



# ROS Workshop

## (Robot Operating System)



Source: Willow garage



# Chapter I

Abstract examples

# Chapter II

OS, ROS & Users

# Chapter III

ROS specific



# Chapter I

Source: sflow



**ROSS**  
**DRESS FOR LESS®**

Source: ROSS

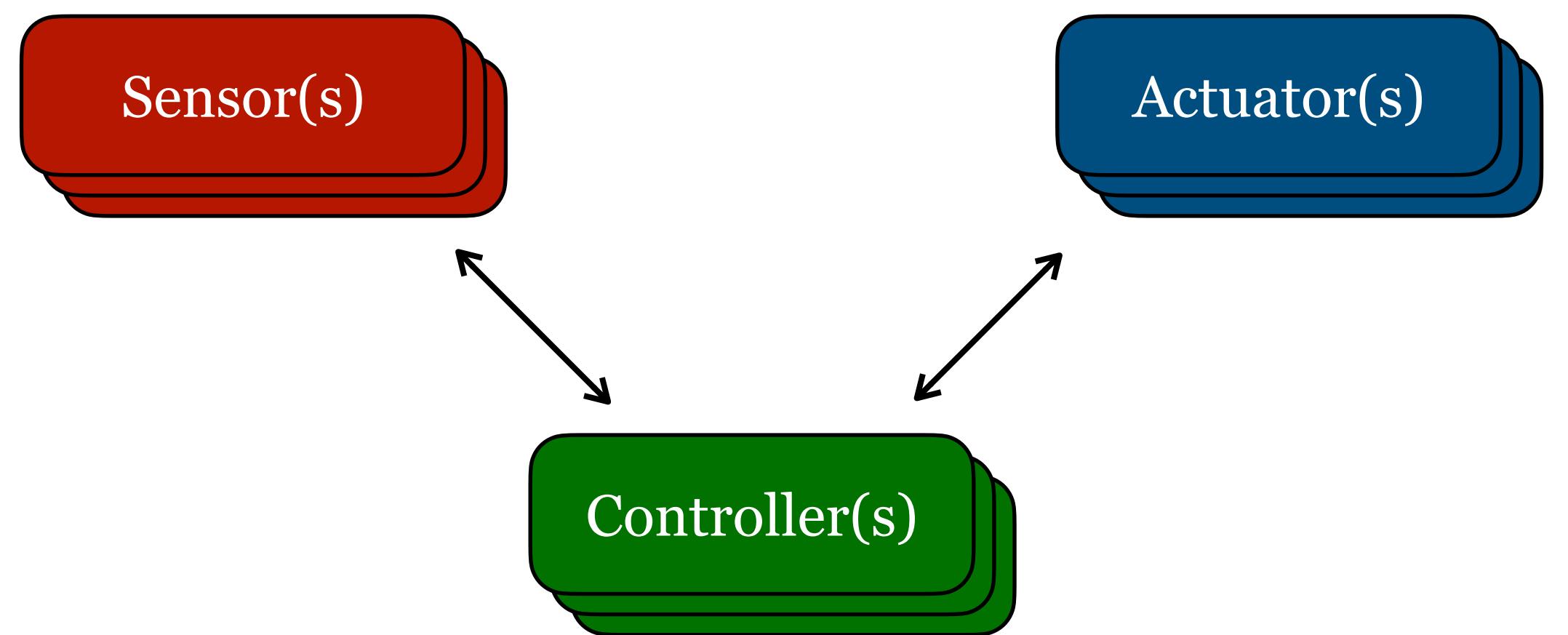


# What is ROS ?

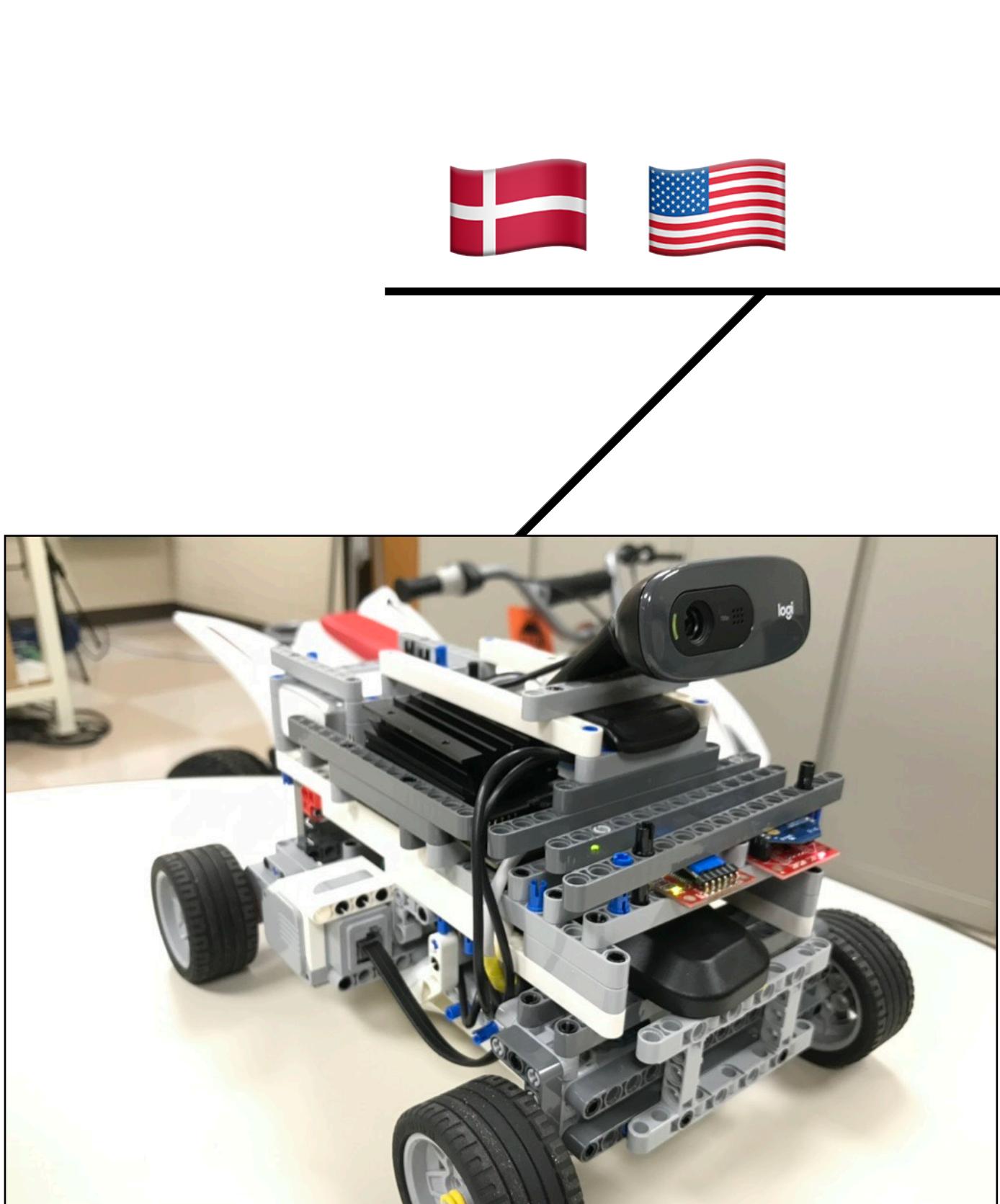
ROS is an open-source framework that helps researchers and developers build and reuse code between robotics applications.

ROS (a middleware) is also a global open-source community of engineers, developers and hobbyists who contribute to making robots better, more accessible and available to everyone.

