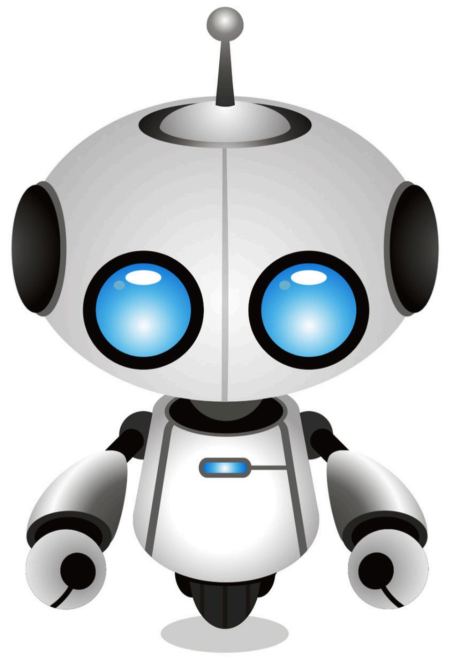
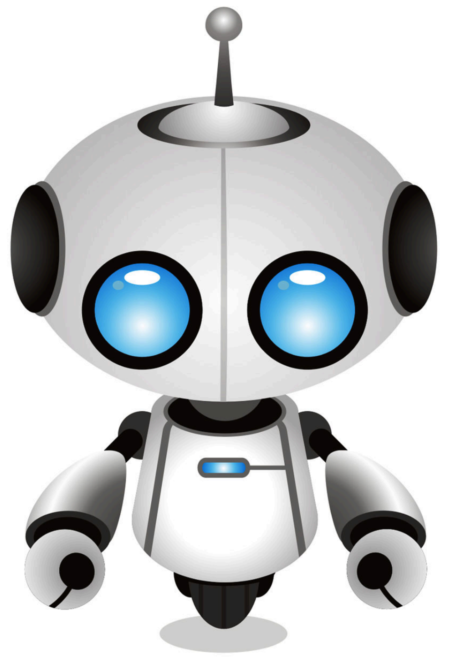
ROS2

