TEAM QUARK

Week 1

ROS2

Presented by : Aditya Sakhare

WEEK 1 : Introduction to ROS

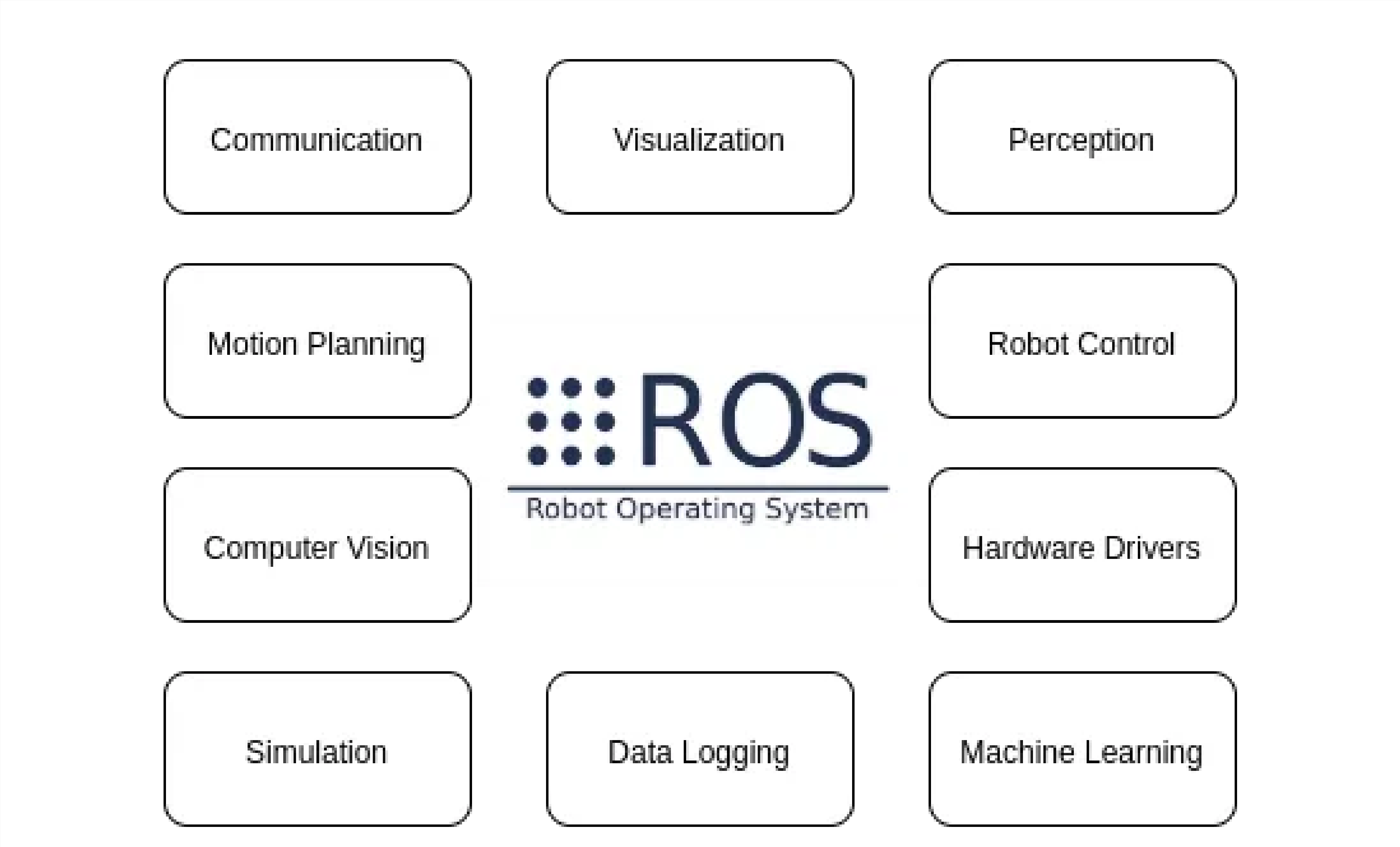
Basics of Robotics

- designing, building, programming machines to perform autonomously or semi-autonomously
- Sensors, Actuators, Control Systems
-
  - Applications
    - Industrial automation.
    - Healthcare (surgical robots).
    - Autonomous vehicles.

What is ROS 2?

- ROS: A middleware for developing robot applications.
- Why ROS2:
  - Modernized for multi-platform support.
  - Enhanced real-time capabilities with DDS.
  - Improved security and scalability.

What is ROS 2?