# Activity - Robot 101

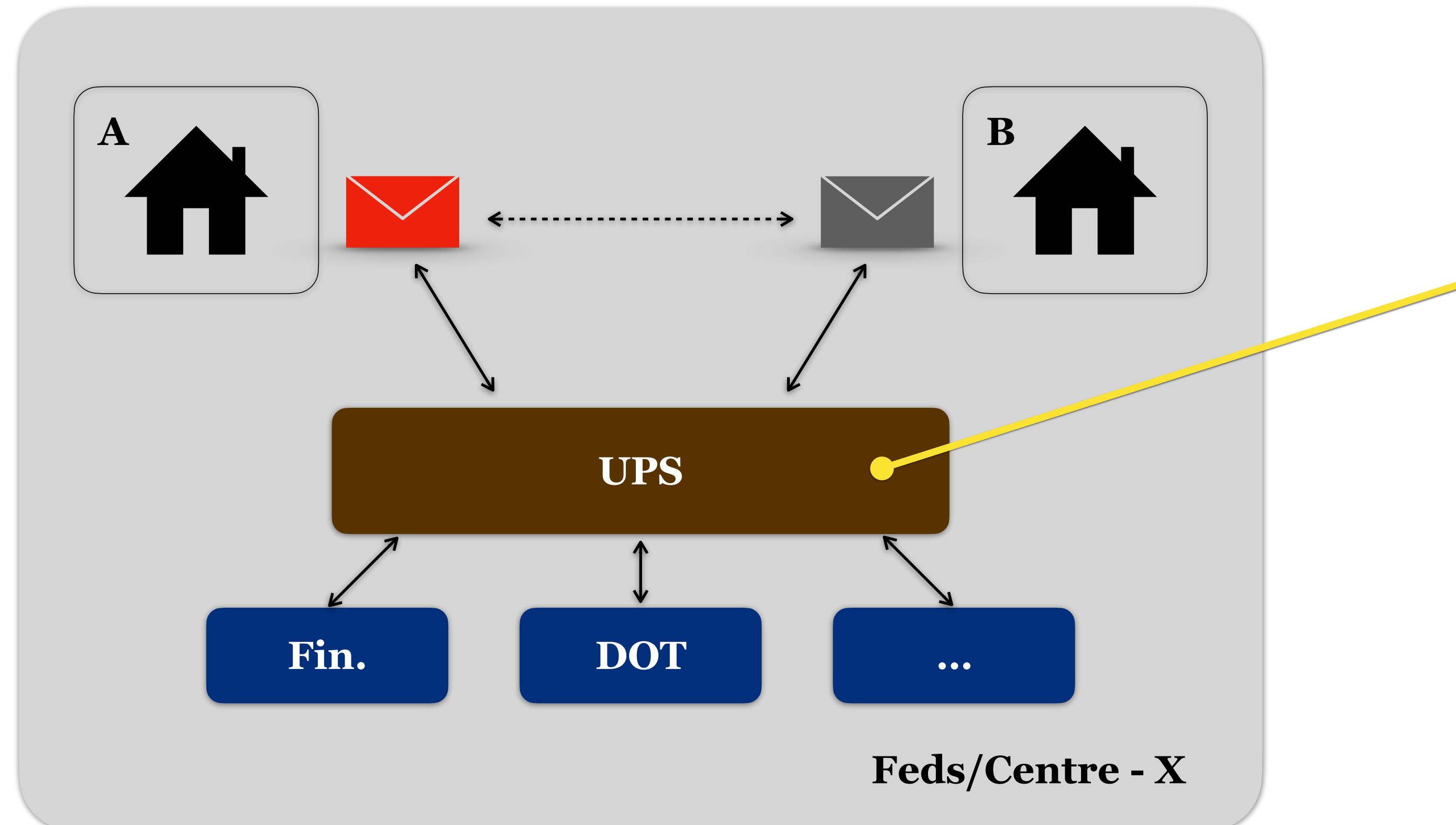


# LEGO



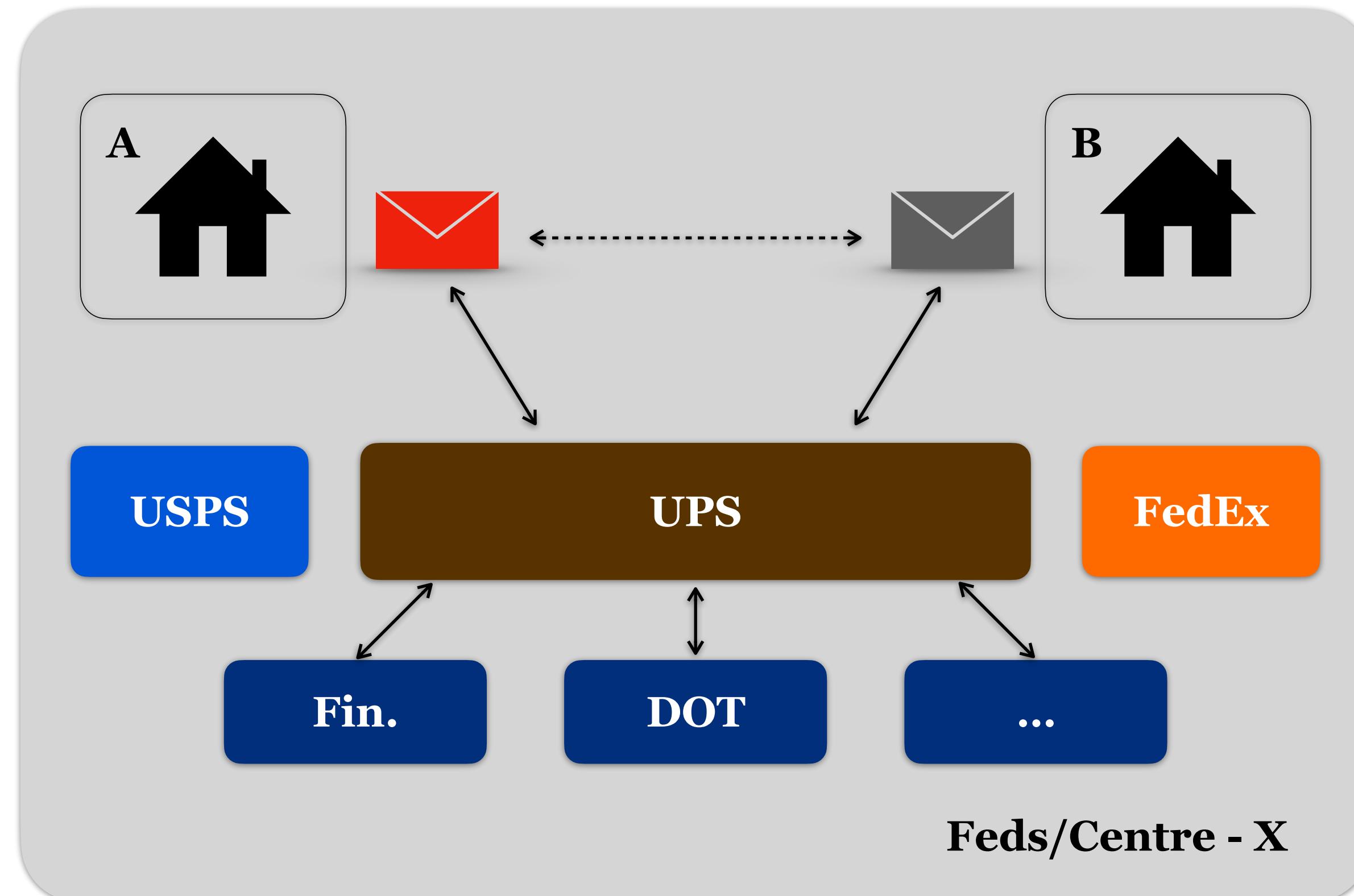
Source: Lego

# Package Delivery



- Tracking tools
- Counterfoil
- Shipping tier options
- Ecosystem

# Package Delivery - Alternatives



## ROS Alternatives

- PLC
- Orocos
- CODESYS
- V-Rep
- Data Turbine
- MatLab / Simulink

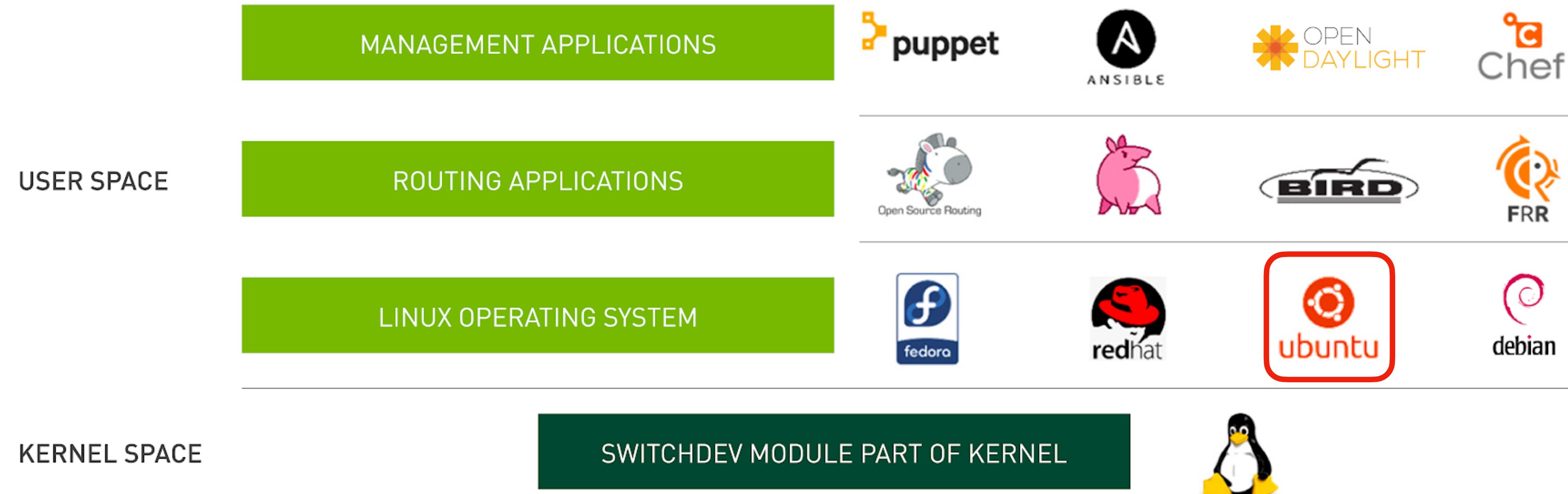
⋮



# Chapter II

Source: sflow

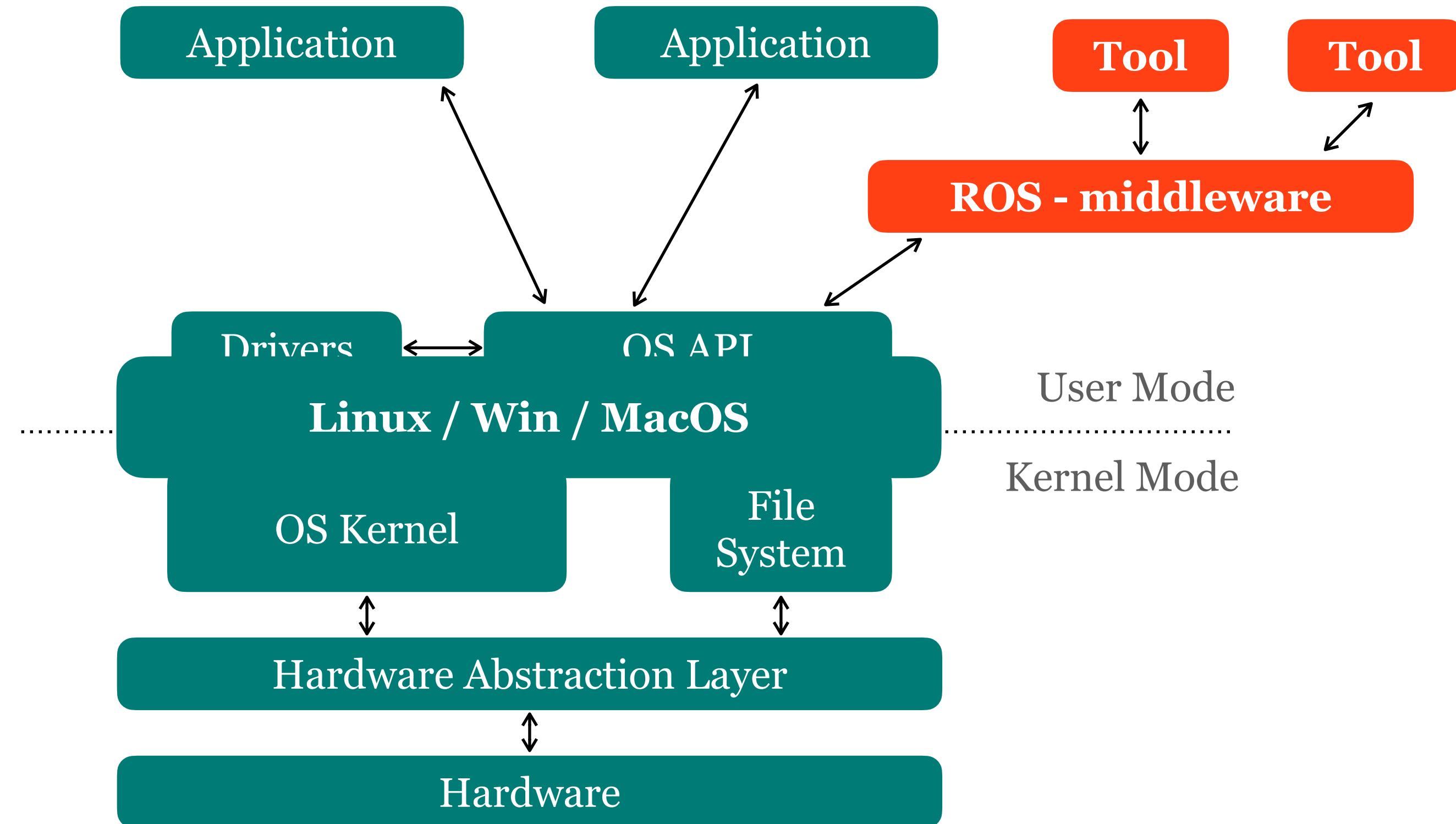