Robot Operating System Version 2





Eric Weikl & Dr. Martin Idel

BTD11, 2018-05-18

TNG TECHNOLOGY CONSULTING

About Us

Eric Weikl
Associate Partner
ROS User

Dr. Martin Idel
Software Consultant
ROS2 Contributor

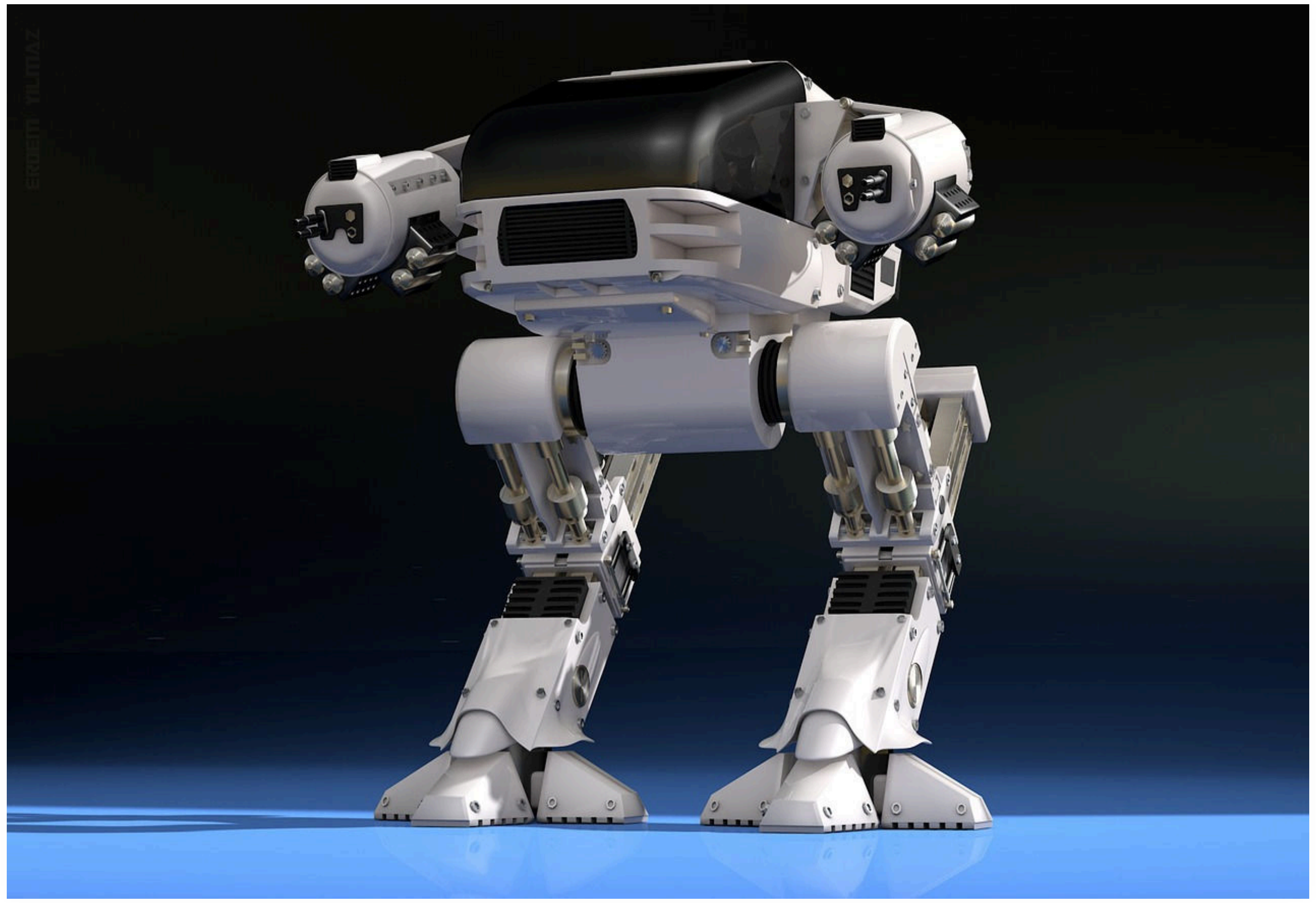
TNG TECHNOLOGY CONSULTING

We solve hard IT problems.

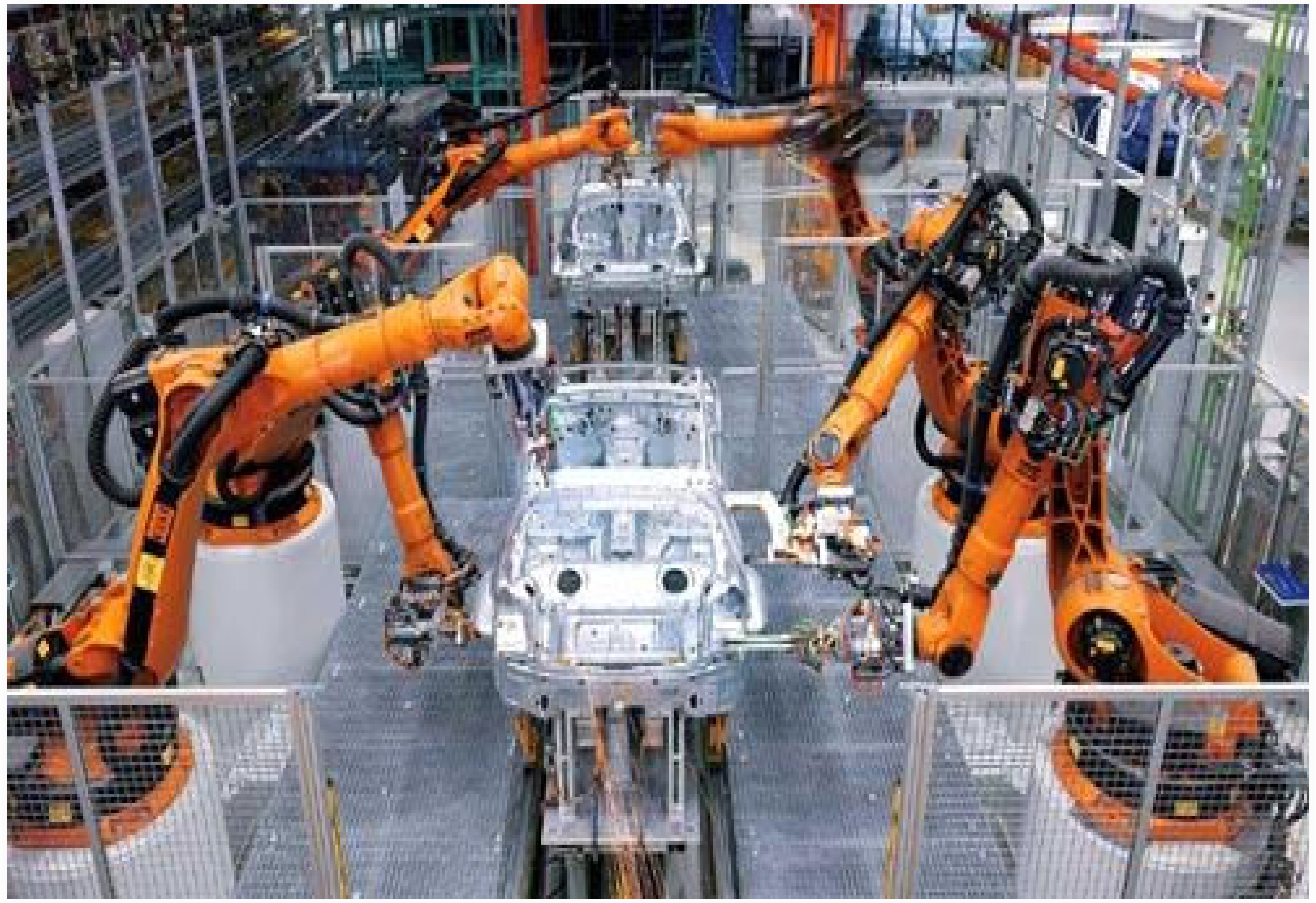
Overview

The Robot Invasion Is Here.

