

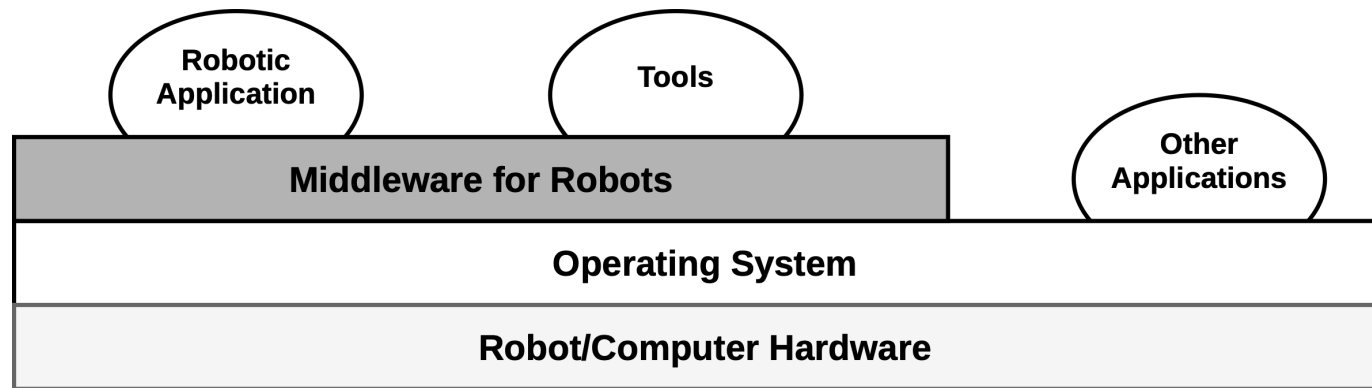
ROS 2 : Introducción

Eugenio Ivorra

2023-2024



- Los robots deben ser programados para ser útiles.
- Necesitamos intermediarios de software (Middlewares).
- Los middlewares para la programación de robots ofrecen controladores, bibliotecas y metodologías.
- Pocos de estos middlewares han sobrevivido al robot para el que fueron diseñados o han trascendido los laboratorios donde se implementaron.
- La gran diferencia la marca la comunidad global de desarrolladores de ROS (Sistema Operativo para Robots).



- ROS es el acrónimo de Robot Operating System
- ROS y ROS2
- Muchos tutoriales, documentación y desarrolladores
- Usaremos Ubuntu 22.04 y ROS2 Humble
- <https://docs.ros.org/en/humble/index.html>

 ROS 2[™]

Elementos clave:

WORKSPACE: el conjunto de software instalado en el robot o en la computadora, los programas que desarrolla el usuario y las herramientas para compilar y lanzar

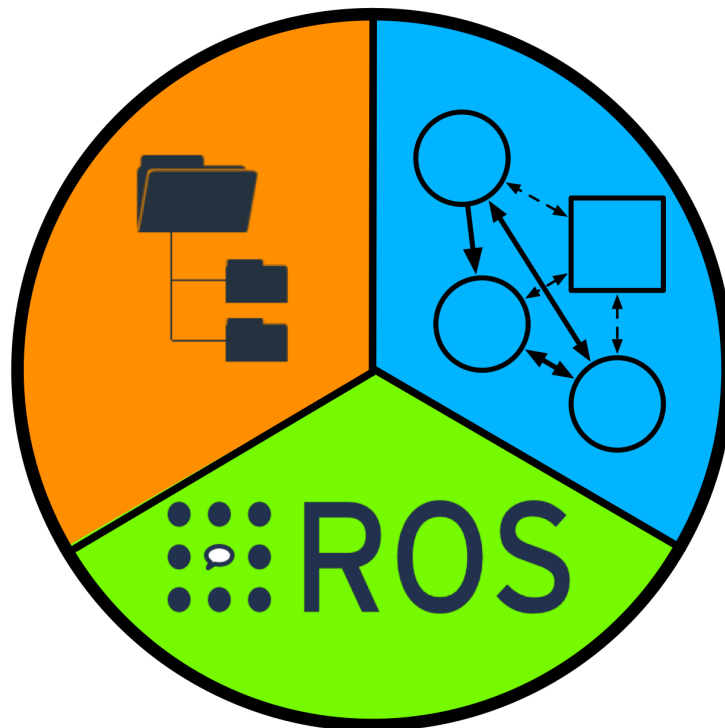


Gráfico Computación:
La aplicación de ROS2 funcionando

Comunidad y openSource: Amplia comunidad de desarrolladores que contribuyen a través de repositorios públicos



El software de un robot se asemeja a un gráfico de computación compuesto por nodos ROS2 que se comunican entre sí para permitir al robot realizar tareas. La lógica de la aplicación se encuentra en los nodos, que son los elementos principales de ejecución en ROS2.