# Understanding Ubuntu OS



- **Ubuntu distro.** is more popular and has a predictable release cycle
- Even numbered releases are ones w/ Long Term Support (LTS)

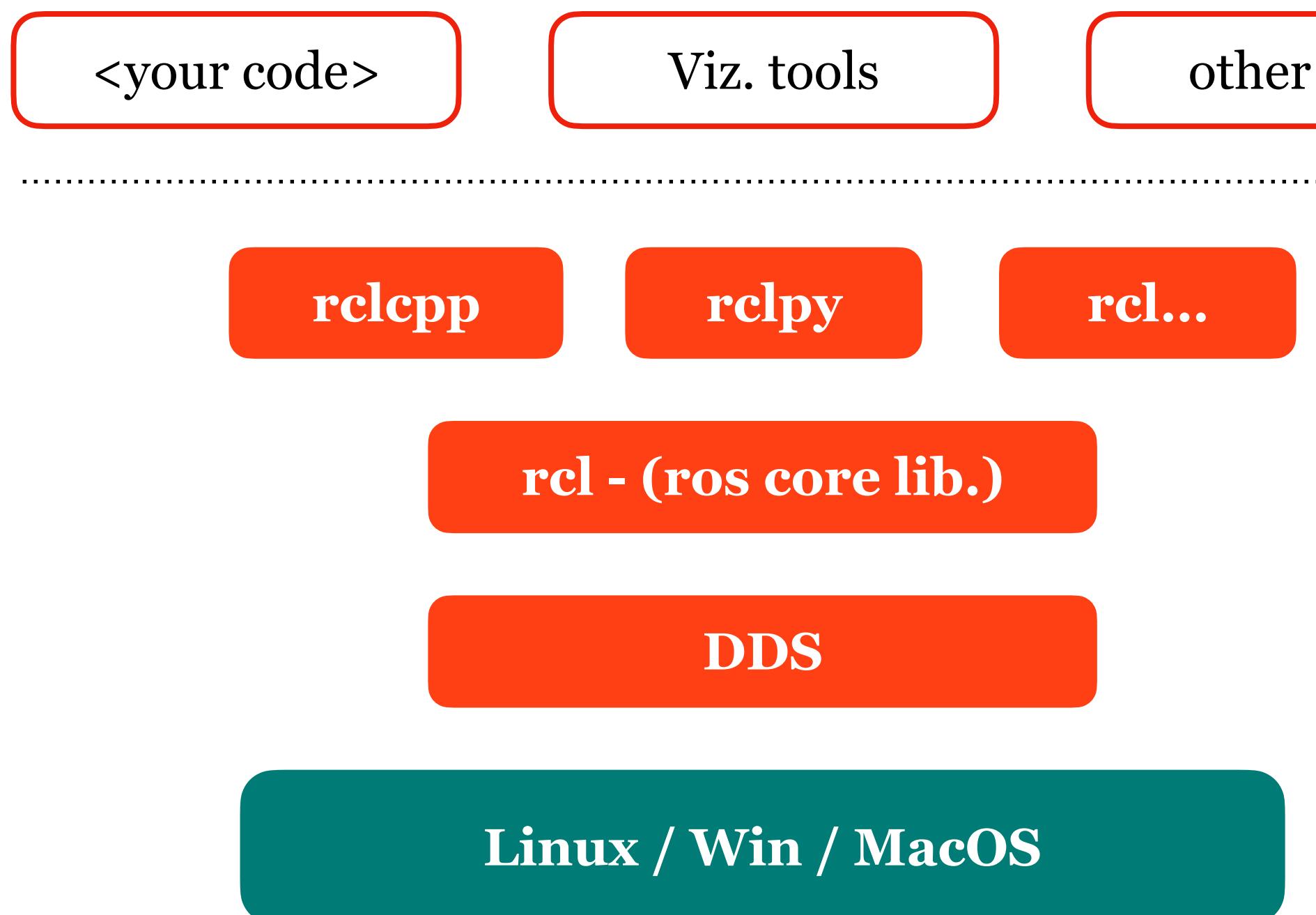
Source: sflow

# Ubuntu/OS Framework & ROS





# Ubuntu/OS Framework & ROS



- C++, Python, Nodejs, ...
- Real time\*
- Safety & Security
- Certification
- Distributed communication
- Compatible w/ industrial applications