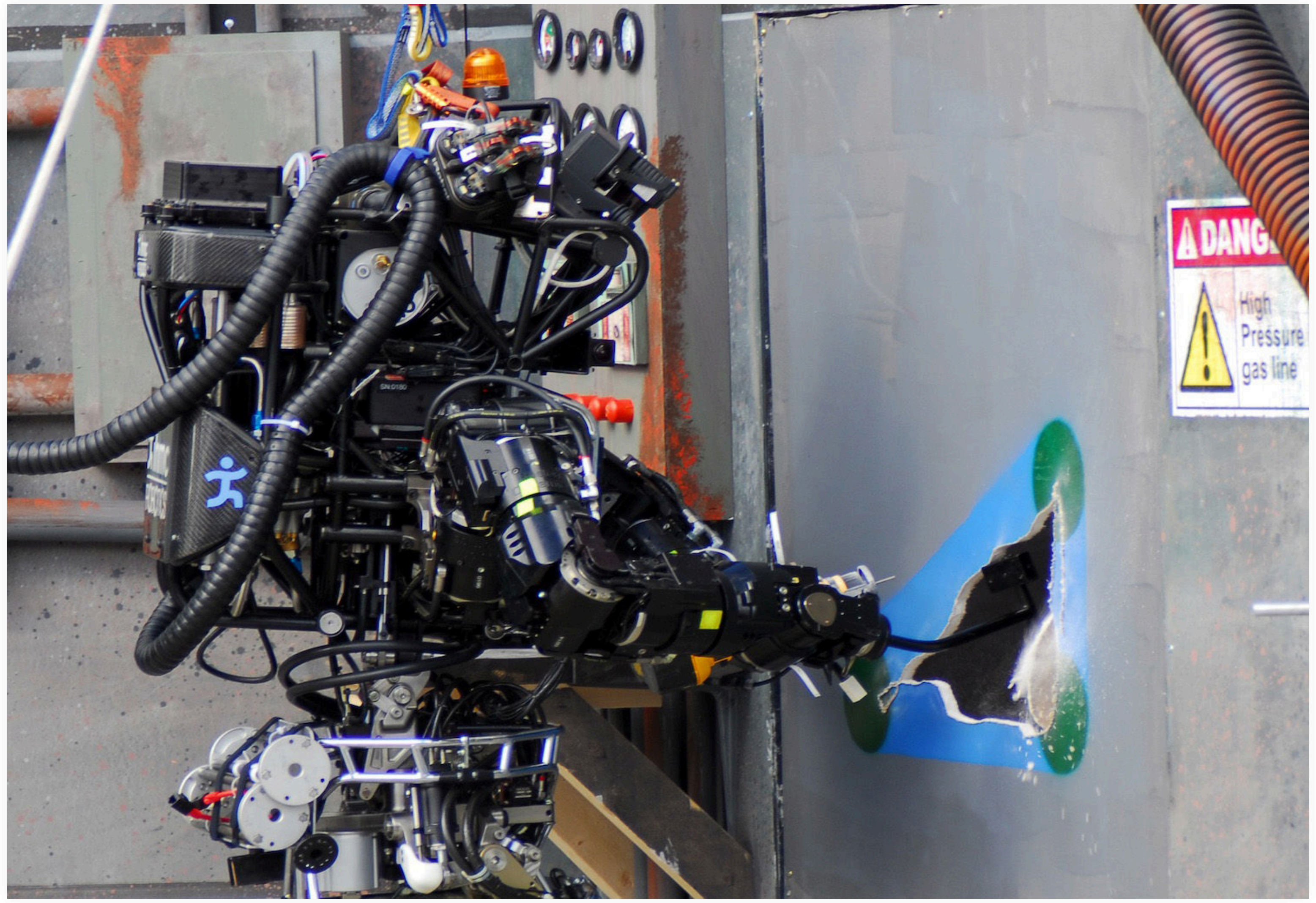
The Robot Invasion Is Here.



The Robot Invasion Is Here.



The Robot Invasion Is Here.



The Robot Invasion Is Here.

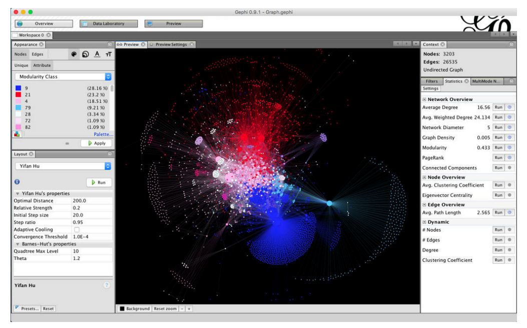
The Robot Invasion Is Here.

The Robot Invasion Is Here.

Why ROS?

ROS

Integration is hard



```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <arpa/inet.h>

void serveur1(portServ ports)
{
    int sockServ1, sockServ2, sockClient;
    struct sockaddr_in monAddr, addrClient, addrServ2;
    socklen_t lenAddrClient;

    if ((sockServ1 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
    }
    if ((sockServ2 = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
    perror("Erreur socket");
    exit(1);
    }

    bzero(&monAddr, sizeof(monAddr));
    monAddr.sin_family = AF_INET;
    monAddr.sin_port = htons(ports.port1);
    monAddr.sin_addr.s_addr = INADDR_ANY;
    bzero(&addrServ2, sizeof(addrServ2));
```

Overview

What is ROS?

RINOS

(ROS Is Not an Operating System)

ROS is...