Communication

Visualization

Perception

Motion Planning

ROS

Robot Operating System

Robot Control

Computer Vision

Hardware Drivers

Simulation

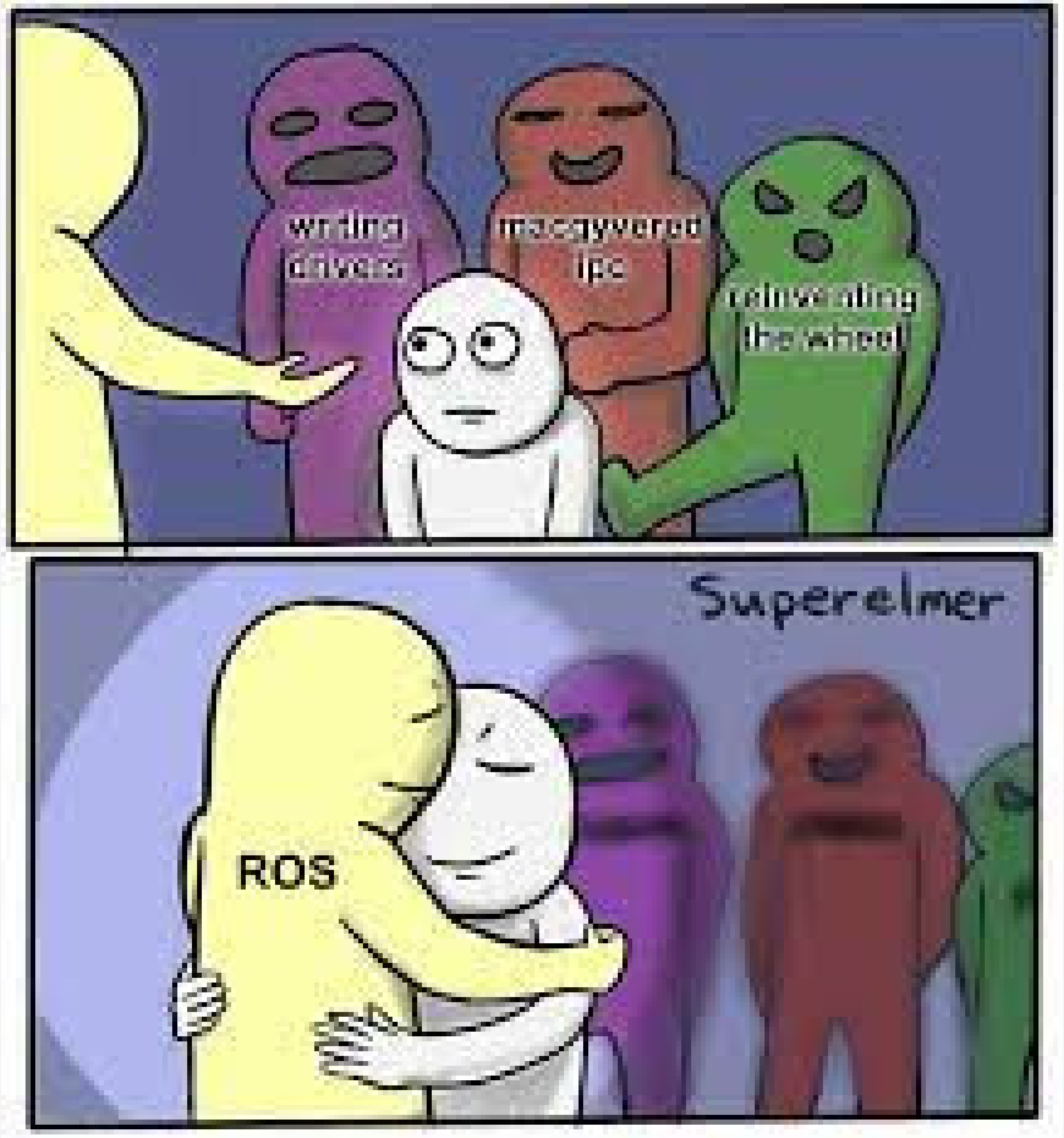
Data Logging

Machine Learning

Manually handling message serialization

Reinventing a Pub/Sub system with REST APIs

Creating custom multi-threading logic for robots

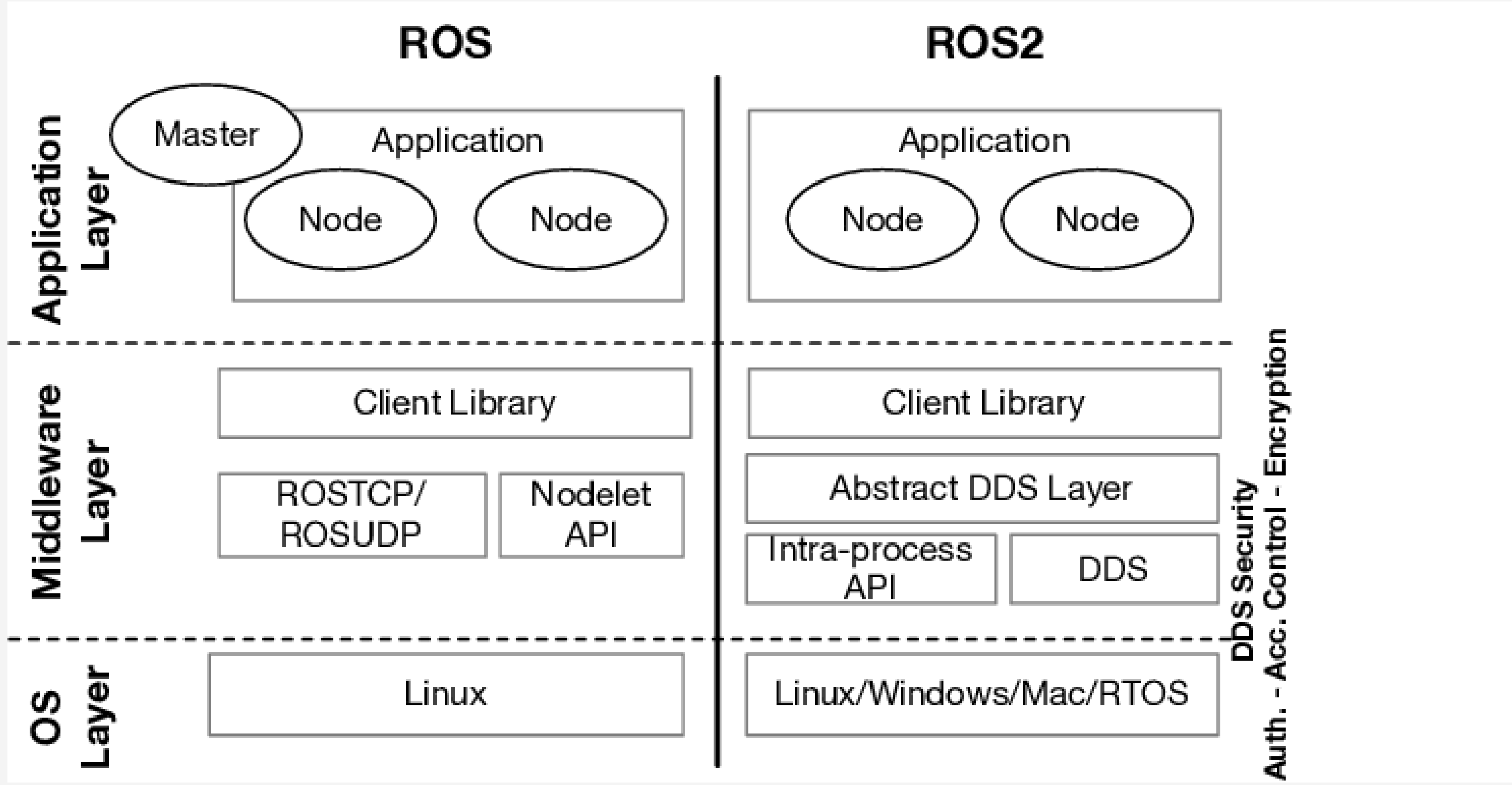


Why struggle? I’ve got a whole ecosystem for you!

Wait, ROS handles this for me?! Sign me up!