# ROS 2 - OS & Hardware Compatibility

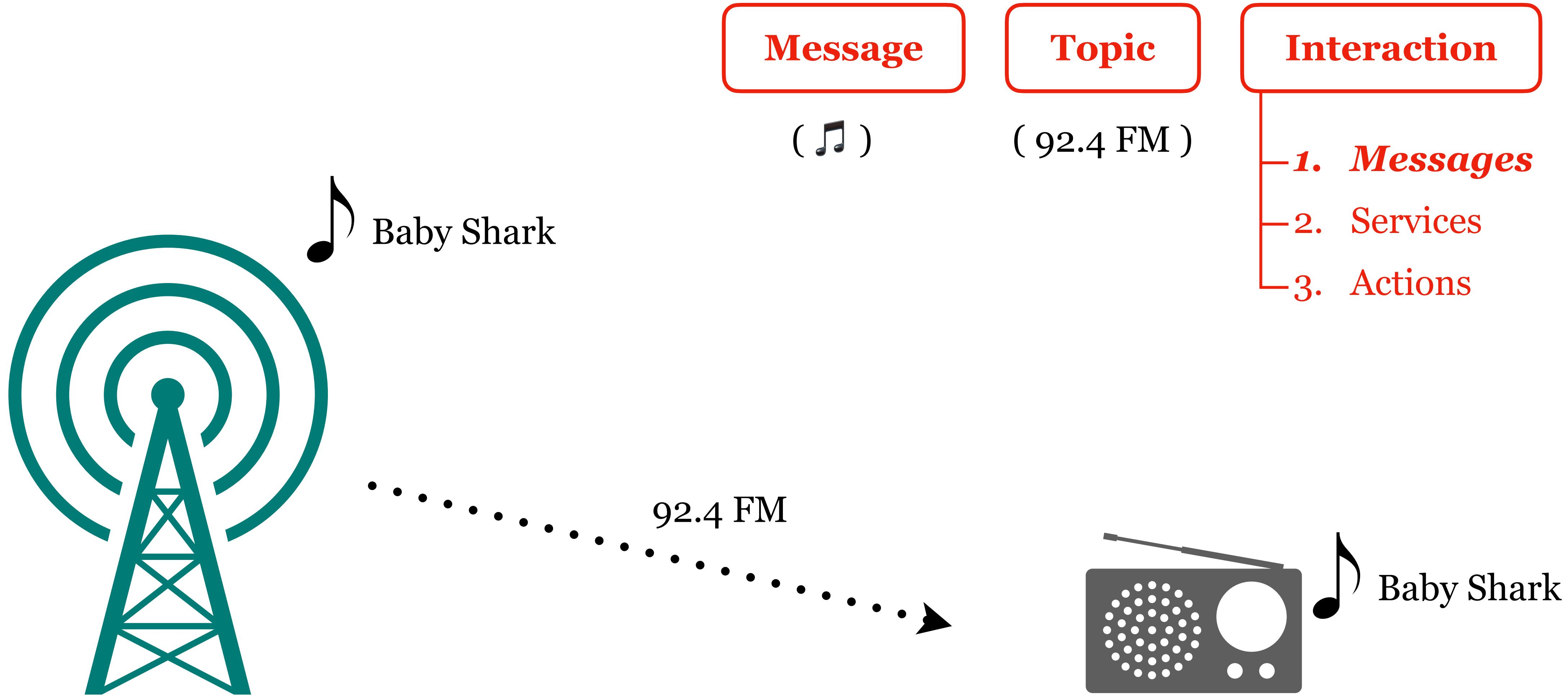
ROS Versions	Ardent 2018	Foxy 2020 - 2023	Humble 2022 - 2027	Jazzy 2024 - 2029
Software	Ubuntu 18	Ubuntu 20	Ubuntu 22	Ubuntu 24
Hardware	Jetson Nano	NVIDIA Jetson Orin (family)	Raspberry Pi 3 and/or 4	PC/NUC

Official support



# Chapter III

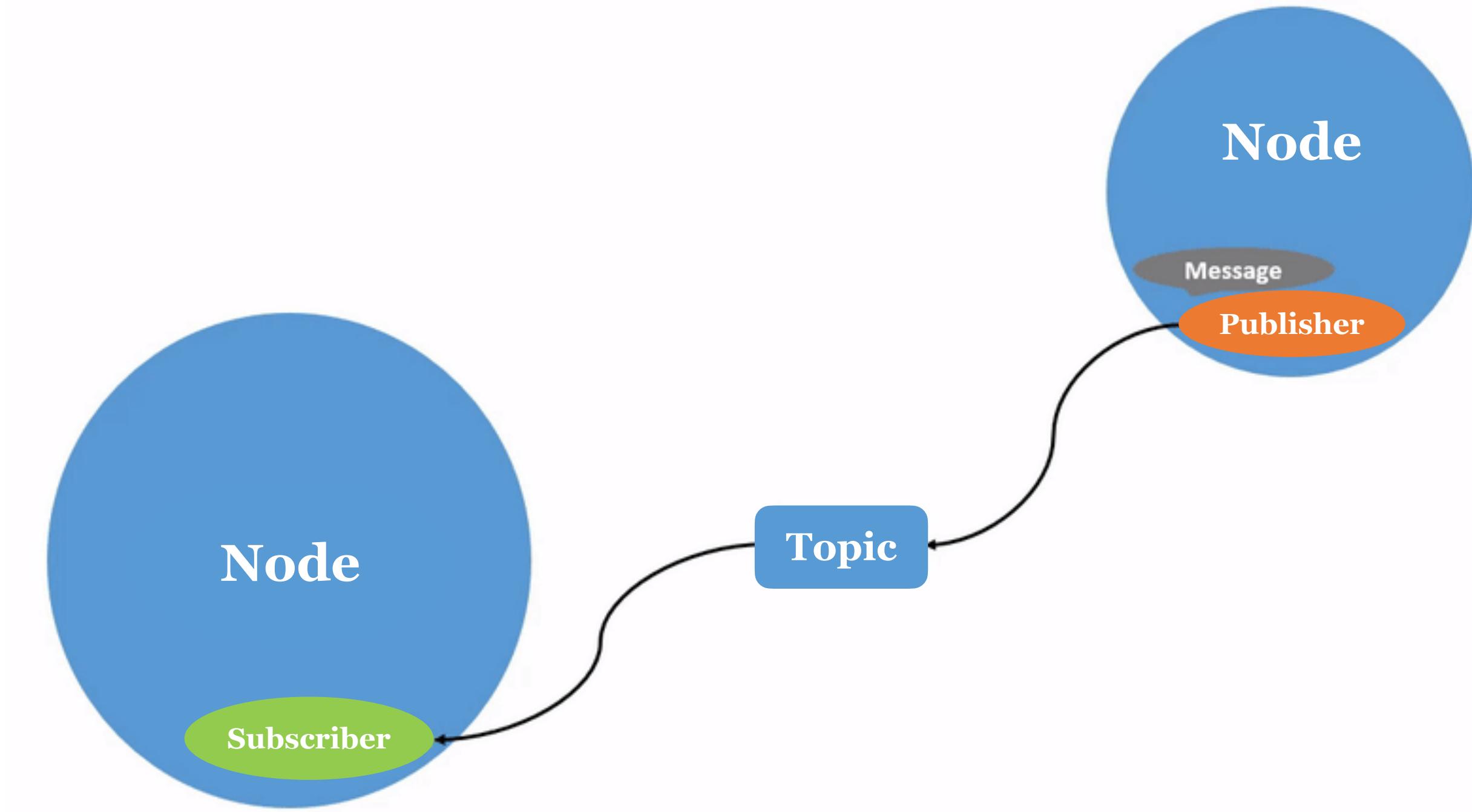
# ROS Core Interaction Types



For illustration purposes

# Understanding (1/3) ‘Messages’ (... and Topics)

- Messages are sent from node to node on a topic
- Broadcast rate
- Message types
- Many-to-many
- Synchronous/Asynchronous