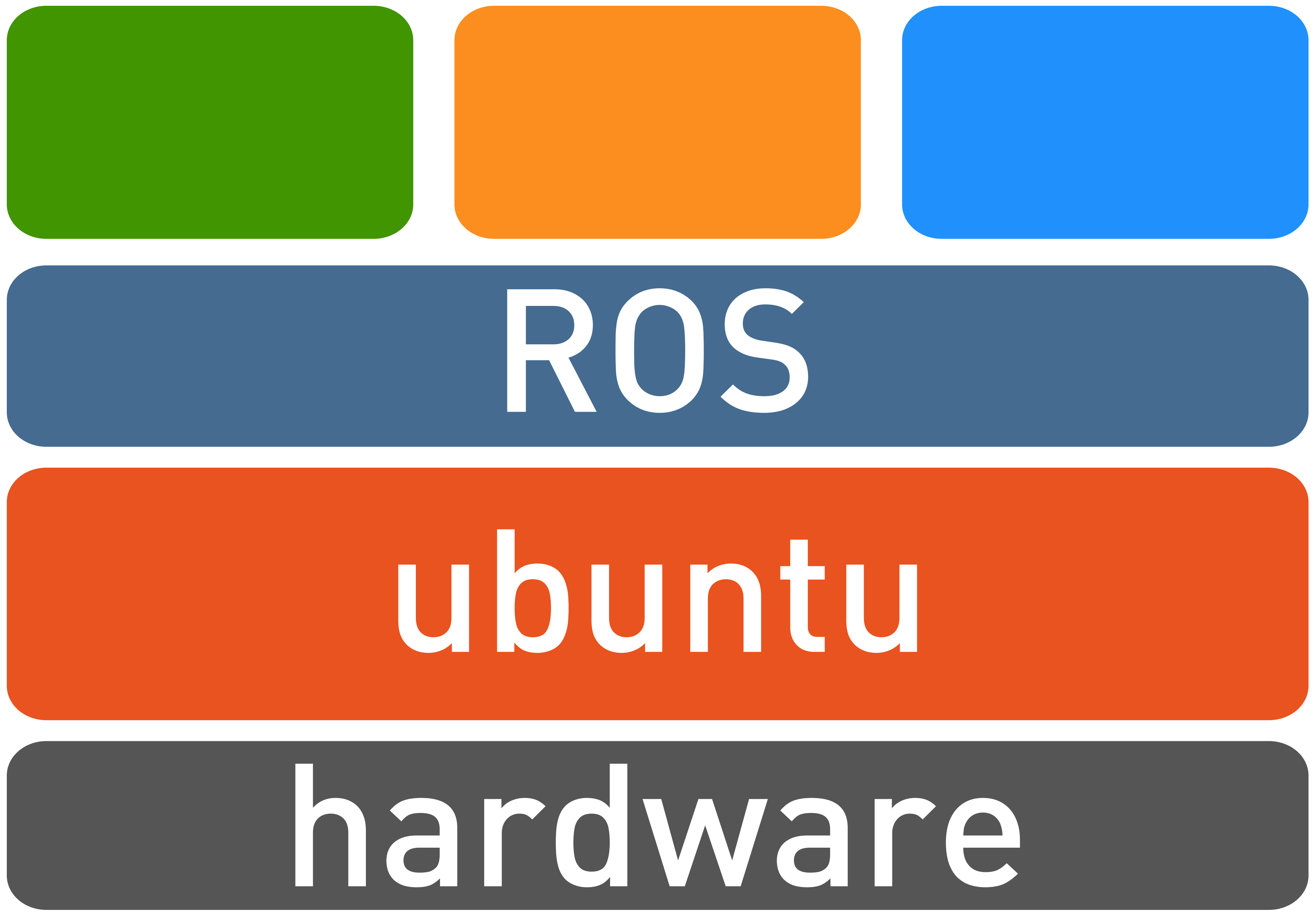
Communication System

+ Framework & Tools

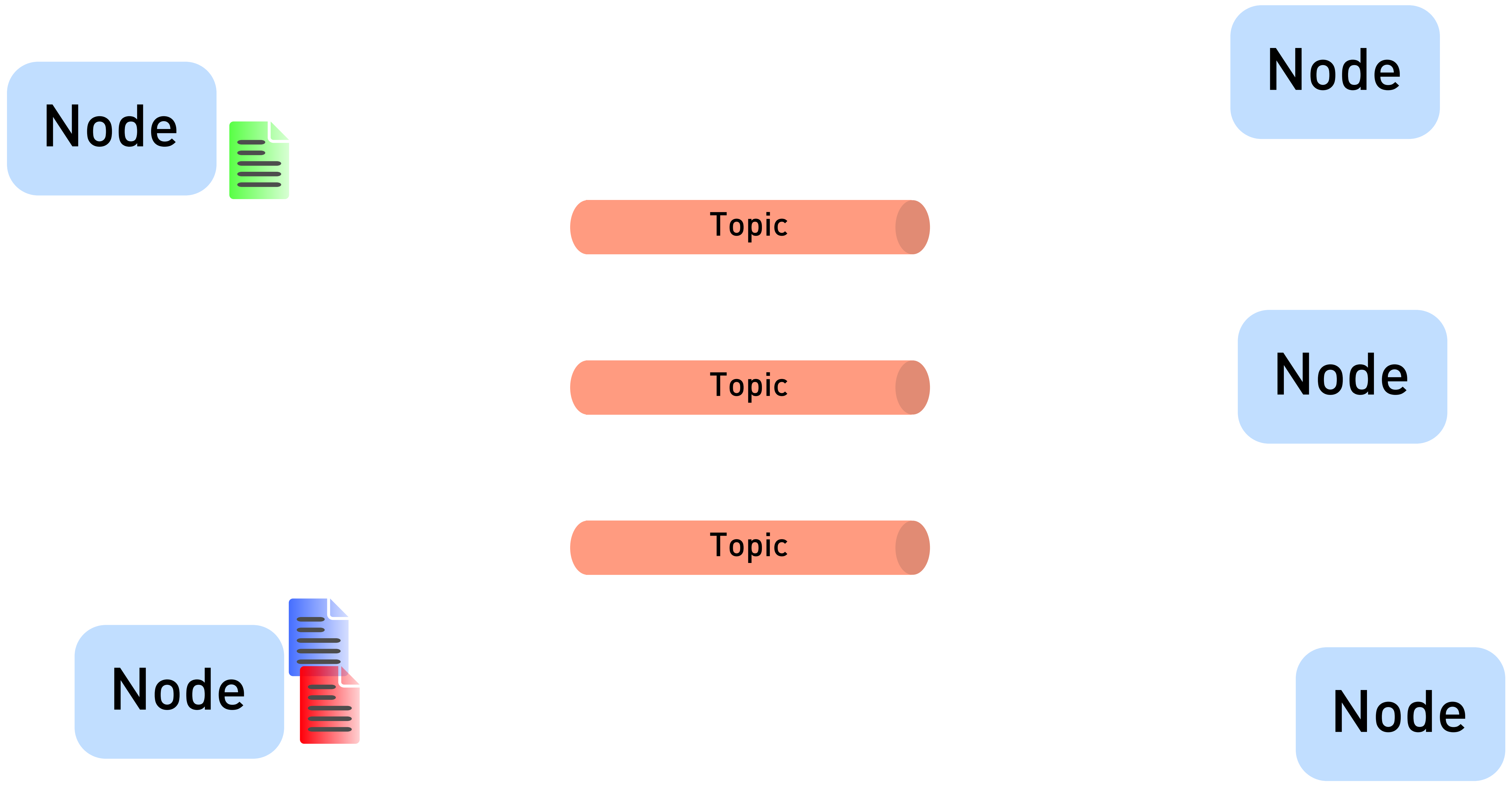
+ Ecosystem

”Linux of Robotics”

High-Level View



Communication System



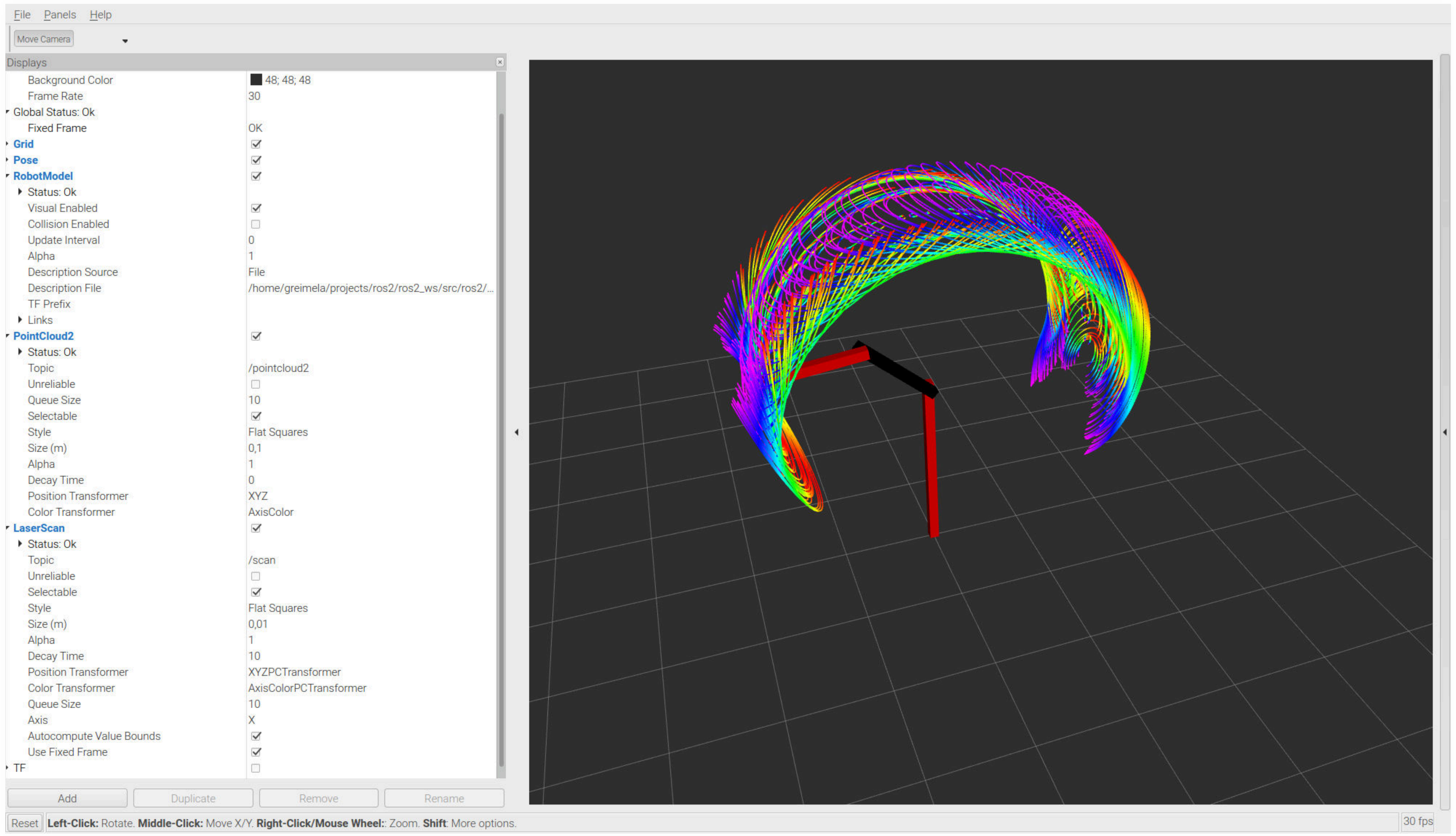
Framework & Tools

- Build system & dependency management
- Visualization
- Persistence

Build system & dependency management

- catkin / ament (based on cmake)
- colcon as a command line build tool
- Binary & source-based dependency management
- Message definition sharing