ROS1 vs ROS2?

Feature	ROS 1	ROS 2
Middleware	Custom XML-RPC	Data Distribution Service (DDS)
Communication Model	Master-Slave architecture	Distributed, no central master
Network Reliability	Less robust in unreliable networks (TCP based)	Better performance in unreliable networks (DDS with QoS)
Platform Support	Primarily Linux	Cross-platform (Linux, Windows, macOS)
Security	Limited security features	Built-in security support
Build System	Less flexible build system	Supports multiple build systems (including plain Python packages)
Nodelet Functionality	Separate feature requiring additional implementation	Integrated directly into core as "Components"
Development Status	Mostly deprecated, limited active development	Actively developed and maintained
Key Advantages	Large existing community and legacy code base	Modern architecture, improved reliability, better network support, security features



again....ROS

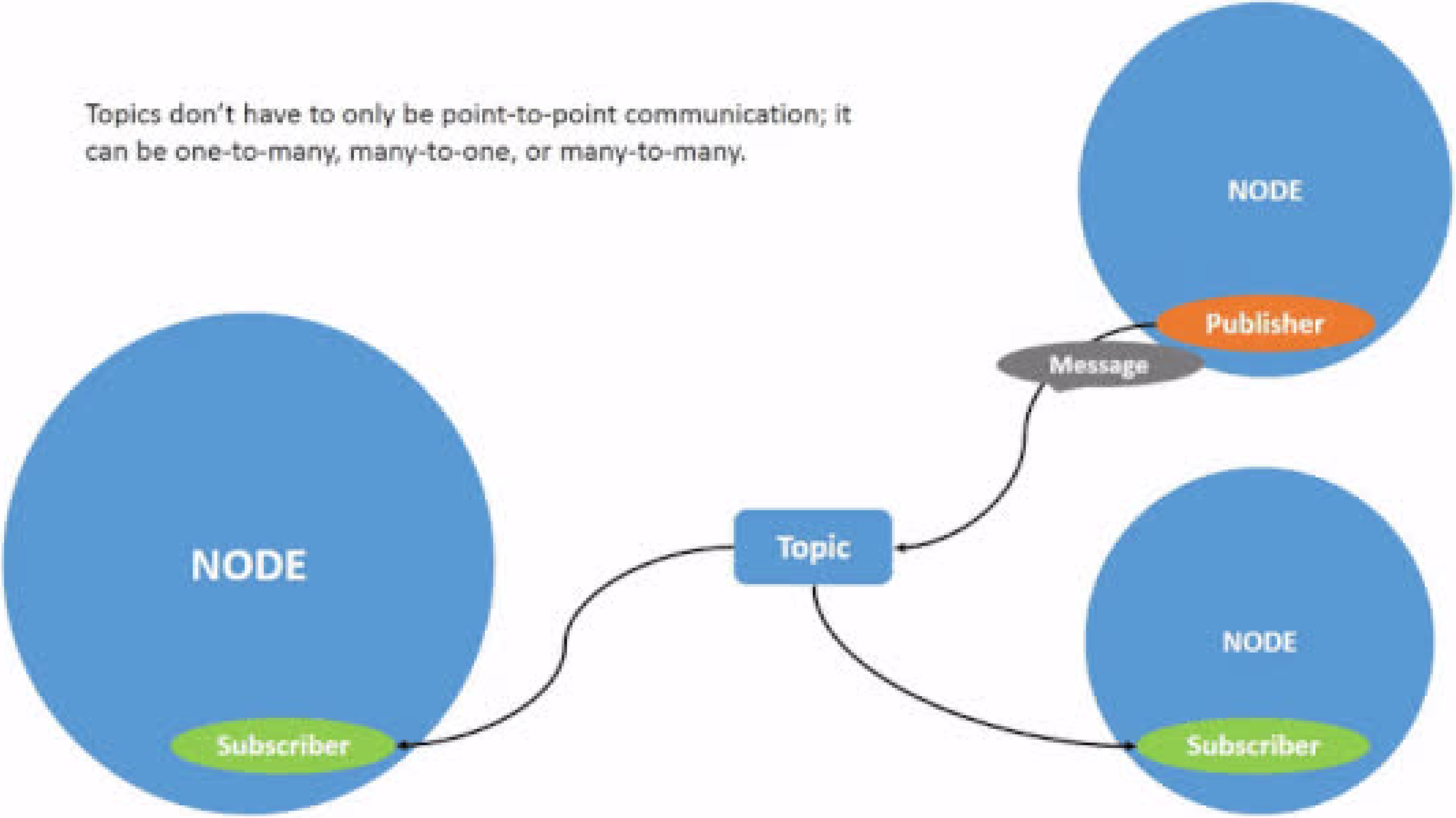
ITS NOT A OS

Its a framework for COMMUNICATION

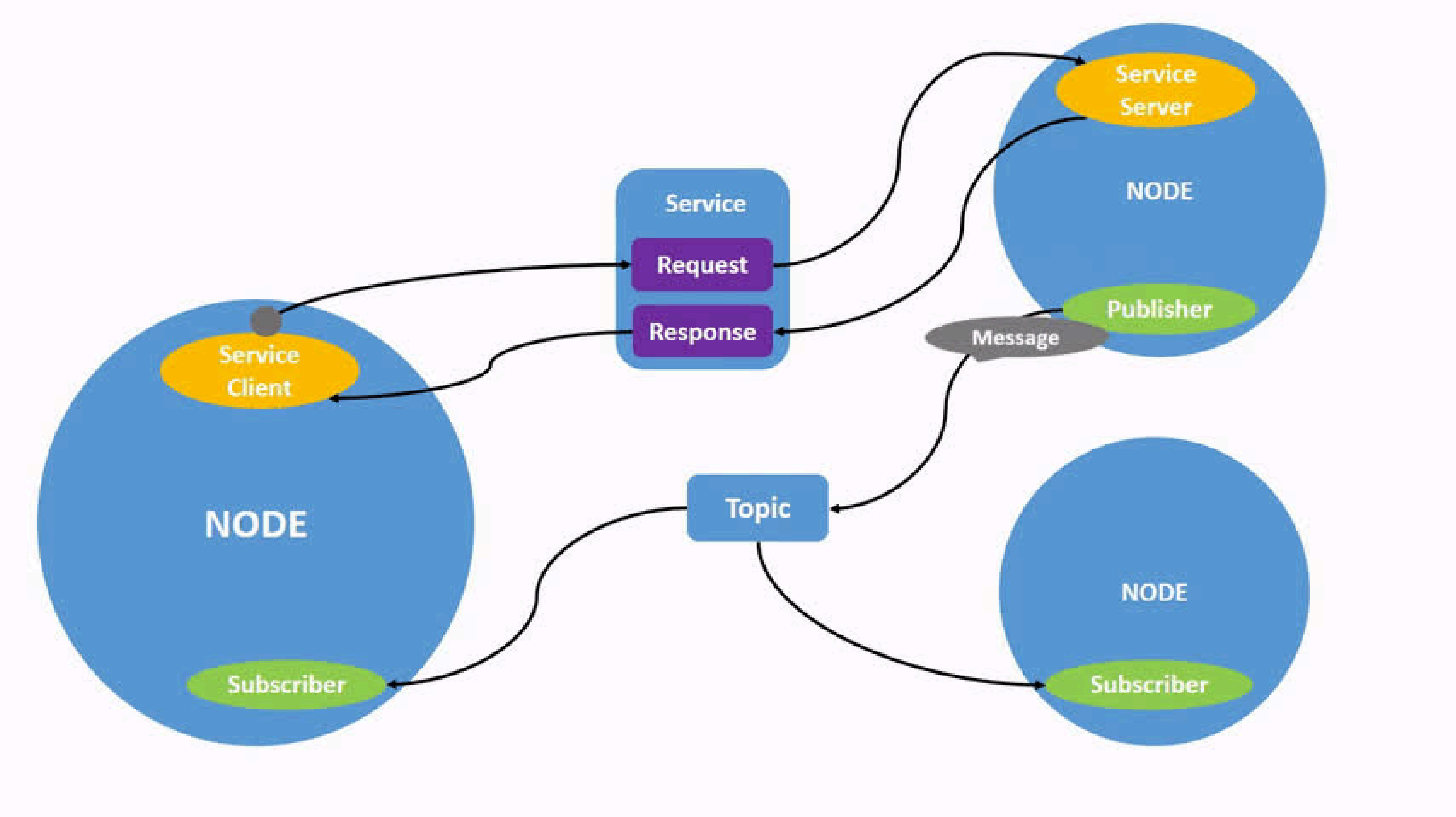
Core concepts

- Nodes: Small units of computation.
- Topics: Channels for communication.
- Messages: Data structures exchanged between nodes.
- DDS: Handles communication, replacing ROS Master.
- Publishers and subscribers

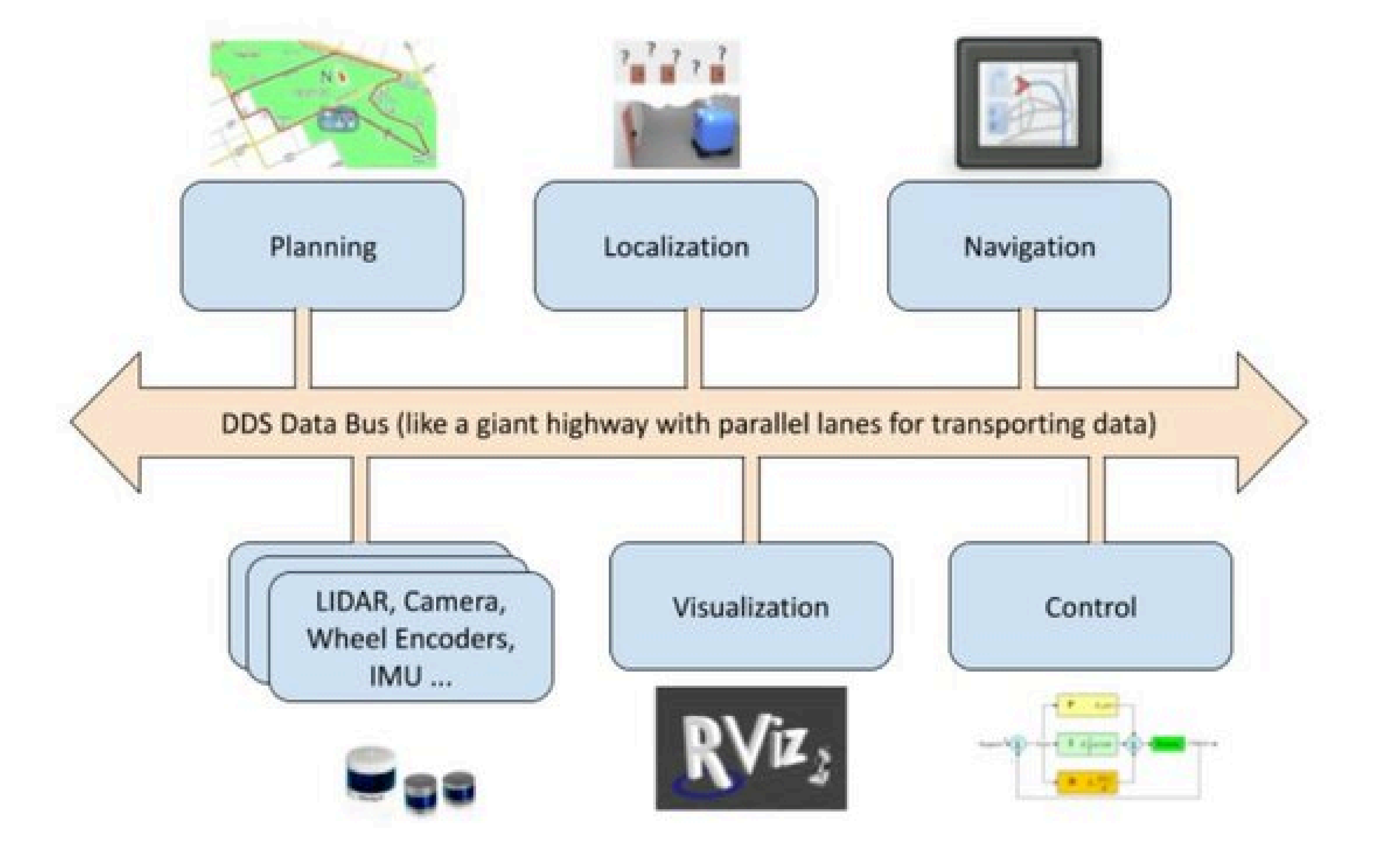
Topics don't have to only be point-to-point communication; it can be one-to-many, many-to-one, or many-to-many.

