

AWS  
re:Invent

O P N 2 0 1

# Change is coming to robotics development: The shift to ROS 2

**Alejandra Olvera-Novack**

AWS Developer Relations  
Amazon Web Services

Let's talk...

I make it a point to answer all my DMs and emails from customers after I give a talk

[quetzal@amazon.com](mailto:quetzal@amazon.com)

@QuetzalliAle

# Agenda

What is ROS? What can it do?

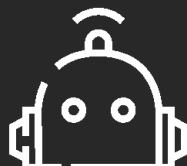
What's the same between ROS 1 and ROS 2?

What's different between ROS 1 and ROS 2?

Who should be porting to ROS 2?

Who is still on ROS 1 and doesn't really need to rush to ROS 2 yet?

Why did Rover Robotics decide it was worth the effort to port to ROS 2?





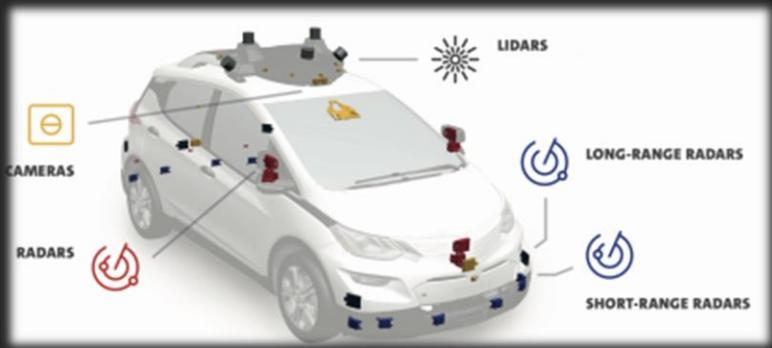
International Space Station