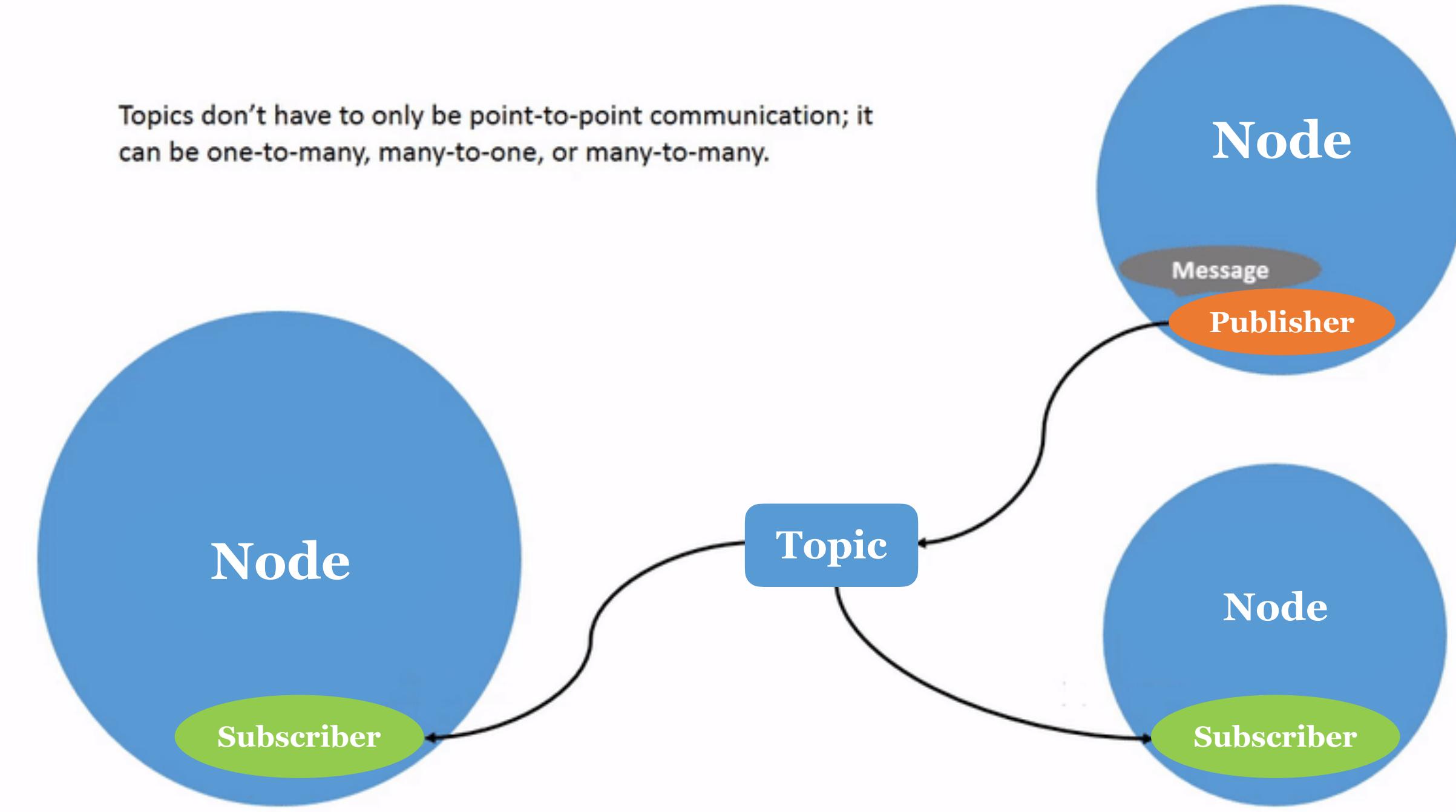


Source: ROS docs

# Understanding (1/3) ‘Messages’

- Point-to-point
- One-to-many
- Many-to-one
- Many-to-many

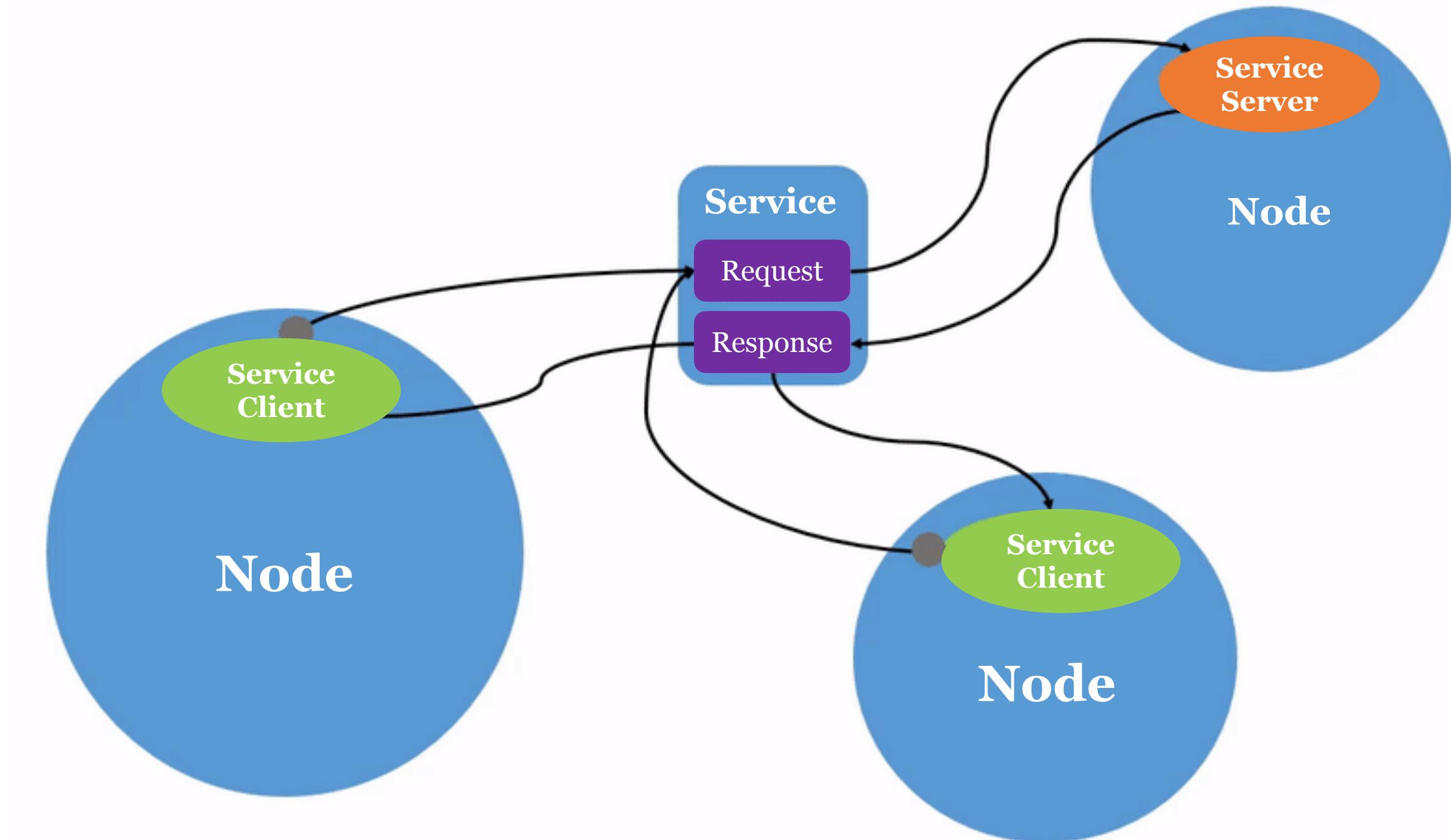
Topics don't have to only be point-to-point communication; it can be one-to-many, many-to-one, or many-to-many.