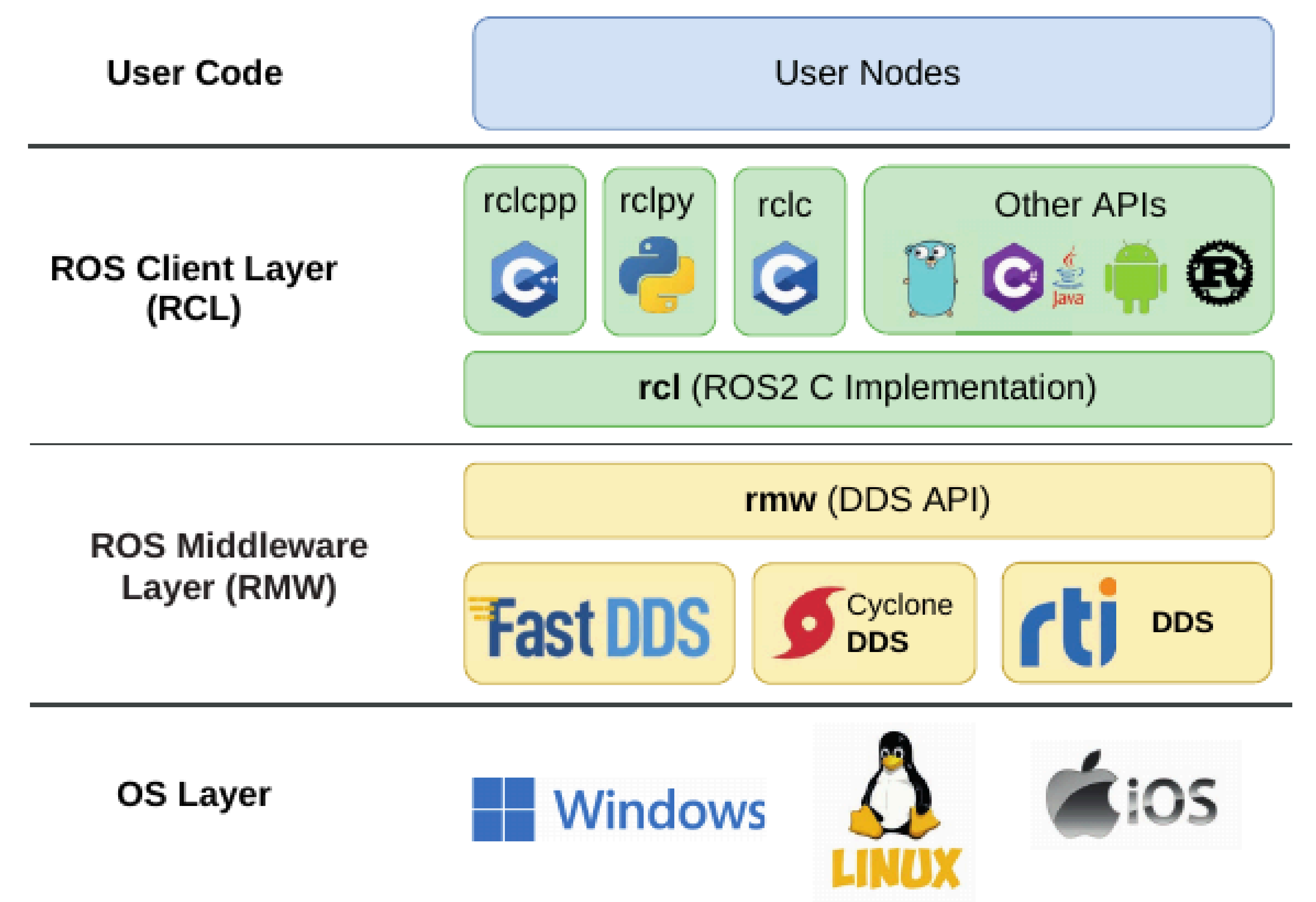


one more ...


Service
Server
NODE
Publisher
Message
Service
Request
Response
Service
Client
NODE
Topic
NODE
Subscriber
Subscriber

ROS 2 Architecture


Planning
Localization
Navigation
DDS Data Bus (like a giant highway with parallel lanes for transporting data)
LIDAR, Camera, Wheel Encoders, IMU ...
Visualization
Control
RViz



Now

lets install...... some stuff (vbox, ubuntu 22.04, ros2 humble tobespecific)

Installation (we’ll be using VM):