Los mecanismos de comunicación en ROS2 son:

- Publicación/Suscripción:** Asíncrona N:M

- Servicios:** Síncrona 1:1

- Acciones:** Asíncrona 1:1

- Publicación/Suscripción:** Un nodo publica mensajes en un tema y otros nodos pueden suscribirse a ese tema para recibir los mensajes. Este es un mecanismo de comunicación asíncrona, ya que el nodo que publica el mensaje no necesita esperar a que los nodos suscritos lo reciban.

- Servicios:** Un nodo solicita un servicio a otro nodo y el nodo que proporciona el servicio responde con un resultado. Este es un mecanismo de comunicación síncrona, ya que el nodo solicitante debe esperar a recibir el resultado del nodo que proporciona el servicio antes de continuar.

- Acciones:** Un nodo solicita una acción a otro nodo y el nodo que realiza la acción envía información sobre el progreso de la acción y el resultado final. Este es un mecanismo de comunicación asíncrona, ya que el nodo solicitante puede continuar con otras tareas mientras el nodo que realiza la acción completa la tarea solicitada.



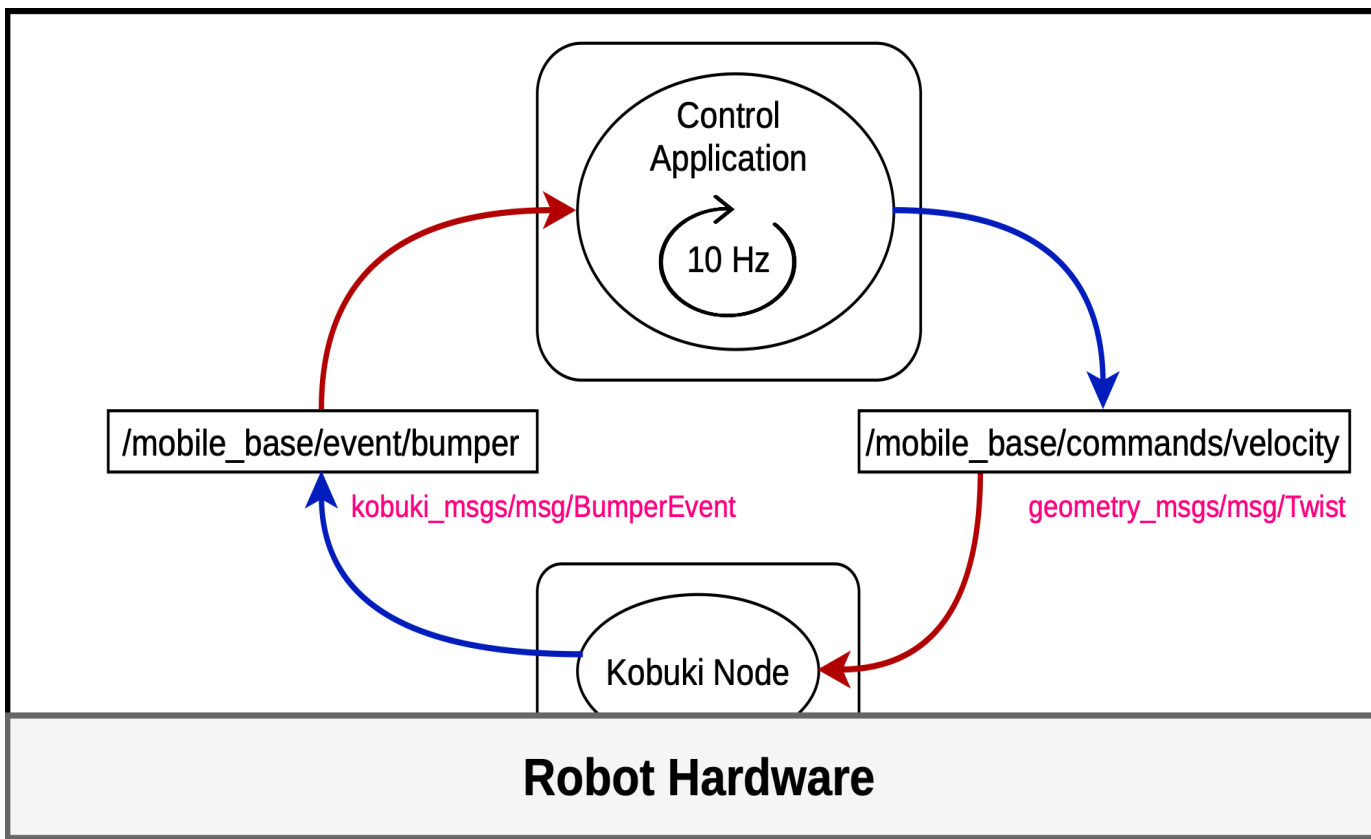
Convención de nomenclatura

Los nombres de los recursos de ROS2 dependen de:

- El tipo de recurso: relativo, absoluto o privado
- El nombre del nodo
- El namespace

name	Result: (node: my_node / ns: none)	Result: (node: my_node / ns: my_ns)
my_topic	/my_topic	/my_ns/my_topic
/my_topic	/my_topic	/my_topic
~my_topic	/my_node/my_topic	/my_ns/my_node/my_topic





node name

ROS2 Node



Publication

Process

Process



Subscription

topic name

Topic



Action

topic msg type

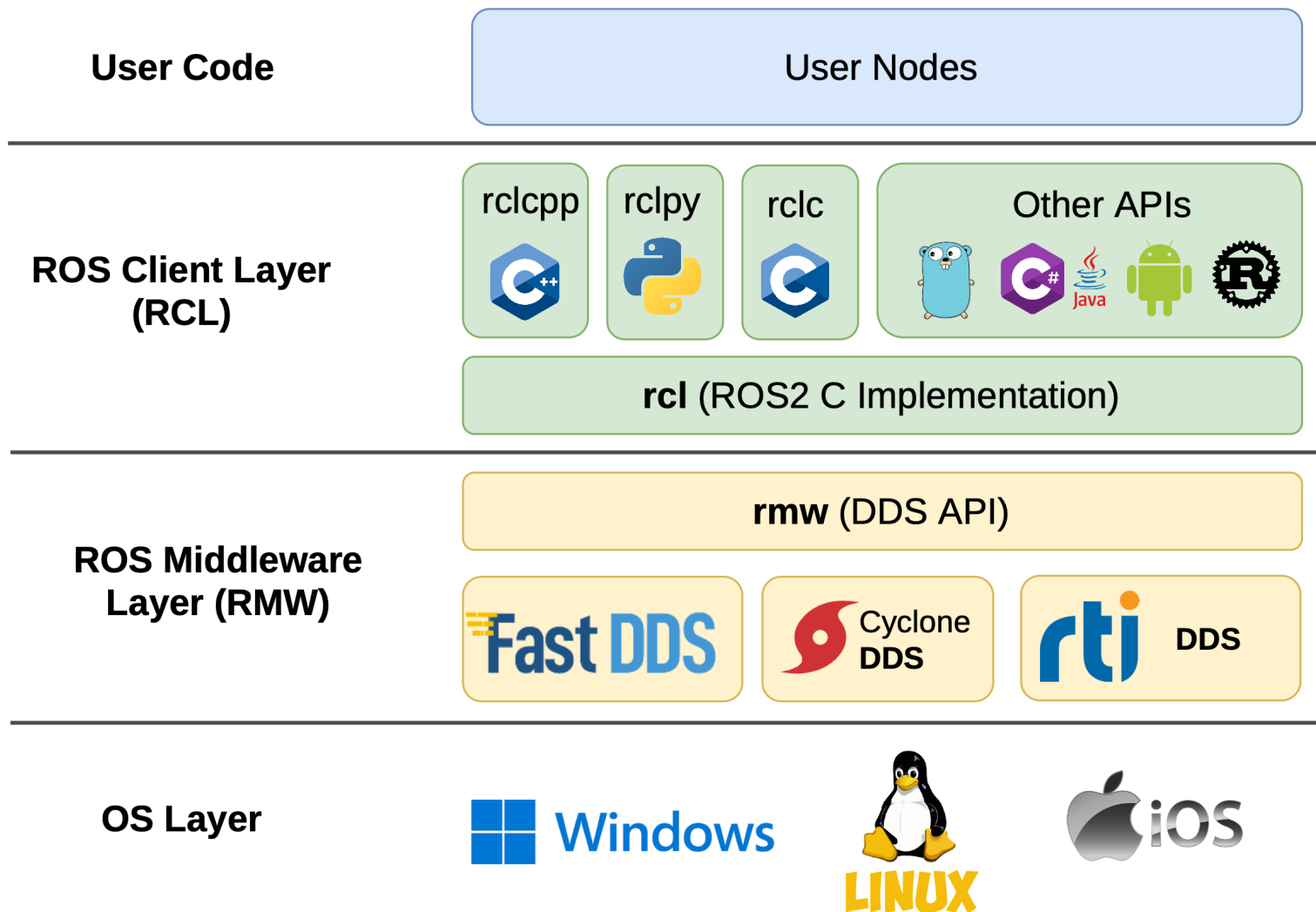


El software ROS2 se puede abordar desde un punto de vista estático, considerando dónde se instala y organiza, y todas las herramientas y procesos que permiten lanzar un gráfico de computación. Esto incluye el sistema de compilación y las herramientas de inicio de nodos.

Los elementos principales del software ROS2 desde un punto de vista estático son:

- **Paquete:** Es el conjunto funcional mínimo de software. Contiene ejecutables, bibliotecas o definiciones de mensajes con un propósito común.
- **Workspace:** Es un directorio que contiene paquetes. Se puede activar para que estén disponibles para su uso.
- **Underlay y overlay:** Son mecanismos para agrupar y organizar paquetes. **Underlay** proporciona una capa base de paquetes que se pueden utilizar en varios espacios de trabajo, mientras que **overlay** proporciona una capa de paquetes que se puede superponer a **underlay** para personalizar un espacio de trabajo específico.





RICO, Francisco Martín. *A concise introduction to robot programming with ROS2*. CRC Press, 2022.

<https://docs.ros.org/en/humble/index.html>