Autonomous walker



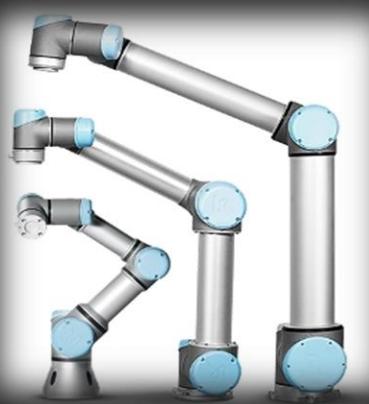
Drones



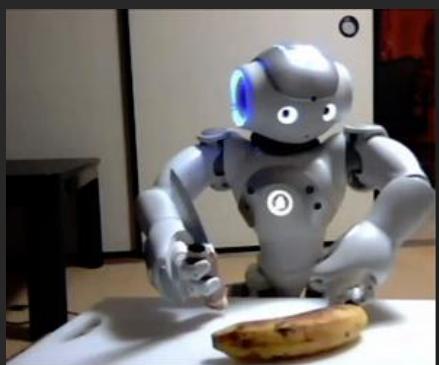
Self-driving vehicles



Water

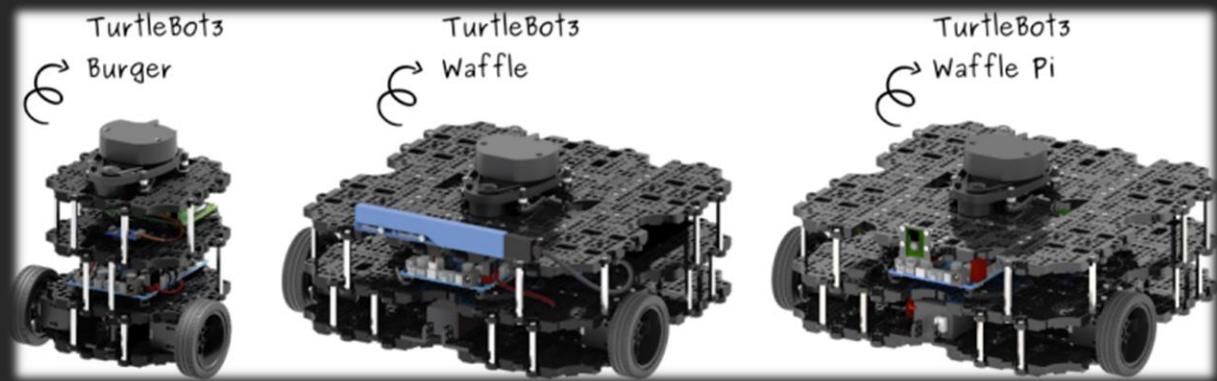


Robotic arms



Home

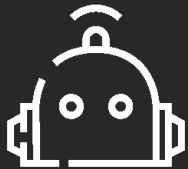
Education



Rover



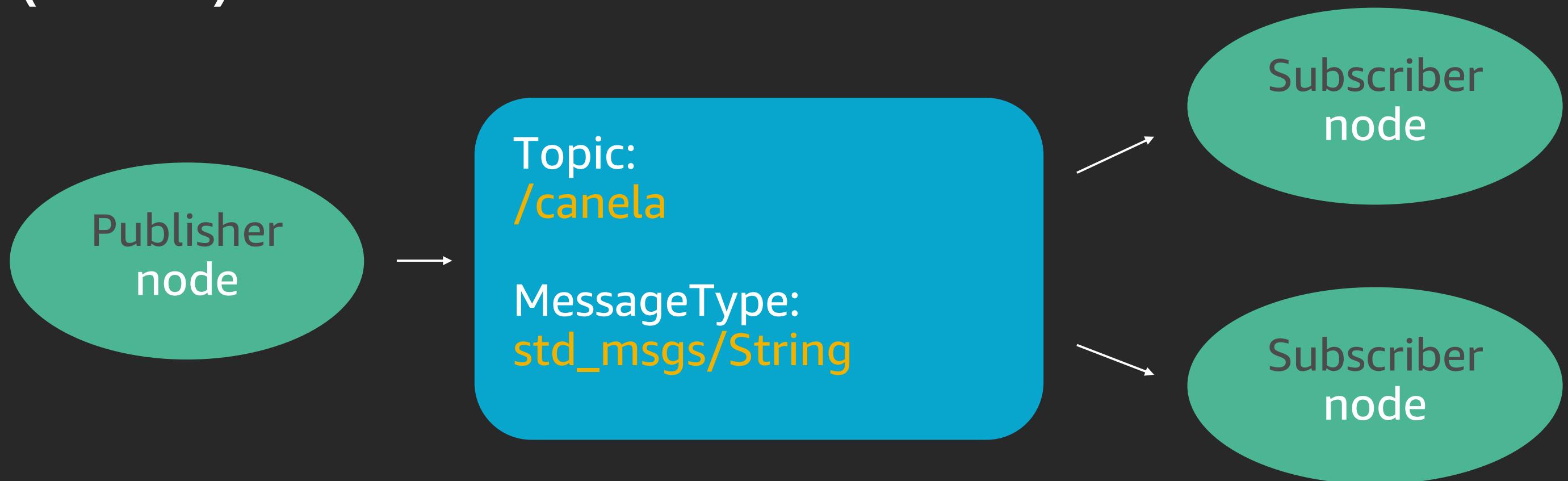
# Why do I care about robots?





# What is ROS? What can it do?

ROS is a **middleware** based on an anonymous **publish/subscribe mechanism** that allows for message passing between different ROS processes (nodes)



# Robot Operating System

ROS

- Many-to-many publish subscribe framework
- Open source, BSD
- Largest ecosystem of packages and tools
- Largest active developer community
- Used in both research and commercial applications