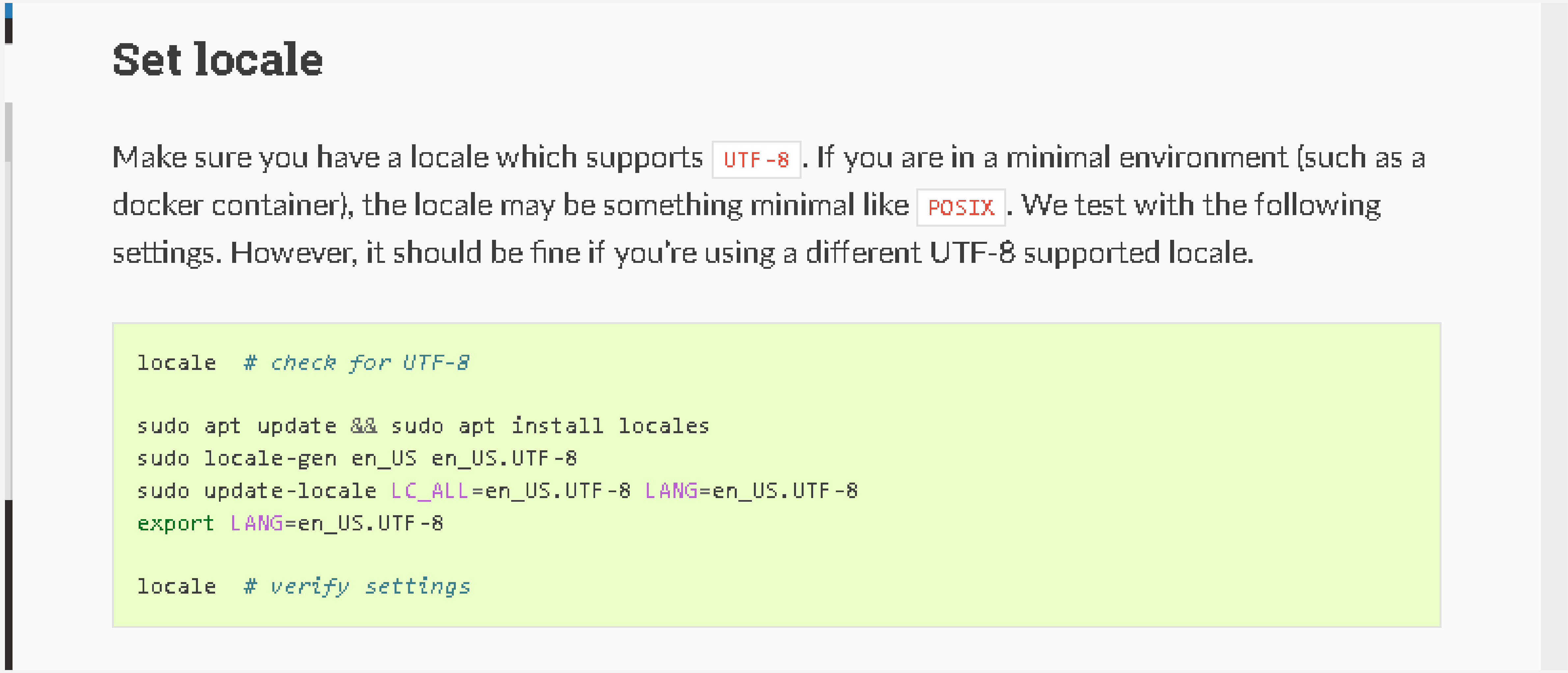
- Vbox:
  - https://www.virtualbox.org/wiki/Downloads

- Get the ubuntu 22.04 desktop image (iso):
  - https://releases.ubuntu.com/jammy/

- Install ubuntu on vbox (give ~30gb disk space and 4cores)
  - Tip: keep the password small and easy (cuz you’ll have to type that so many times)
  - https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview

go to this link and paste all the commands one by one

https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html



Set locale

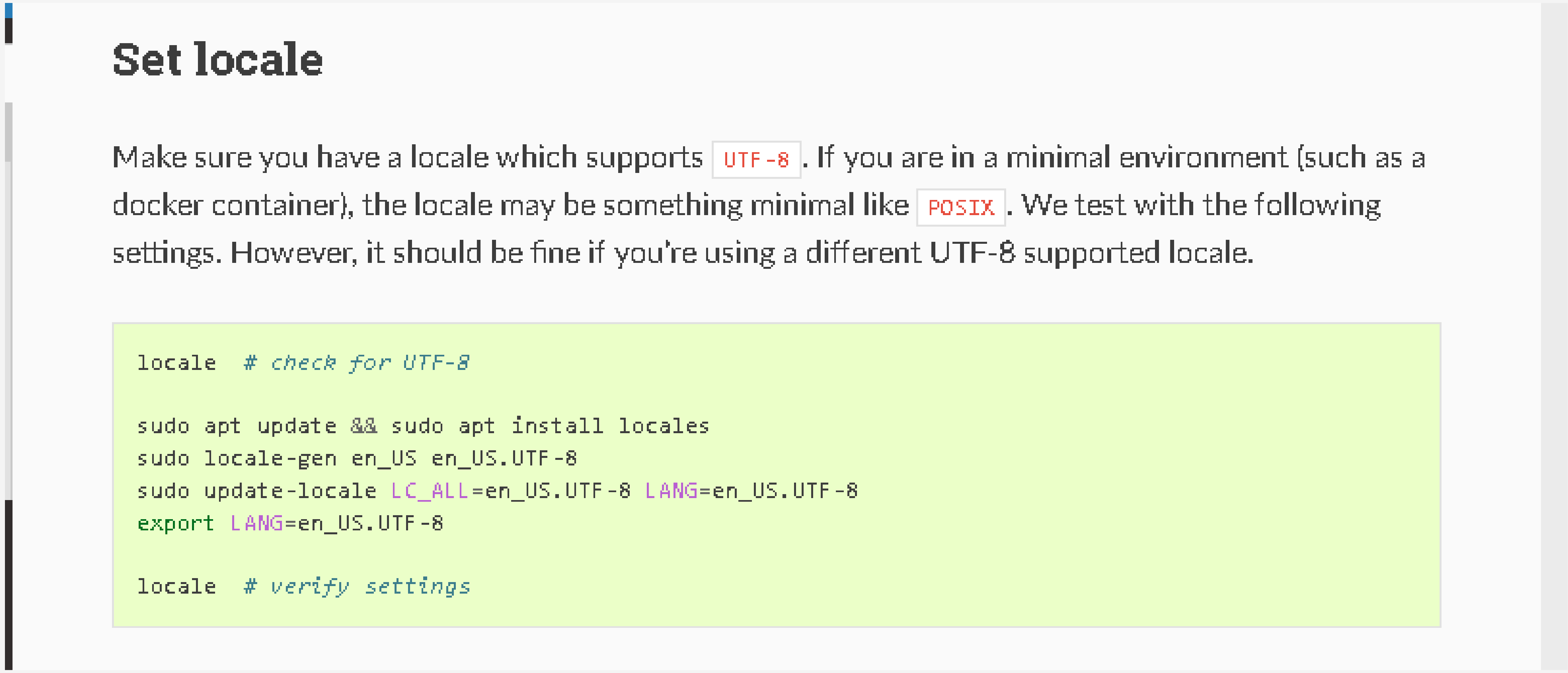
Make sure you have a locale which supports `UTF-8`. If you are in a minimal environment (such as a docker container), the locale may be something minimal like `POSIX`. We test with the following settings. However, it should be fine if you're using a different UTF-8 supported locale.

```
locale  # check for UTF-8

sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8

locale  # verify settings
```

https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html



Set locale

Make sure you have a locale which supports `UTF-8`. If you are in a minimal environment (such as a docker container), the locale may be something minimal like `POSIX`. We test with the following settings. However, it should be fine if you're using a different UTF-8 supported locale.