Visualization



Background Color	■ 48; 48; 48
Frame Rate	30
Global Status: Ok	
Fixed Frame	OK
Grid	✓
Pose	✓
RobotModel	✓
Status: Ok	
Visual Enabled	✓
Collision Enabled	☐
Update Interval	0
Alpha	1
Description Source	File
Description File	/home/greimela/projects/ros2/ros2_ws/src/ros2/…
TF Prefix	
Links	
PointCloud2	✓
Status: Ok	
Topic	/pointcloud2
Unreliable	☐
Queue Size	10
Selectable	✓
Style	Flat Squares
Size (m)	0,1
Alpha	1
Decay Time	0
Position Transformer	XYZ
Color Transformer	AxisColor
LaserScan	✓
Status: Ok	
Topic	/scan
Unreliable	☐
Selectable	✓
Style	Flat Squares
Size (m)	0,01
Alpha	1
Decay Time	10
Position Transformer	XYZPCTransformer
Color Transformer	AxisColorPCTransformer
Queue Size	10
Axis	X
Autocompute Value Bounds	✓
Use Fixed Frame	✓
TF	☐

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift**: More options.

Persistence

- Recording & replaying of messages
- filtering, splitting, joining
- preserves order and timing
- provides introspection capabilities

Ecosystem

- Open Source Community
- Various language bindings
- Drivers (lidar, camera, etc.)
- Libraries (e.g. Pointcloud, Google Cartographer)
- Vendor-supplied bridges to proprietary solutions
- Open Source synergies, e.g. simulation (Gazebo)

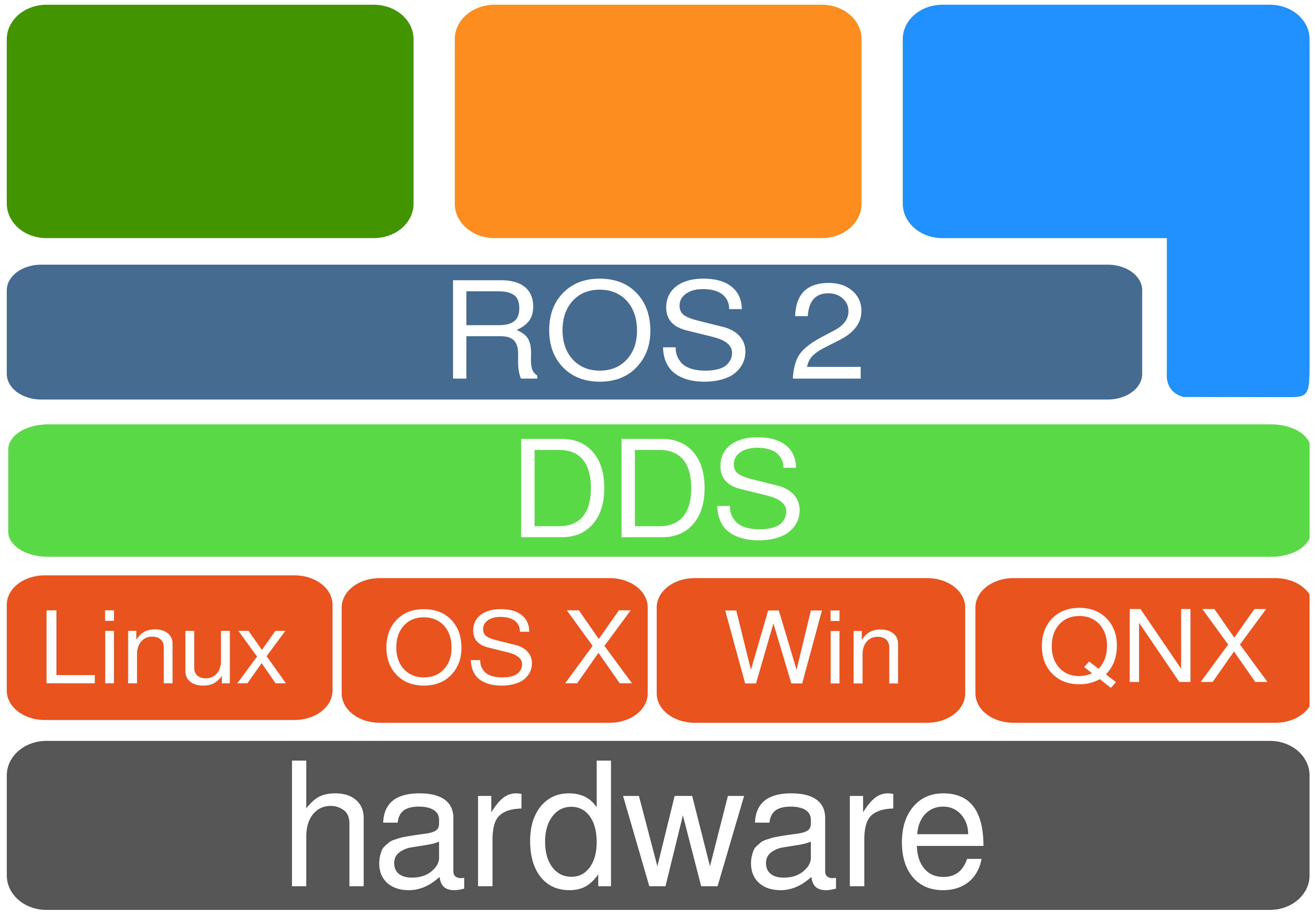
Overview

What’s New in ROS2?