Willow Garage

# Robot Operating System 2

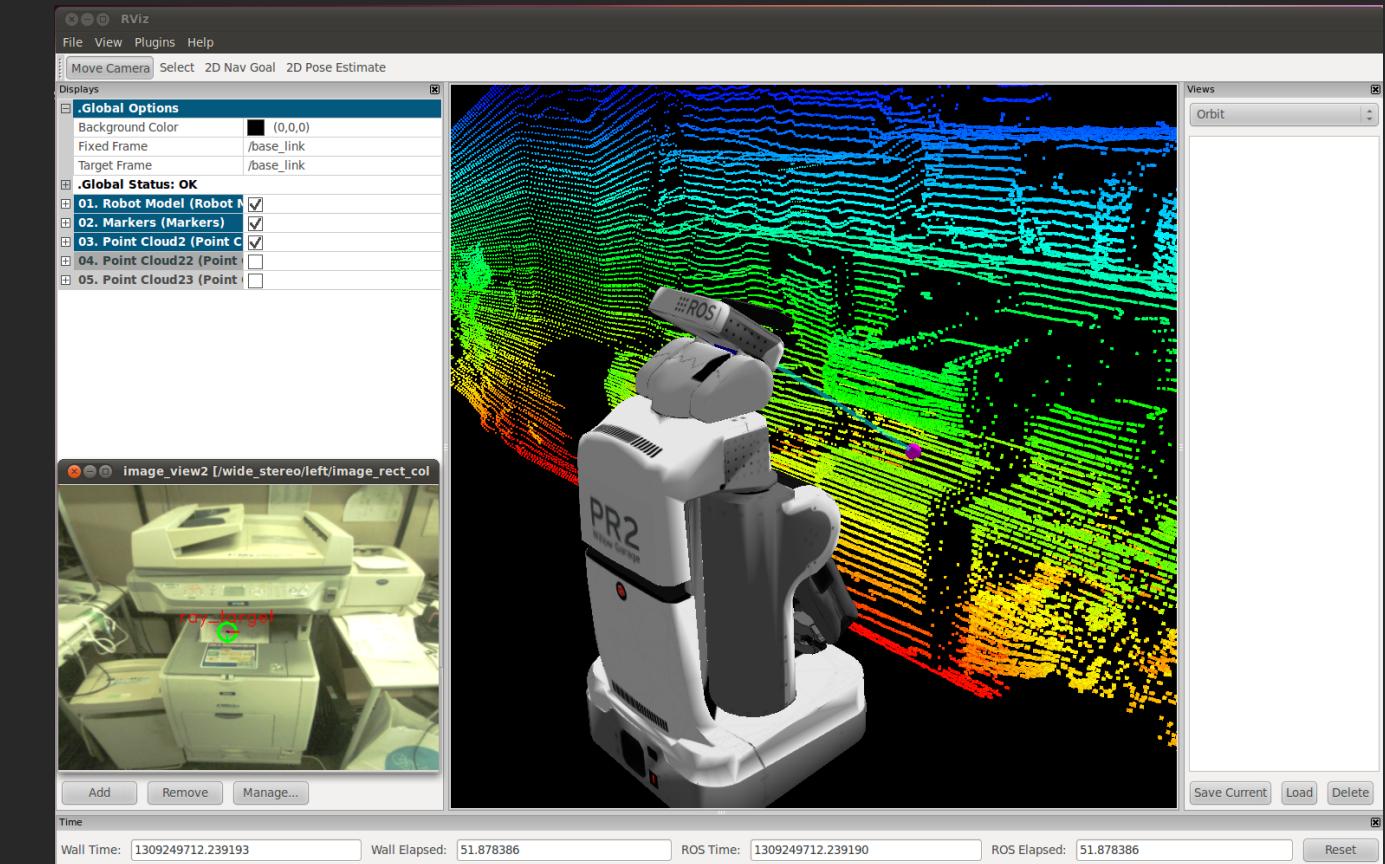
ROS 2

- Designed specifically for commercial applications with reliability, **security**, and consistency in mind
- Open source, Apache 2.0
- Open **DDS** (Data Distribution Service) standards
- **Industry support** – Amazon, Intel, Bosch, Microsoft, LG Electronics, Toyota Research Institute, etc.

# What can ROS do?

You can find packages for all kinds of things...

- Sensors (LIDAR, depth camera)
- Algorithms (Navigation, SLAM so robots know where they are in their environment)
- Simulations (2D, 3D)
- Debugging tools
- Logging & testing





# What's the same between ROS 1 and ROS 2?

# What's the same?

- Nodes, messages, publishers, subscribers
- Command line & graphical tools
  - RViz
  - rqt
  - Gazebo





# What's different between ROS 1 and ROS 2?

# Example project layouts: ROS 1 vs. ROS 2

Project

```
├── doc  
│   └── index.md  
├── project_msgs  
│   ├── msg  
│   │   ├── Foo.msg  
│   │   └── Bar.msg  
│   └── CMakeLists.txt  
└── package.xml
```

project\_utils

```
├── launch  
│   └── launch_robot.xml  
└── scripts  
    └── do_stuff.cpp
```

```
└── CMakeLists.txt
```

```
└── package.xml
```

Project

```
├── doc  
│   └── index.md  
├── project_msgs  
│   ├── msg  
│   │   ├── Foo.msg  
│   │   └── Bar.msg  
│   └── setup.py  
└── package.xml
```

project\_utils

```
├── launch  
│   └── ...  
└── scripts  
    └── do_stuff.py
```

```
└── setup.py
```

```
└── package.xml
```

# What's different between ROS 1 and ROS 2?

1. Security
2. DDS
3. `Catkin_make` is gone: Catkin has been replaced by colcon
4. Launch files using .xml format are gone

## 5. Platforms:

- ROS 1 is only CI tested in Ubuntu right now
- ROS 2 is currently being CI tested and supported on Ubuntu Xenial, OS X El Capitan, and Windows 10

## 6. Languages:

### a. C++

- ROS 1 targets C++03 and doesn't make use of C++11 features in its API
- ROS 2 uses C++11 extensively and some C++14. (Maybe also C++17 soon!)