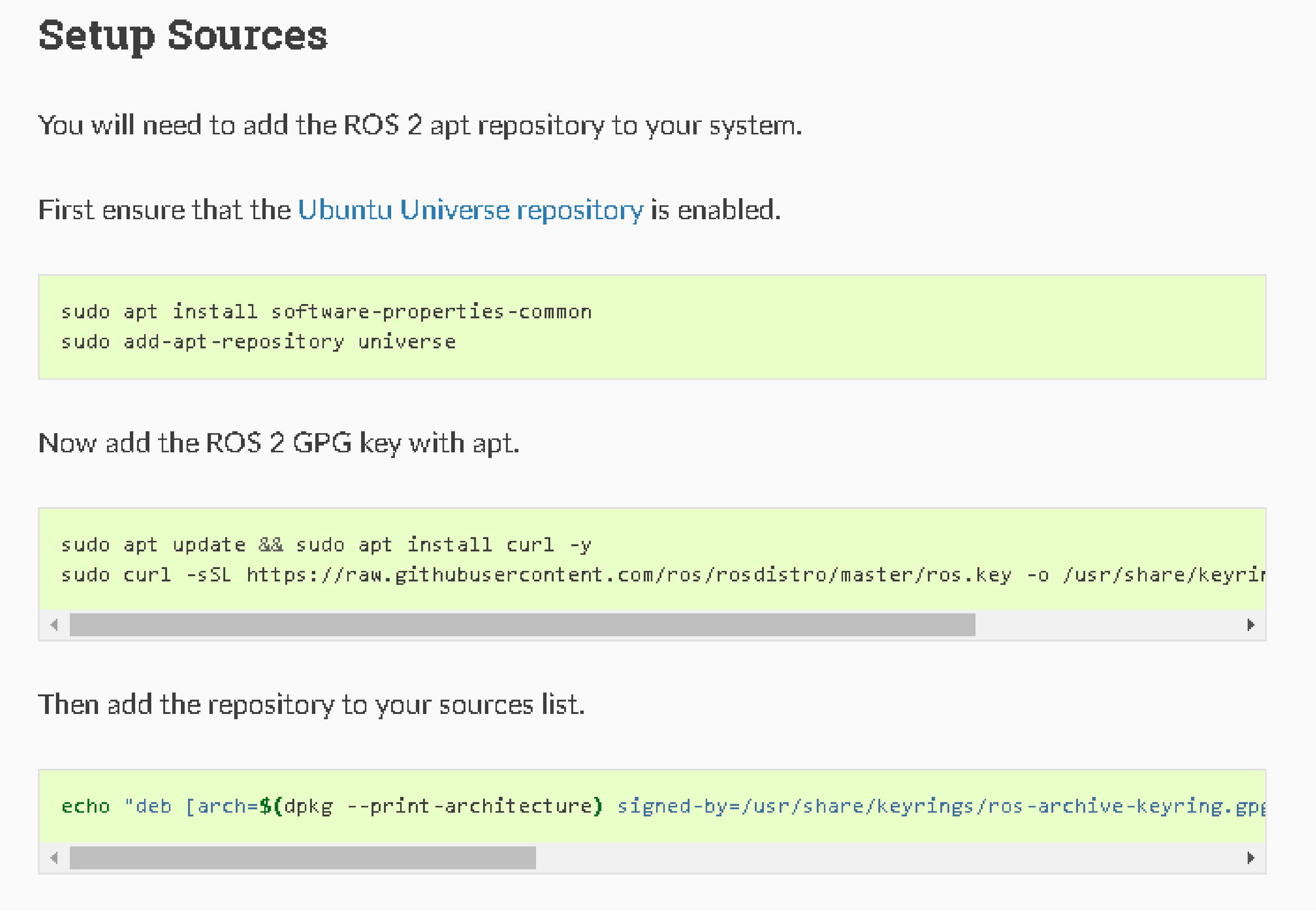
```
locale  # check for UTF-8

sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8

locale  # verify settings
```

(if you get any error change the service provider (jio,vi etc) for this step)

https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html



Setup Sources

You will need to add the ROS 2 apt repository to your system.

First ensure that the Ubuntu Universe repository is enabled.

```
sudo apt install software-properties-common
sudo add-apt-repository universe
```

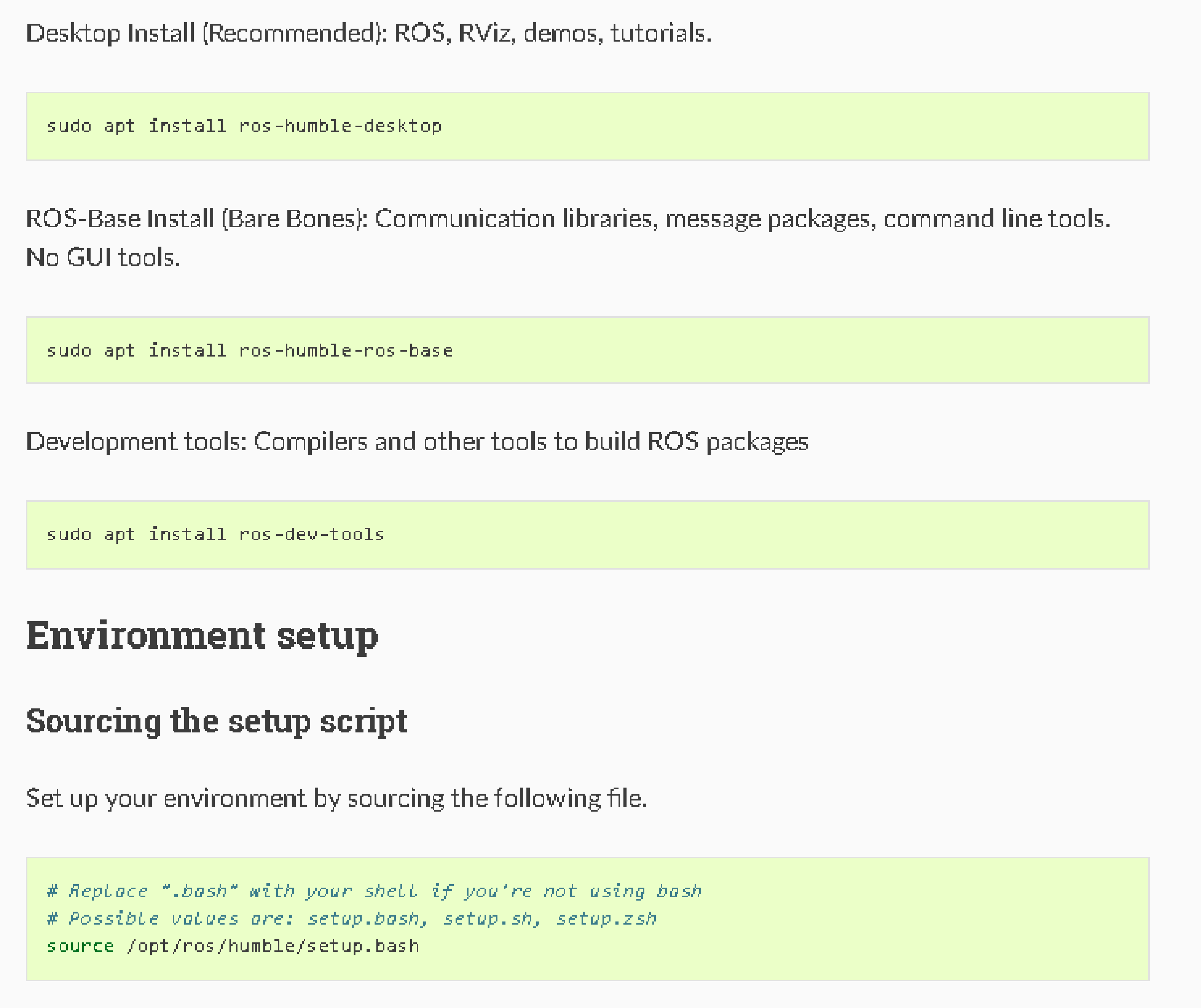
Now add the ROS 2 GPG key with apt.

```
sudo apt update && sudo apt install curl -y
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrin
```