Production Focus with DDS

- Near real-time
- Reduced resource requirements
- Improved network resiliency
- Lifecycle management for ROS nodes
- P2P, no single point of failure

Extended Platform Support



ROS2 Status

ROS2 Version 1.0 Released

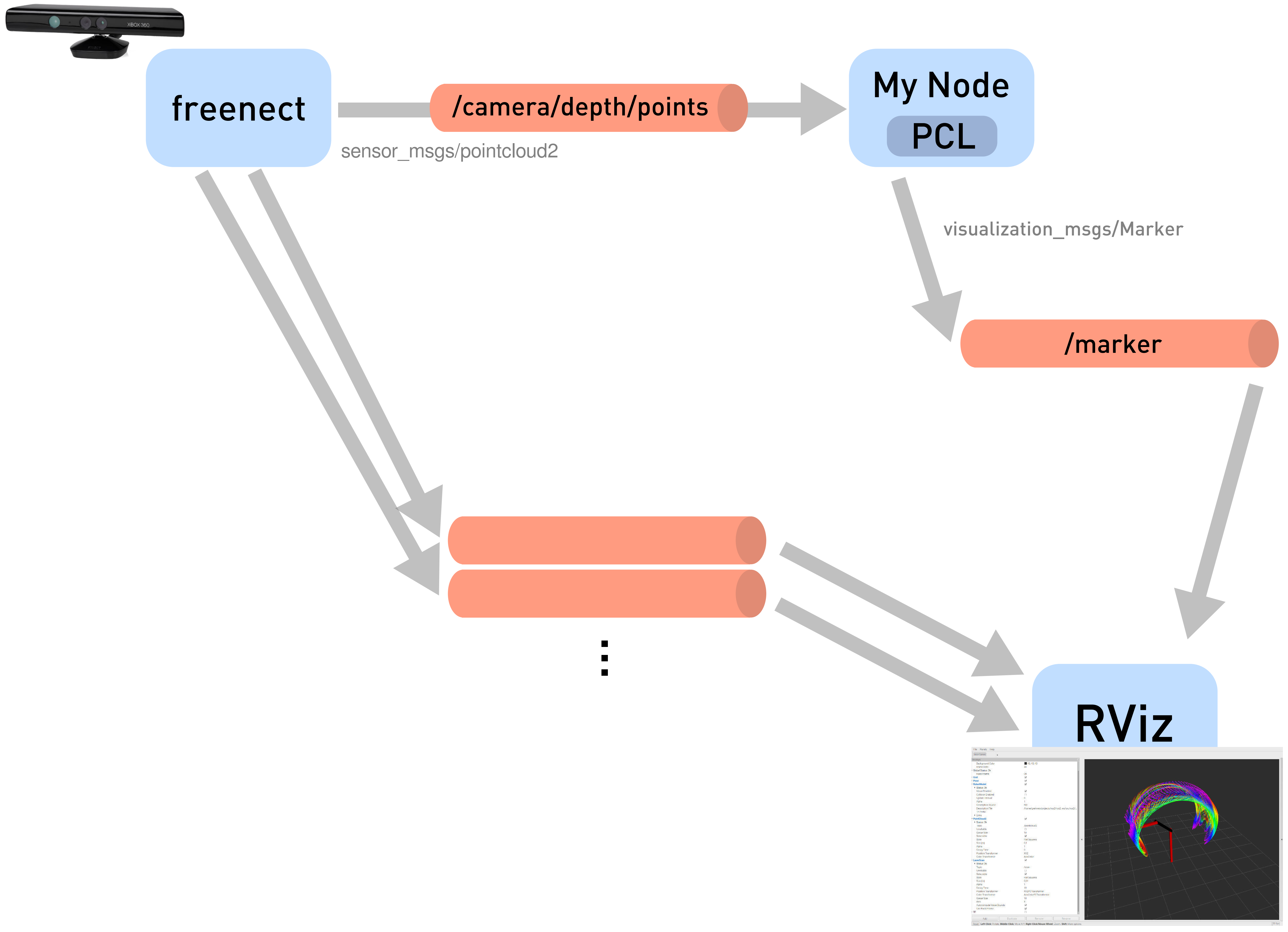
December 2017

Ready for Prime Time?

- Message system in place
- No feature parity with ROS1 (RViz, rosbag). ROS Bridge available
- Documentation is a work in progress
- Not all examples and drivers are ported

