µROS

New client library and middleware features

Ralph Lange (Robert Bosch GmbH, Corporate Research)
Maria Merlan (eProsima)

October 2021

Overview and Mission

Embedded world: Networks of µCs and larger processors

(XRCE) eXtremely Resource-Constrained Environments

Lack of common standard dev framework and OS

micro-ROS bridges this gap for ROS 2

µROS

FULL PORTABILITY

Any RTOS and Bare metal Library Generator!

Any low-mid range MCU!

Typical features:

~ 150 KB of flash memory

> 25 KB of RAM memory

General purpose input/output pins

Peripherals: GPIO, USB, Ethernet, SPI, UART, I2C, CAN, etc

REFERENCE HW

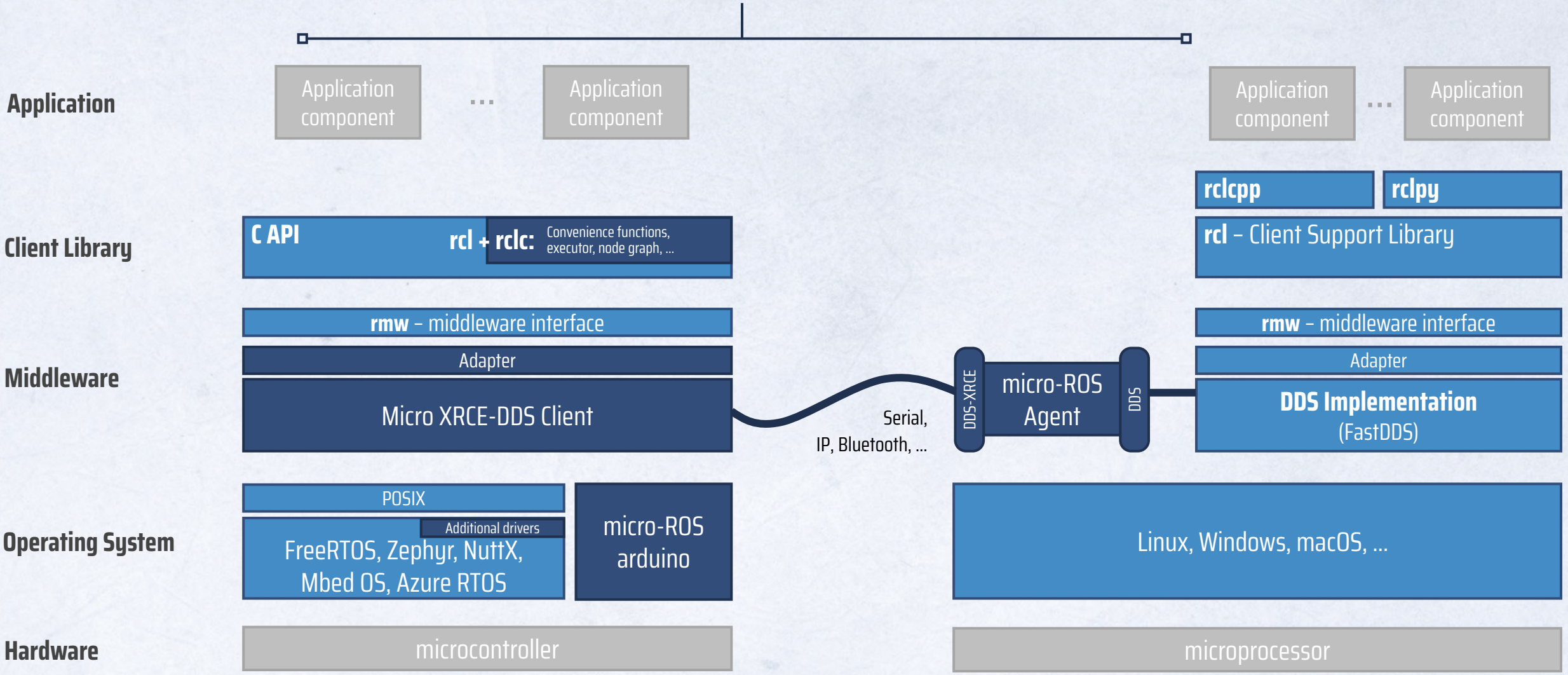
- **Renesas RA6M5**
- **Arduino**
- **Raspberry Pi Pico**
- **Arduino Nano RP2040 Connect** 1st Arduino with Raspberry Pi silicon
- **ESP-IDF v4.3 & ESP32-S2/C3**
- **Teensy 3.2 / 3.5 / 4.1 / 4.2**
- **OpenCR support**
- **STM32CubeMX & STM32CubeIDE**
- **Crazyflie 2.1 drone, ...**