Source: ROS docs

## Understanding (2/3) ‘Services’

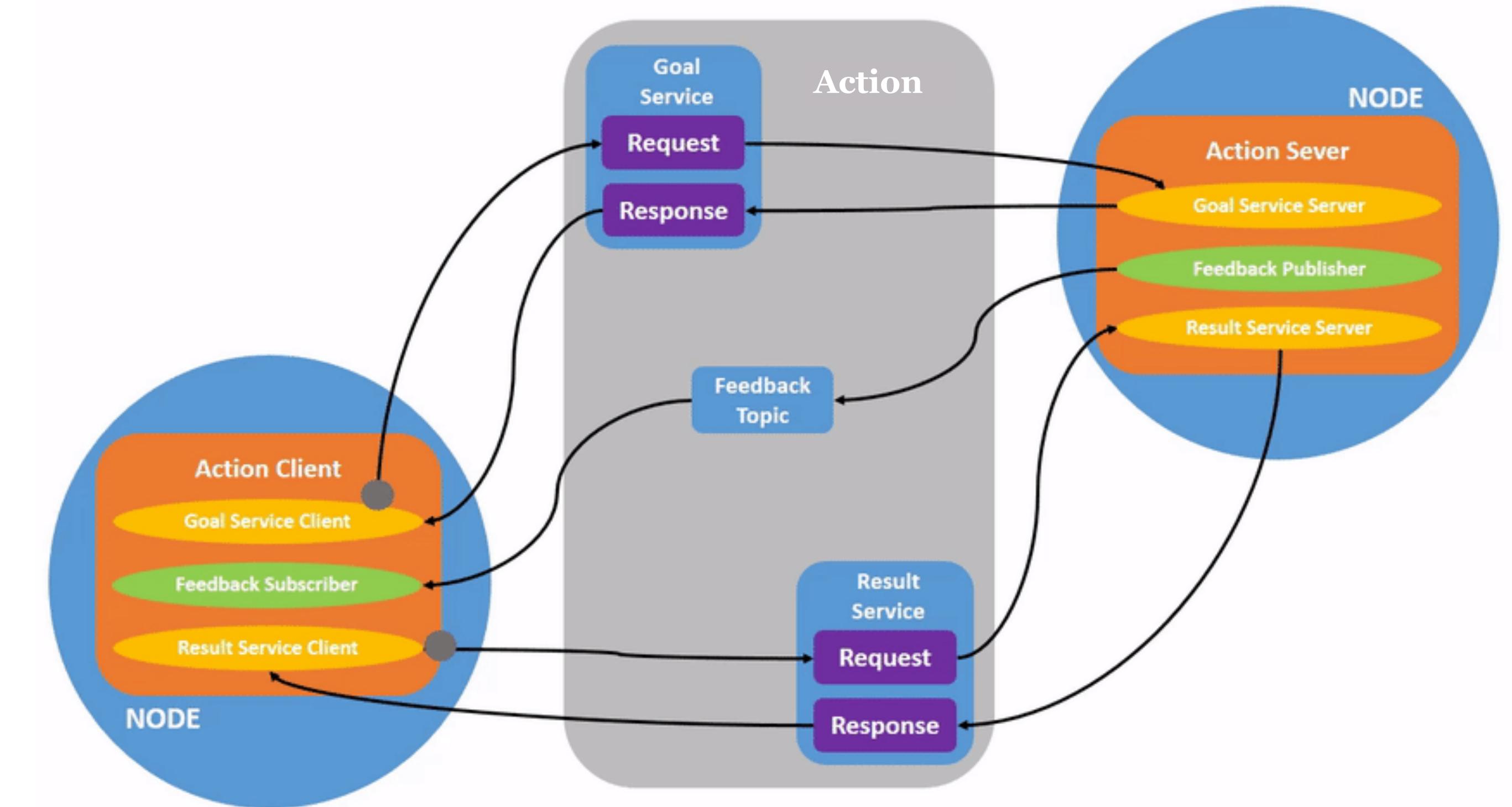
- Service call to acknowledge an event
- Uses client-server model
- Synchronous/Asynchronous
- Eg. open door then proceed
- Eg. Release brakes before motion



Source: ROS docs

## Understanding (3/3) ‘Actions’

- Initiate/Trigger an Action
- Built on Topics & Services
- Asynchronous com.
- Longer running process. Eg. go to waypoint
- Steady feedback is sent back by action server



Source: ROS docs

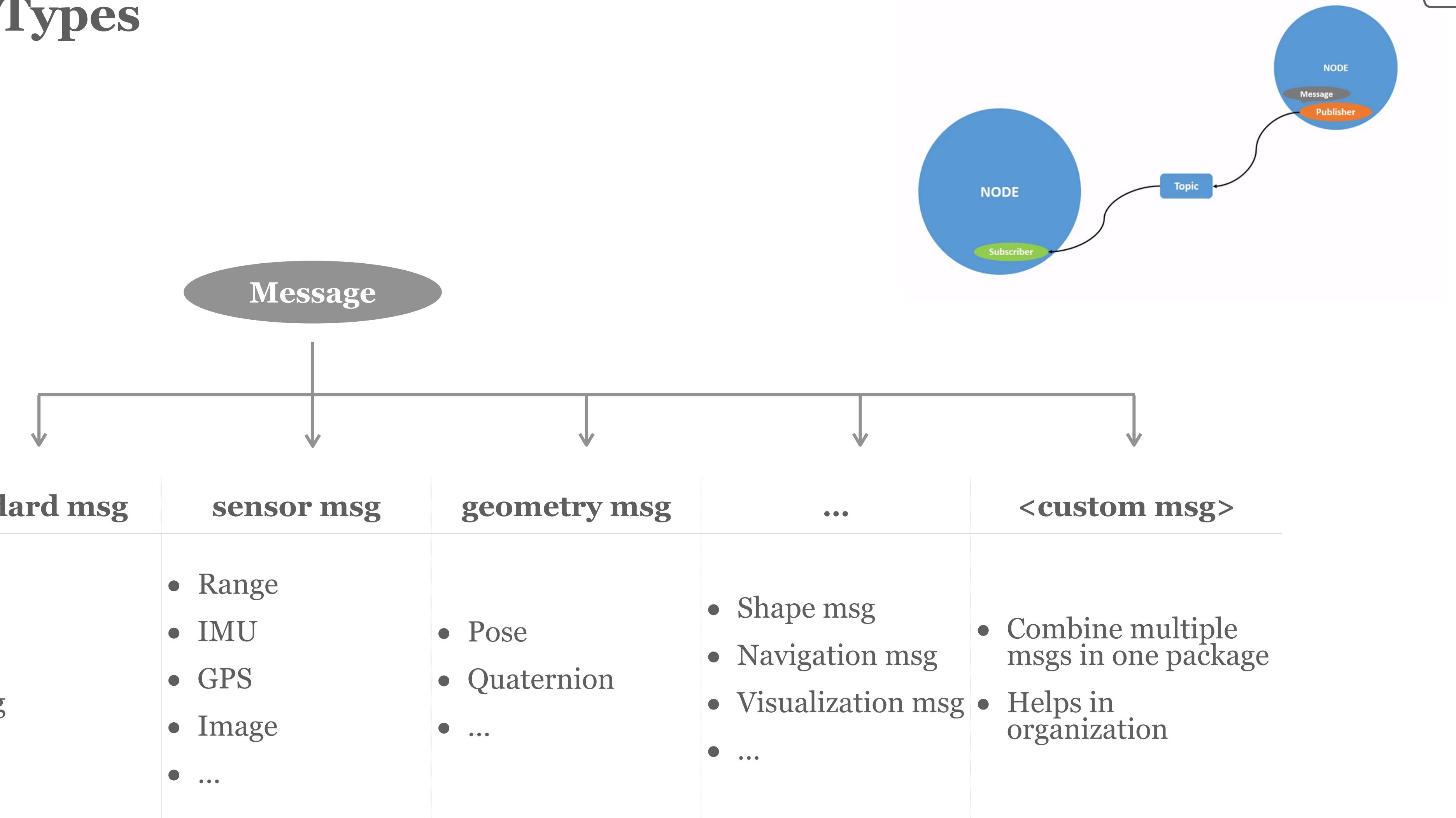


# Summary

Type	Strengths	Weaknesses
Message	<ul style="list-style-type: none"><li>Good for most sensors</li><li>One-to-many</li></ul>	<ul style="list-style-type: none"><li>Message drop w/o knowledge</li><li>Easy to overload</li></ul>
Service	<ul style="list-style-type: none"><li>Knowledge of missed call</li><li>Well defined feedback</li></ul>	<ul style="list-style-type: none"><li>Blocks until completion</li><li>Connection typically re-establish for each service call (slow activity)</li></ul>
Actions	<ul style="list-style-type: none"><li>Monitor long running processes</li><li>Handshake</li></ul>	<ul style="list-style-type: none"><li>Complicated</li></ul>