Overview

ROS in Action


freenect
/camera/depth/points
sensor_msgs/pointcloud2
My Node
PCL
visualization_msgs/Marker
/marker
⋮
RViz

main.cpp

```cpp
#include <cstdint>
#include <memory>
#include <ros/ros.h>
#include "./pointcloud_segmentation_node.cpp"

int main(int argc, char** argv)
{
  ros::init(argc, argv, "sub_pcl");
  ros::NodeHandle nh;

  auto subscriber = std::make_unique<PointCloudSubscriber>(nh);

  while(nh.ok()) {
    ros::spin();
  }
  return 0;
}
```

point_cloud_subscriber.cpp

```cpp
class PointCloudSubscriber
{
public:
  PointCloudSubscriber(ros::NodeHandle nh)
  : publisher(std::make_unique<MarkerPublisher>(nh)),
  subscriber(nh.subscribe<pcl::PointCloud<pcl::PointXYZ>>(
    "/camera/depth/points", 1, &PointCloudSubscriber::callback, this))
  {}

  void callback(const pcl::PointCloud::ConstPtr & msg) {
    BoundingBox box = pointcloud_segmentation(msg);
    publisher->publishBoundingBox(*box.pose, *box.dimensions);
  }

private:
  {...}

  std::unique_ptr<MarkerPublisher> publisher;
  ros::Subscriber subscriber;
};
```

sensor_msgs/PointCloud2 message

```
std_msgs/Header header
uint32 height
uint32 width
sensor_msgs/PointField[] fields
bool is_bigendian
uint32 point_step
uint32 row_step
uint8[] data
bool is_dense
```

point_cloud_subscriber.cpp

```
pcl::PointCloud<pcl::PointXYZ>::Ptr filterPointCloud(
  const pcl::PointCloud<pcl::PointXYZ>::ConstPtr & cloud)
{
  pcl::PointCloud<pcl::PointXYZ>::Ptr point_cloud_without_nan(
    new pcl::PointCloud<pcl::PointXYZ>);
  std::vector<int> indices;
  pcl::removeNaNFromPointCloud(*cloud, *point_cloud_without_nan, indices);

  pcl::PointCloud<pcl::PointXYZ>::Ptr point_cloud_filtered(
    new pcl::PointCloud<pcl::PointXYZ>);
  pcl::PassThrough<pcl::PointXYZ> filter;
  filter.setInputCloud(point_cloud_without_nan);
  filter.setFilterFieldName("z");
  filter.setFilterLimits(0.1, 3);  // 10cm - 3m
  filter.filter(*point_cloud_filtered);

  pcl::PointCloud<pcl::PointXYZ>::Ptr point_cloud(
    new pcl::PointCloud<pcl::PointXYZ>);
  pcl::VoxelGrid<pcl::PointXYZ> voxel_down_sampling;
  voxel_down_sampling.setInputCloud(point_cloud_filtered);
  voxel_down_sampling.setLeafSize(0.01f, 0.01f, 0.01f);
  voxel_down_sampling.filter(*point_cloud);
  return point_cloud;
}
```

point_cloud_subscriber.cpp

```
std::vector<pcl::PointIndices> cluster_extraction(
  const pcl::PointCloud<pcl::PointXYZ>::Ptr & point_cloud)
{
  pcl::search::KdTree<pcl::PointXYZ>::Ptr tree(
    new pcl::search::KdTree<pcl::PointXYZ>);
  tree->setInputCloud(point_cloud);

  pcl::EuclideanClusterExtraction<pcl::PointXYZ> ec;
  ec.setClusterTolerance(0.02); // 2cm
  ec.setMinClusterSize(100);
  ec.setMaxClusterSize(250000);
  ec.setSearchMethod(tree);
  ec.setInputCloud(point_cloud);

  std::vector<pcl::PointIndices> cluster_indices;
  ec.extract(cluster_indices);
  return cluster_indices;
}
```

Overview

Summary

ROS2 is extremely promising

- builds on experience with ROS1
- safety-critical environments
- DDS-based systems

ROS2 is not done yet

Community still focused on ROS1

If you’re a happy ROS1 user

- evaluate this year
- consider implementing next year

If you’re a hardware OEM already supporting ROS

Start prototyping now!

Thank you!

@ericweikl
eric.weikl@tngtech.com
https://www.xing.com/profile/Eric_Weikl

martin.idel@tngtech.com

TNG TECHNOLOGY CONSULTING

Images

- https://commons.wikimedia.org/wiki/File:Industrial-robots.jpg - CC BY-SA 4.0 by "ISAPUT"
- https://en.wikipedia.org/wiki/File:Sanbot_King_Kong.jpg - CC0 1.0 by QIHAN Technology
- https://commons.wikimedia.org/wiki/File:%D0%A0%D0%BE%D0%B1%D0%BE%D1%82_%D0%BF%D1%8B%D0%BB%D0%B5%D1%81%D0%BE%D1% - CC BY-SA 3.0 by "Nohau"
- https://pixabay.com/en/highway-auto-traffic-road-drive-1767106/ - CC0 1.0 by Michael Gaida
- https://pixabay.com/en/robot-machine-technology-science-2178590/ - CC0 1.0 by "Pexels"
- https://commons.wikimedia.org/wiki/File:ROS_cat.png - CC BY-SA 4.0 by "Rribal"
- https://de.wikipedia.org/wiki/Datei:Gephi_0.9.1_Network_Analysis_and_Visualization_Software.png - CC BY-SA 4.0 by "SlvrKy"
- https://de.wikipedia.org/wiki/Datei:Xbox-360-Kinect-Standalone.png - Public Domain by "Evan-Amos"
- https://pixabay.com/en/robot-arm-mechanics-technology-2658318/ - CC0 1.0 by kalhh
- https://www.maxpixel.net/Coins-Juggle-Artists-Finance-Money-Circus-1027844 - CC0 1.0
- https://commons.wikimedia.org/wiki/File:Source_code_in_C.png - CC BY-SA 3.0 by Romainhk
- https://pixabay.com/en/panic-button-panic-fear-button-1067100/ - CC0 1.0 by TheDigitalArtist
- https://pixabay.com/en/cute-cartoon-robot-funny-character-807306/ - CC0 1.0 by bcogwene