### b. Python

- ROS 1 targets Python2
- ROS 2 requires at least Python version 3.5



# Overview of ROS 2 Design

The **ROS graph** refers to the network of **nodes** in a ROS system and the connections between them by which they **communicate**

# Quick overview: ROS 2 graph concepts

## Nodes:

An entity that uses ROS to communicate with other nodes

## Messages:

ROS data type used when subscribing or publishing to a **topic**

## Topics:

Nodes can **publish** messages to a topic as well as **subscribe** to a topic, to receive messages

## Discovery:

The automatic process through which **nodes** determine how to talk to each other

ROS node  
(Publisher)

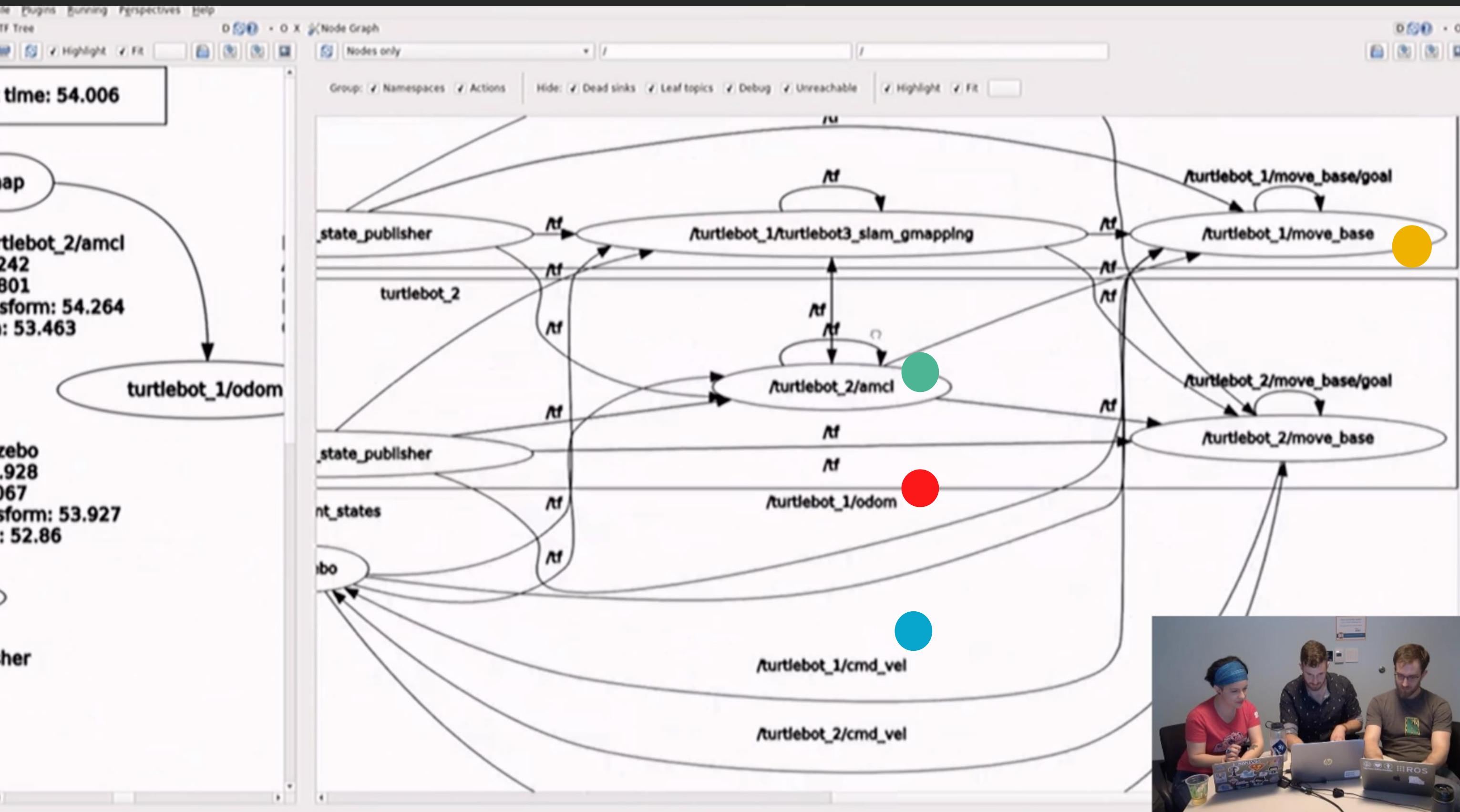
Topic:  
`/dogs`

MessageType:  
`std_msgs/String`

`/camera/depth/camera_info`  
`/camera/depth/image_rect_color`  
`/camera/depth/points`  
`/camera/parameter_descriptions`  
`/camera/parameter_updates`  
`/camera/rgb/camera_info`  
`/camera/rgb/image_raw`  
`/camera/rgb/image_raw/compressed`  
`/camera/rgb/image_raw/compressed/parameter_descriptions`  
`/camera/rgb/image_raw/compressed/parameter_updates`  
`/camera/rgb/image_raw/compressedDepth`  
`/camera/rgb/image_raw/compressedDepth/parameter_descriptions`  
`/camera/rgb/image_raw/compressedDepth/parameter_updates`  
`/camera/rgb/image_raw/theora`

ROS node  
(Subscriber)

ROS node  
(Subscriber)



# What is DDS?

- Stands for Data Distribution Service
- DDS is networking middleware that simplifies complex network programming
- It implements a publish–subscribe pattern for sending and receiving data, events, and commands among the nodes

# DDS in ROS 2: An end-to-end middleware

It just has so many **benefits...**

- a) Less code to maintain
- b) System-level documentation
- c) Recommended use cases
- d) A software API
- e) Etc.

Want to dive deeper?



This article made the case for using DDS as the middleware for ROS 2:

[http://design.ros2.org/articles/ros\\_on\\_dds.html](http://design.ros2.org/articles/ros_on_dds.html)



# Who should be porting to ROS 2?

# Depends on your demographic!

- **Large companies**

(Those who are in R&D groups funded by a large corporate entity)

- Strongly consider ROS 2 to reduce the amount of technical debt in the future
- Put people with experience with ROS 1 on the project

- **New robotics startups**

- Strongly consider ROS 2 to reduce the amount of technical debt in the future