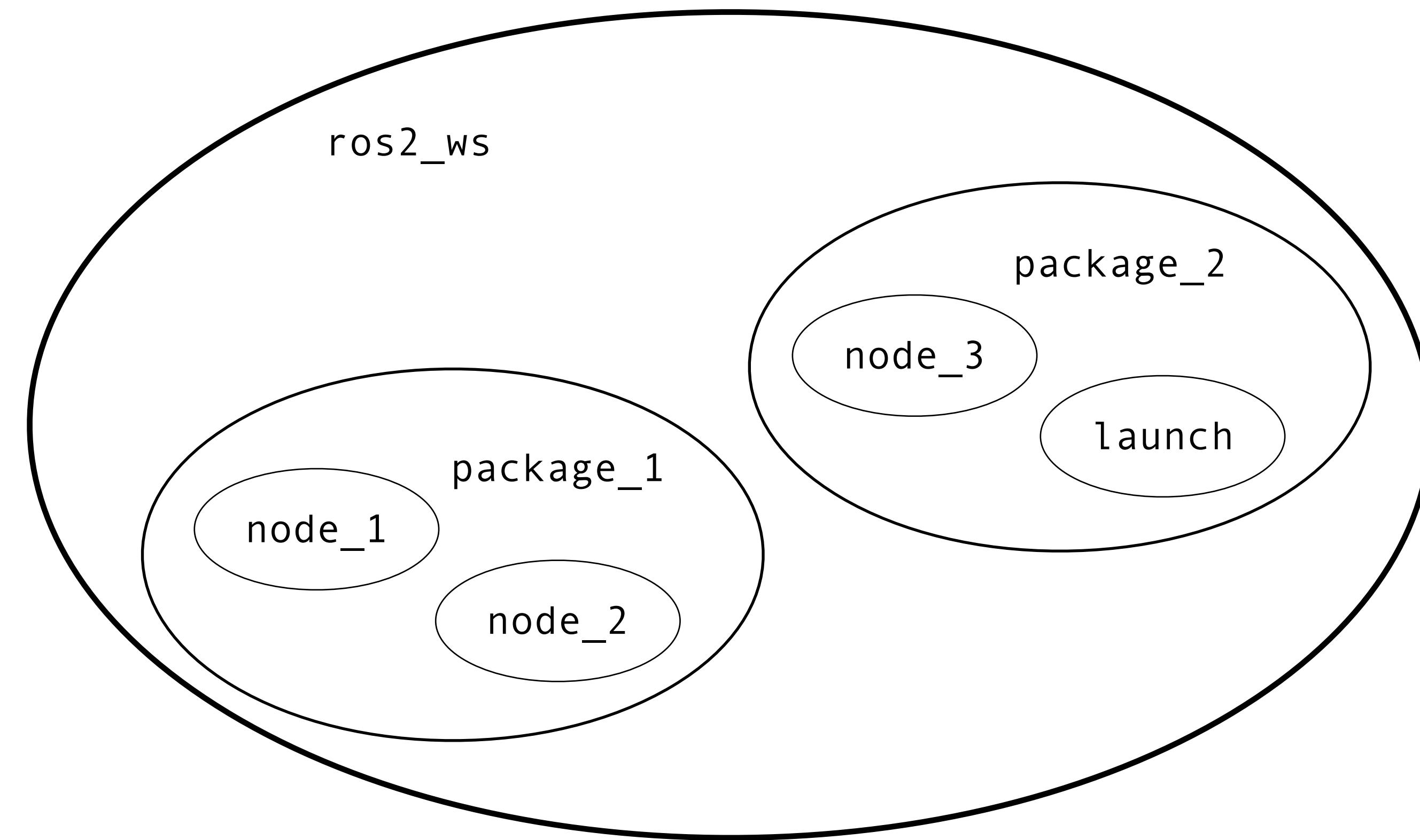
# Message Types

III



Source: ROS docs

# ROS Workspace





# Is ROS for me ?

## Pros

- Recycle, reuse & share
- Debugging tools
- DDS secure service
- Easy network setup
- Large & active community

## Cons

- Overhead
- Intricate & barrier to entry
- Lacks commercial support
- Limited cross-platform sup