Then add the repository to your sources list.

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gp
```

https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debs.html



Desktop Install (Recommended): ROS, RViz, demos, tutorials.

```
sudo apt install ros-humble-desktop
```

ROS-Base Install (Bare Bones): Communication libraries, message packages, command line tools. No GUI tools.

```
sudo apt install ros-humble-ros-base
```

Development tools: Compilers and other tools to build ROS packages

```
sudo apt install ros-dev-tools
```

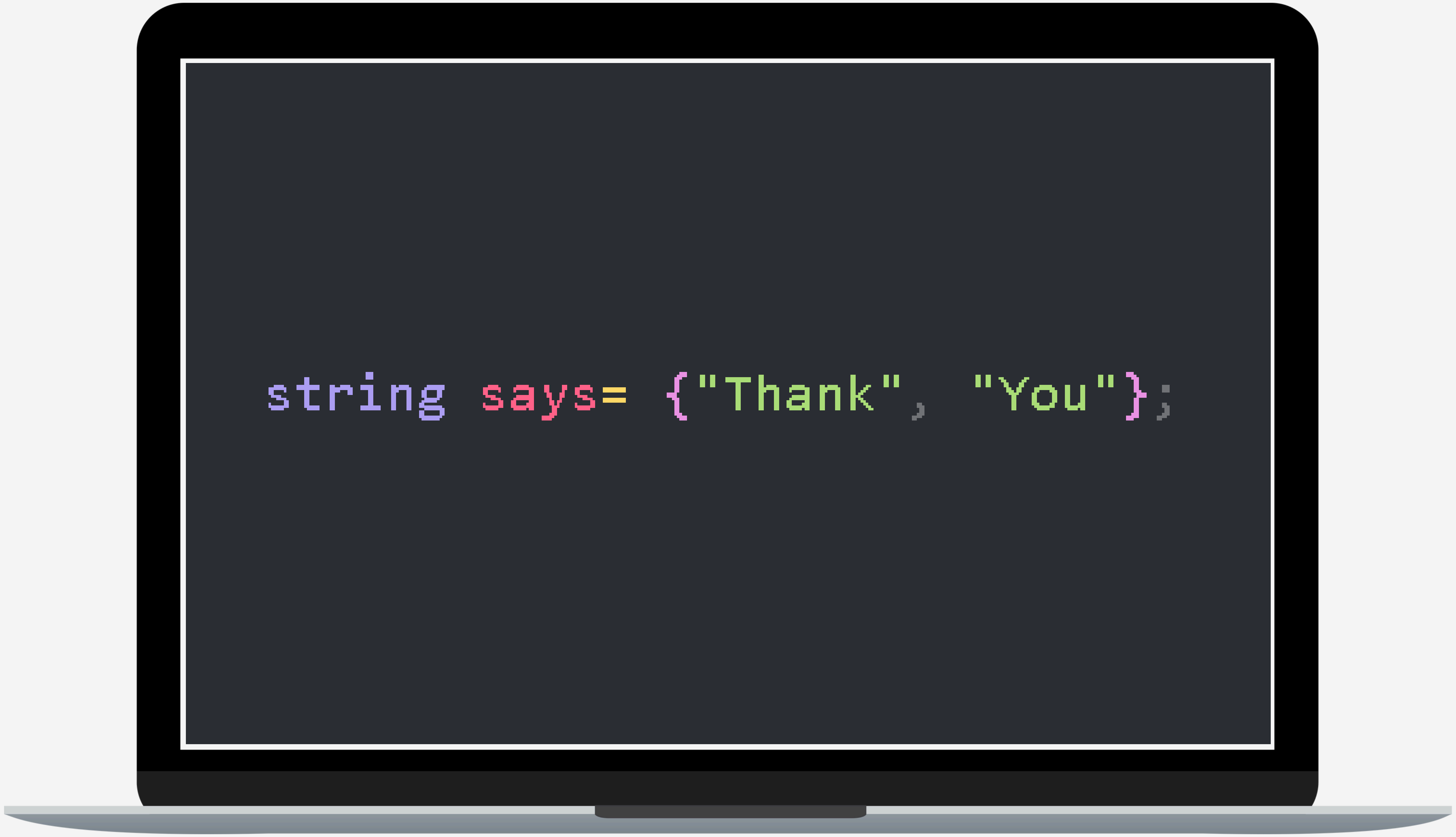
Environment setup

Sourcing the setup script

Set up your environment by sourcing the following file.

```
# Replace ".bash" with your shell if you're not using bash
# Possible values are: setup.bash, setup.sh, setup.zsh
source /opt/ros/humble/setup.bash
```

TEAM QUARK



```
string says= {"Thank", "You"};
```