- Put people with experience with ROS 1 on the project

- **Robotics OEM**

(Those who make either robots, sensors for robots, or anything that needs a ROS driver)

- ROS 2 Dashing is the first LTS release, so it's now safe for OEMs to start porting drivers without fear of new features that will break functionality
- Large industry support: Amazon, Intel, and Microsoft devote significant resources towards ROS 2 development



# Who is still on ROS 1 and doesn't really need to rush to ROS 2 yet?

# Again... Depends on your demographic!

- **Students**

(Those who are just learning to use ROS)

- Stick with ROS 1 for now. Many of the concepts in ROS 1 and ROS 2 are the same so learning ROS 1 will help you to learn ROS 2 later on.

- **Professors**

(Those teaching ROS)

- Keep teaching ROS 1 for now, but start thinking about curriculum for ROS 2

- **Researchers**

(Those using ROS to publish papers)

- Unless your paper is specifically showing off new features of ROS 2, you probably want to stick with ROS 1 for the time being



# Why did Rover Robotics decide it was worth the effort to port to ROS 2?

# Rover Robotics

# Rover Robotics: Who are they?

- They took the #1 police and public safety robot and made it compatible with ROS and open source software
- They “want robots to be as affordable and feature-rich as modern-day laptops and smartphones”

The 2-wheel drive (2WD) rover excels at indoor use. When used on flat terrain it can handle payloads up to 100kg (220lbs).

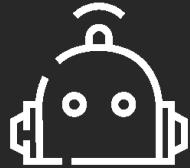


# Rover Robotics: Why port to ROS 2?

- ❑ The **reliability**
- ❑ The **big company players** who are developing with ROS 2
- ❑ The **big paradigm shift** happening to ROS 2

Check out their blog, they detail their adventures with ROS 2...  
<https://blog.roverrobotics.com>

# Want to Dive Deeper?



Here are the docs for ROS 2 concepts:

<https://index.ros.org/doc/ros2/Concepts/Overview-of-ROS-2-concepts>

Changes between ROS 1 and ROS 2:

<http://design.ros2.org/articles/changes.html>

ROS 2 on DDS:

[http://design.ros2.org/articles/ros\\_on\\_dds.html](http://design.ros2.org/articles/ros_on_dds.html)

Let's talk...

I make it a point to answer all my DMs and emails from customers after I give a talk

[quetzal@amazon.com](mailto:quetzal@amazon.com)

@QuetzalliAle

# Thank you!

**Alejandra Olvera-Novack**

quetzal@amazon.com



Please complete the session  
survey in the mobile app.