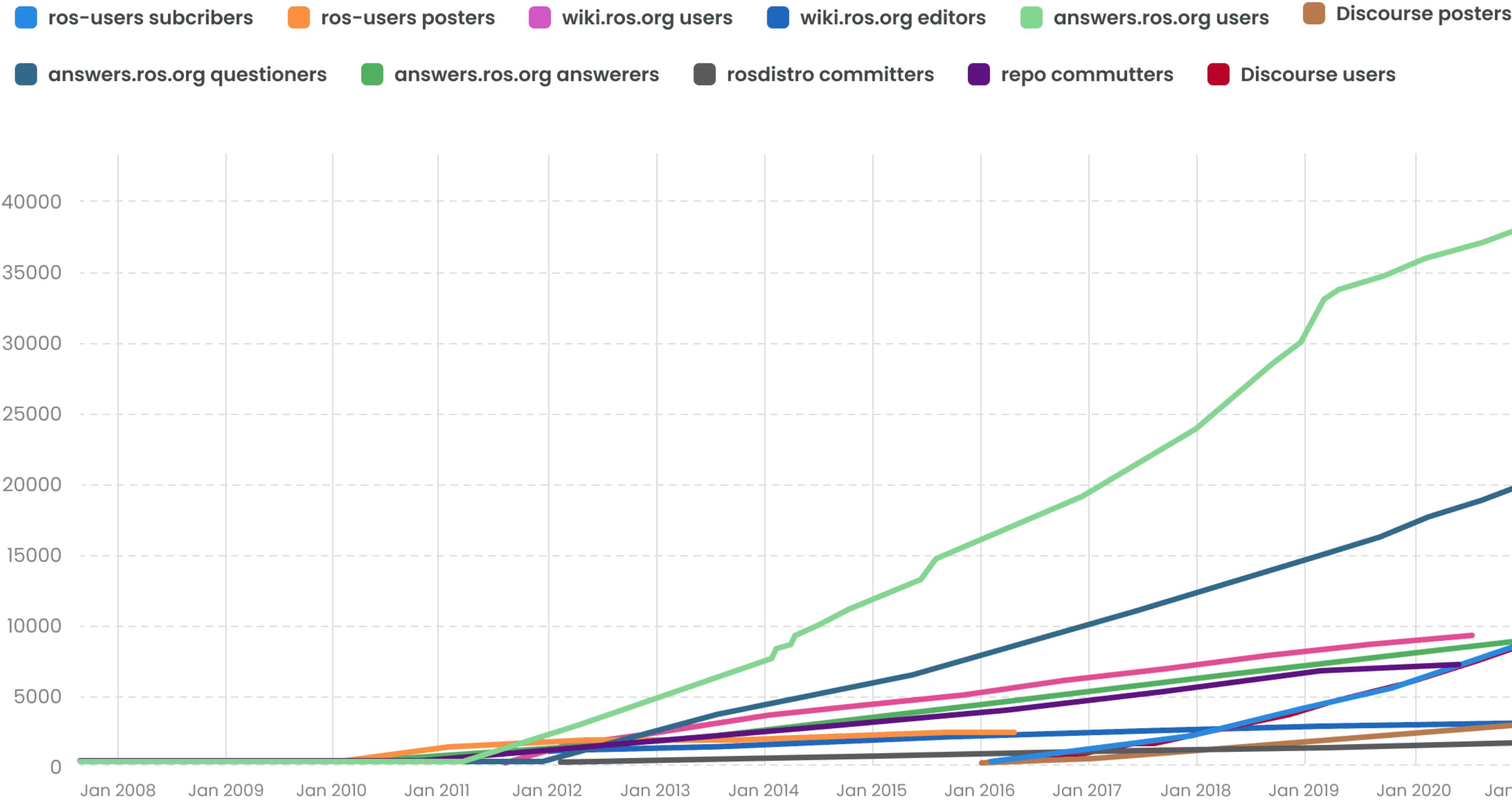


# State of ROS

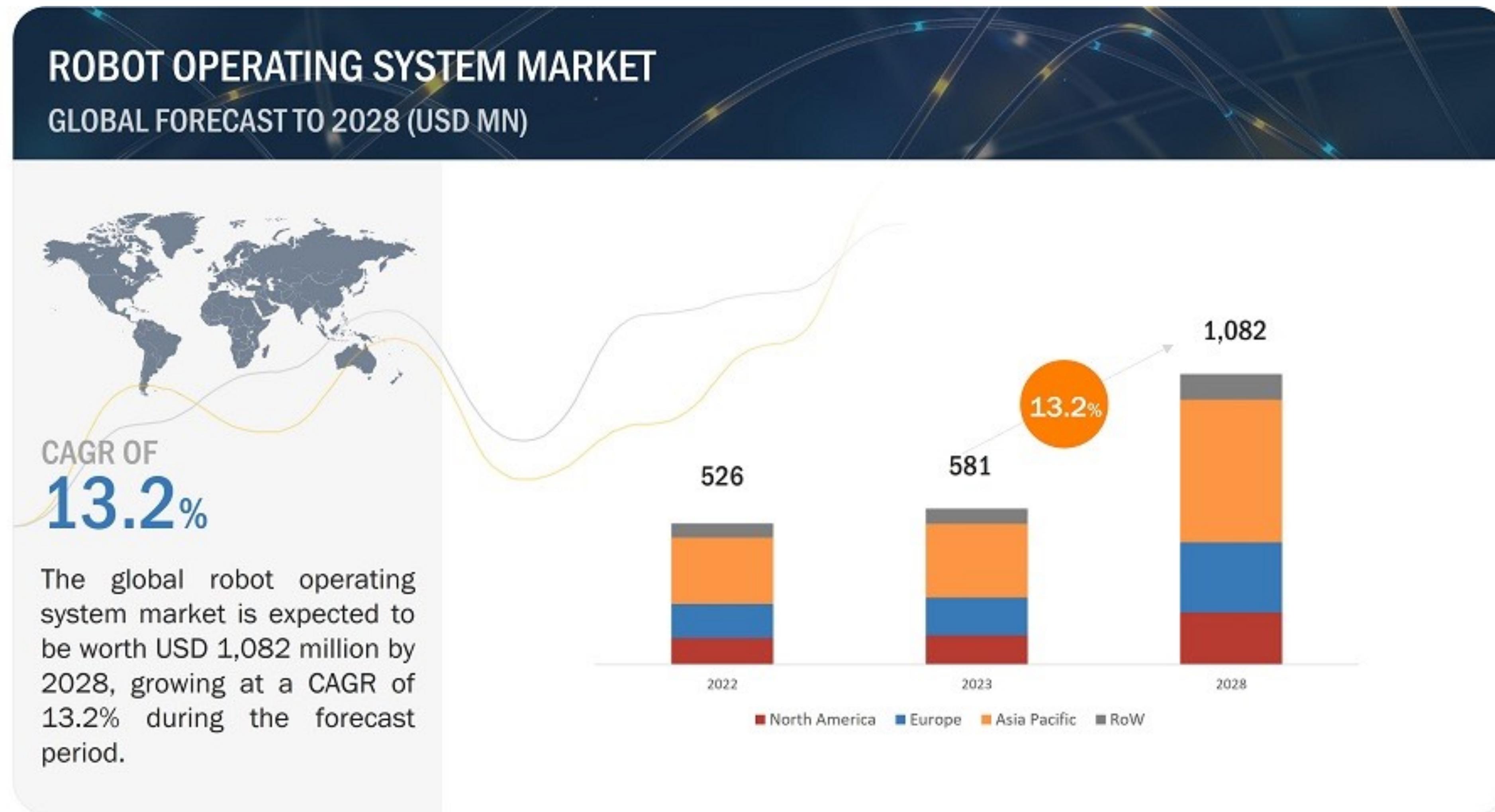
Hobbyist	University	R&D*	Industry
<ul style="list-style-type: none"><li>• Price sensitive</li><li>• Limited expertise</li></ul>	<ul style="list-style-type: none"><li>• Scholarly inclination</li><li>• Generally price insensitive</li><li>• Intermediate- advanced expertise</li><li>• Larger scale projects</li></ul>	<ul style="list-style-type: none"><li>• Profit inclination</li><li>• Generally price insensitive</li><li>• Intermediate- advanced expertise</li><li>• Larger scale projects</li></ul>	<ul style="list-style-type: none"><li>• Profit oriented</li><li>• Price sensitive</li><li>• Hardened &amp; Robust solution</li><li>• Long term support</li></ul>

# Fun Facts ...

## Number of ROS Users


Source: [metrics.ros.org](https://metrics.ros.org)

## Fun Facts ...

Source: [metrics.ros.org](https://metrics.ros.org)



III

# Fun Facts ...

The screenshot shows the top navigation bar of the NASA VIPER website. It includes a "Explore" button with a dropdown arrow, a search bar containing "Search...", the NASA logo, and three main menu items: "News & Events", "Multimedia", and "NASA+ LIVE". Below the main menu, there's a secondary navigation bar with links for "... / VIPER / VIPER Lunar Operations", "VIPER", "News", "Multimedia", "Overview", "Team", and "Media Resources".

## VIPER Lunar Operations

### VIPER's Mission Operations

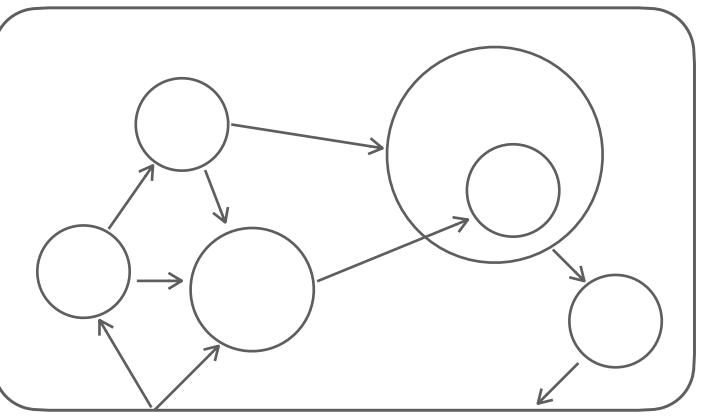
For the Volatiles Investigating Polar Exploration Rover, or VIPER, the lunar environment poses unique challenges alongside the rewards of discovering water and other resources as NASA seeks to establish a long-term presence on our nearest neighbor.

Operated interactively in near-real time by a NASA team back on Earth, the rover must navigate across the rugged terrain of the lunar South Pole while surviving ever-changing extremes in lighting and temperature and periods of communication blackouts. This scenario requires complex and dynamic route planning and waypoint driving to maximize the science return, while keeping the rover in good working condition until the end of its approximately 100-day mission.

Collaborations with industry are also feeding into key aspects of VIPER's software operating system. Rather than creating only custom code, the rover's flight and ground-based software will make extensive use of open-source software, including key components adapted from the Robot Operating System 2 (ROS 2) which is widely used in everything from robot vacuum cleaners to autonomous drones and self-driving cars. Once the mission is over, the VIPER team intends to release the rover's software for general use. This approach allows for a rapid, agile, and cost-effective way of developing the rover's software systems that can also benefit future rovers on the Moon and beyond.

Source: nasa

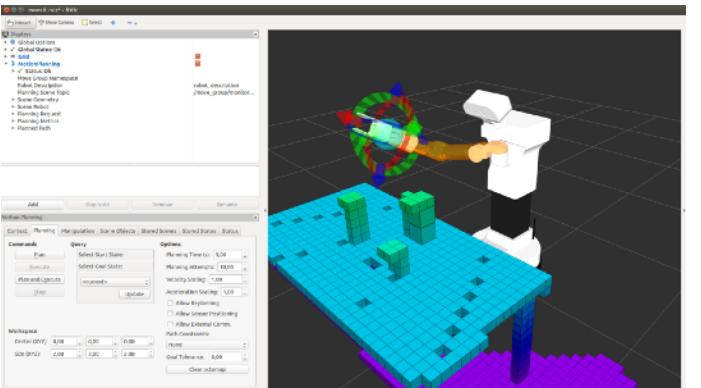
# Again ... what is ROS ?



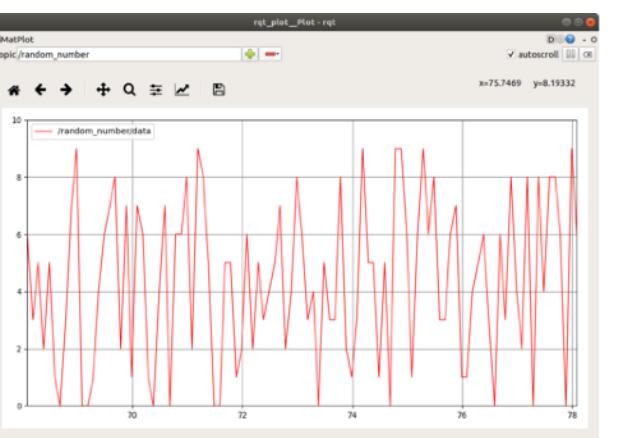
Plumbing



Ecosystem



Capabilities



Tools



# Key Takeaways

- ROS offers a “plumbing service”
- Message and Topics
- Workspace, Package, and Node



**END - PART I**