REFERENCE RTOS

- **Mbed RTOS 6.8 / 6.9 / 6.10**
- **FreeRTOS**
- **NuttX 10.0 / 10.1**
- **Zephyr RTOS 2.4 / 2.5**
- **Azure RTOS ThreadX**

Check full list of supported HW & RTOS
https://micro.ros.org/docs/overview/hardware/

Architecture



Build Systems

For your ROS tooling:

```
$ git clone github.com/micro-ROS/micro_ros_setup.git
$ colcon build

$ ros2 run micro_ros_setup
    ↳  create_firmware_ws.sh [RTOS] [MCU]
    ↳  configure_firmware.sh [MY_APP] -t [TRANSPORT]
    ↳  build_firmware.sh
    ↳  flash_firmware.sh

$ ros2 run micro_ros_setup
    ↳  create_agent_ws.sh
    ↳  build_agent.sh
    ↳  micro_ros_agent [TRANSPORT]
```

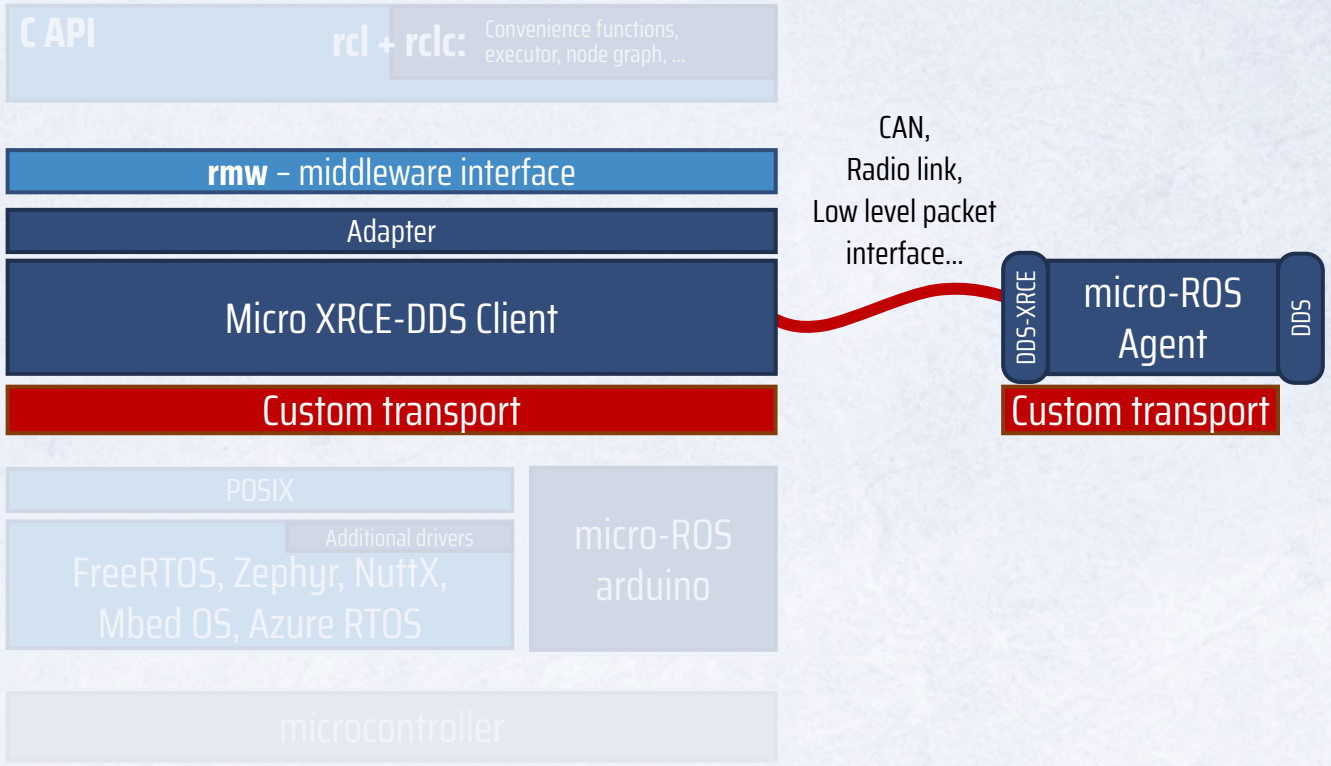
For your favorite tool:

- Renesas e2 studio
- STM32CubeMX/IDE
- RasPi Pico SDK
- Arduino
- ESP32 IDF
- Zephyr

µROS

NEW FEATURES

Custom Transports

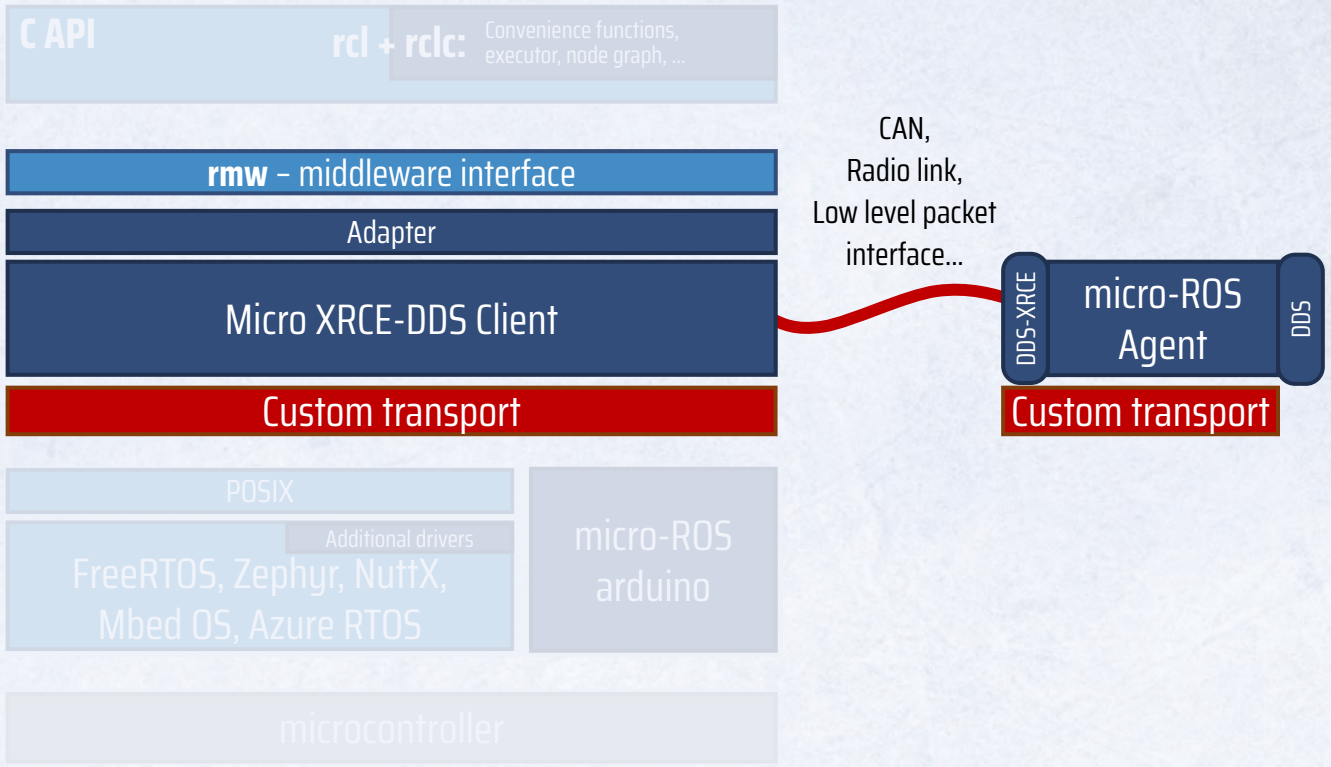


API for micro-ROS Agent and Client based on just four callbacks:

```
rmw_uros_set_custom_transport(
    true, // Framing enabled here.
    (void *) &args,
    my_custom_transport_open,
    my_custom_transport_close,
    my_custom_transport_write,
    my_custom_transport_read
);
```

