

# 1. Introduction to ROS2

---

## 1. Overview of ROS2

---

ROS2 is the second generation Robot Operating System, an upgraded version of ROS1, which solves some problems of ROS1. The earliest version of ROS2, Arden, was in 2017. With the iteration of the version, it has been continuously updated and optimized, and now it has a stable version. Compared with ROS1, the choice of Linux version and ROS2 version is also related. The corresponding versions of the two are as follows:

ROS2 version	Ubuntu version
Foxy	Ubuntu20.04
Galactic	Ubuntu20.04
Humble	Ubuntu22.04

According to your own Linux version, download the corresponding ROS2 version. This product course is based on the Foxy version.

## 2. ROS2 features

---

### 2.1. ROS2 fully supports three platforms

- Ubuntu
- Mac OS X
- Windows 10

### 2.2. Implemented distributed architecture

Cancel the Master central node to realize distributed discovery, publish/subscribe, and request/response communication of nodes.

### 2.3. Support real-time

### 2.4. Use new versions of programming languages

- C++11
- Python3.5+

### 2.5. Using the new compilation system Ament (ROS is Catkin)

### 2.6. ROS1 can communicate with ROS 2 through rosbridge

## 3. The difference between ROS2 and ROS1

---

### 3.1. Platform

ROS1 currently only supports running and using in Linux systems, and is most commonly built and used in Ubuntu. ROS2 can currently be built and used on Ubuntu, Windows and even embedded development boards, and the platform is more extensive.

## 3.2.Language

- C++

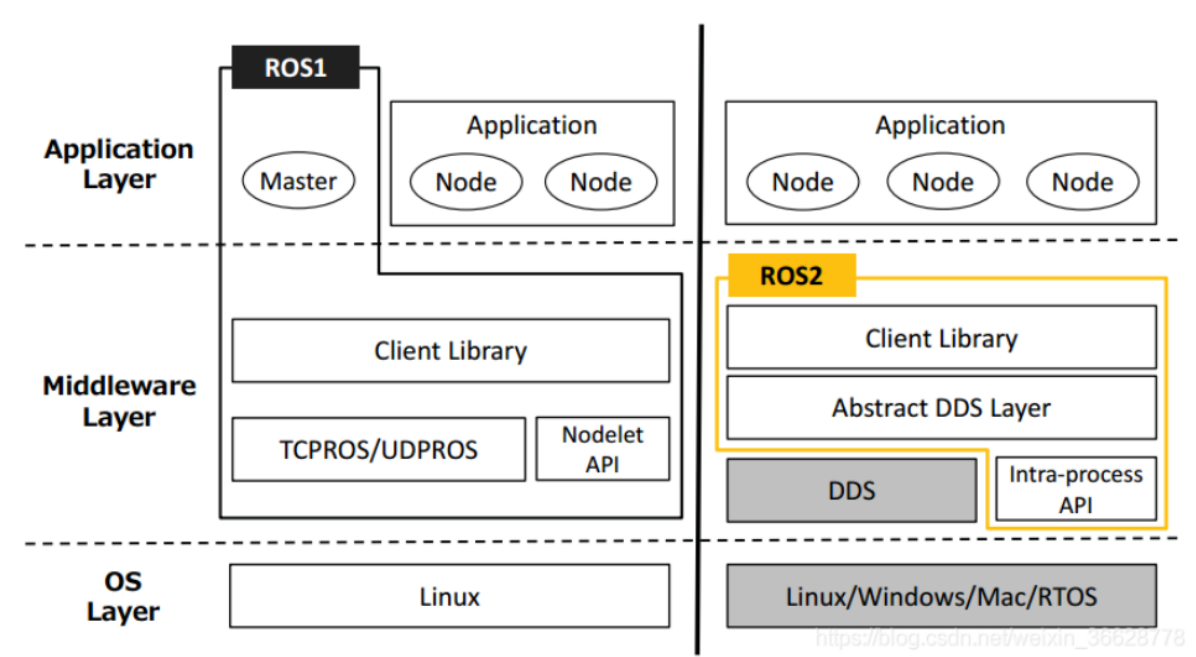
ROS1 is C++03 at its core, while ROS2 makes extensive use of C++11.

- Python

The Python version used by ROS1 is Python2, while the Python version used by ROS2 is at least 3.5, and the Python version used by Foxy is 3.8.

## 3.3. Middleware

Roscore needs to be turned on before starting ROS1. This master controls the communication between all nodes, but ROS2 does not. It only has an abstract middleware interface through which data is transmitted. Currently, all implementations of this interface are based on the DDS standard. This enables ROS 2 to provide a variety of high-quality QoS service policies to improve communication across different networks.



## 3.4. Compile command

The compilation command of ROS1 is `catkin_make`, while the compilation command of ROS2 uses the `colcon build` command.