- Packet or stream mode based on HDLC
- Custom arguments

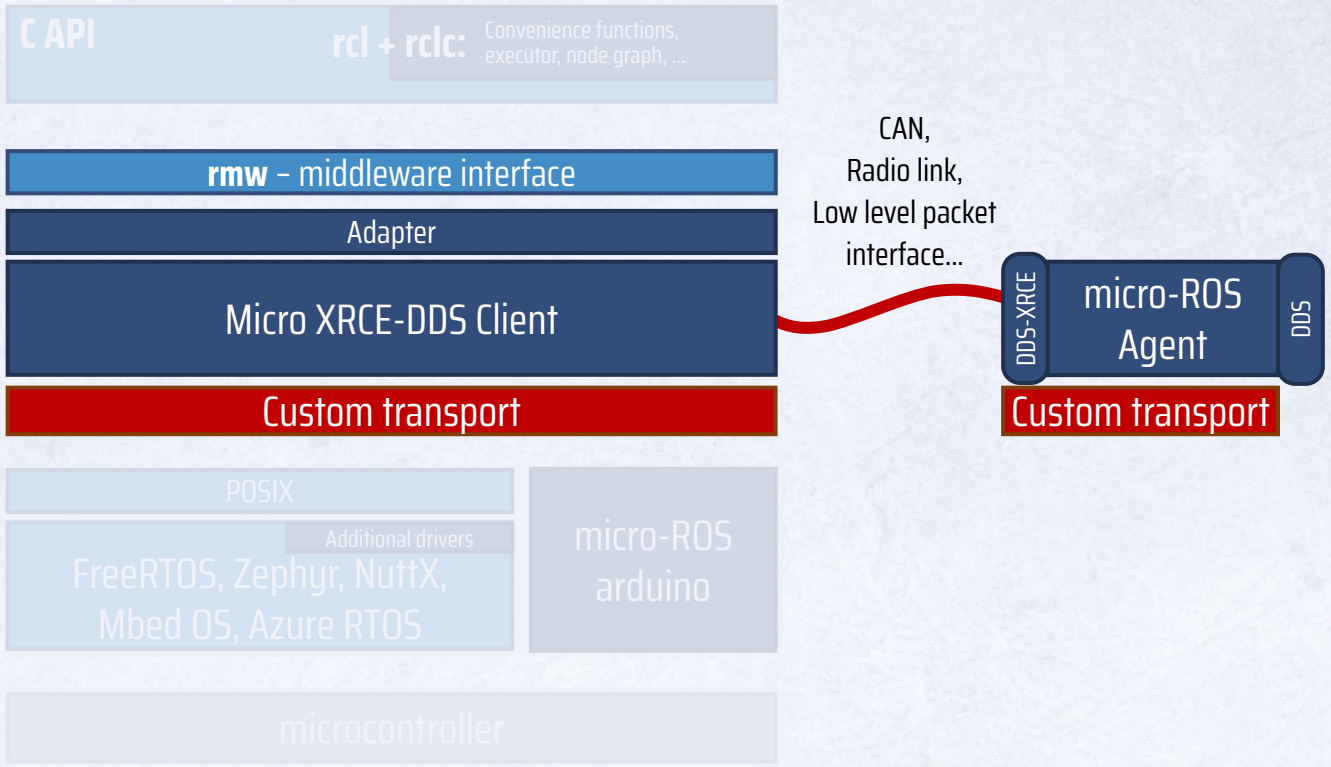
Custom Transports



Micro XRCE-DDS cares for error handling:

- Custom transport does not need to provide
  (1) reliability, i.e., messages may be dropped
  (2) ordering, i.e., messages may arrive out of order
  (3) notification of dropped messages
- Details in Section 11.1 of DDS-XRCE standard at www.omg.org/spec/DDS-XRCE/

Custom Transports



Example in Crazyflie demo

- github.com/micro-ROS/freertos_apps/↵ microros_crazyflie21_extensions/src/↵ microros_transports.c

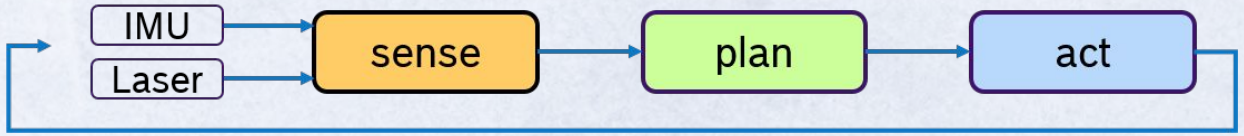
Documentation at micro.ros.org

- Tutorials → Advanced → Creating custom micro-ROS transports

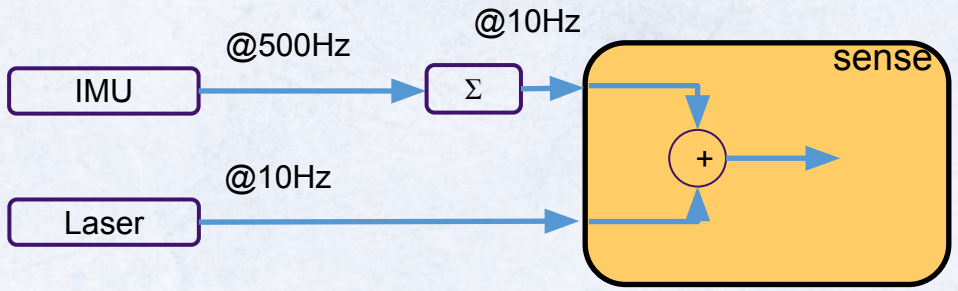
Executor Worker Concept

Extended API for typical patterns

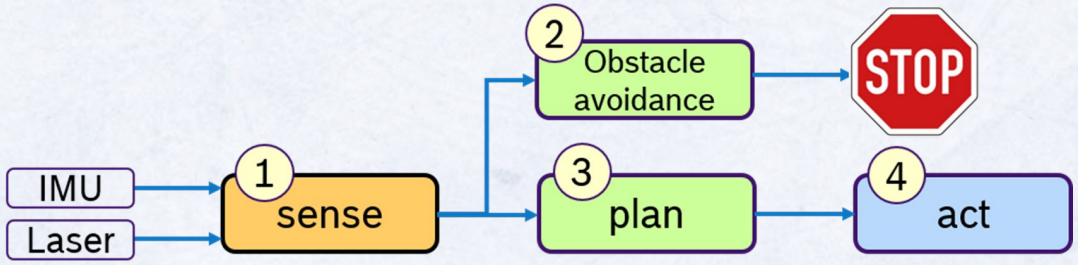
- Control loops



- Data fusion

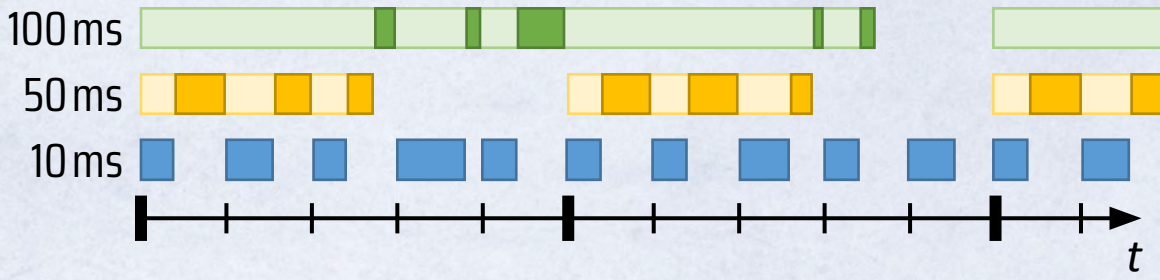


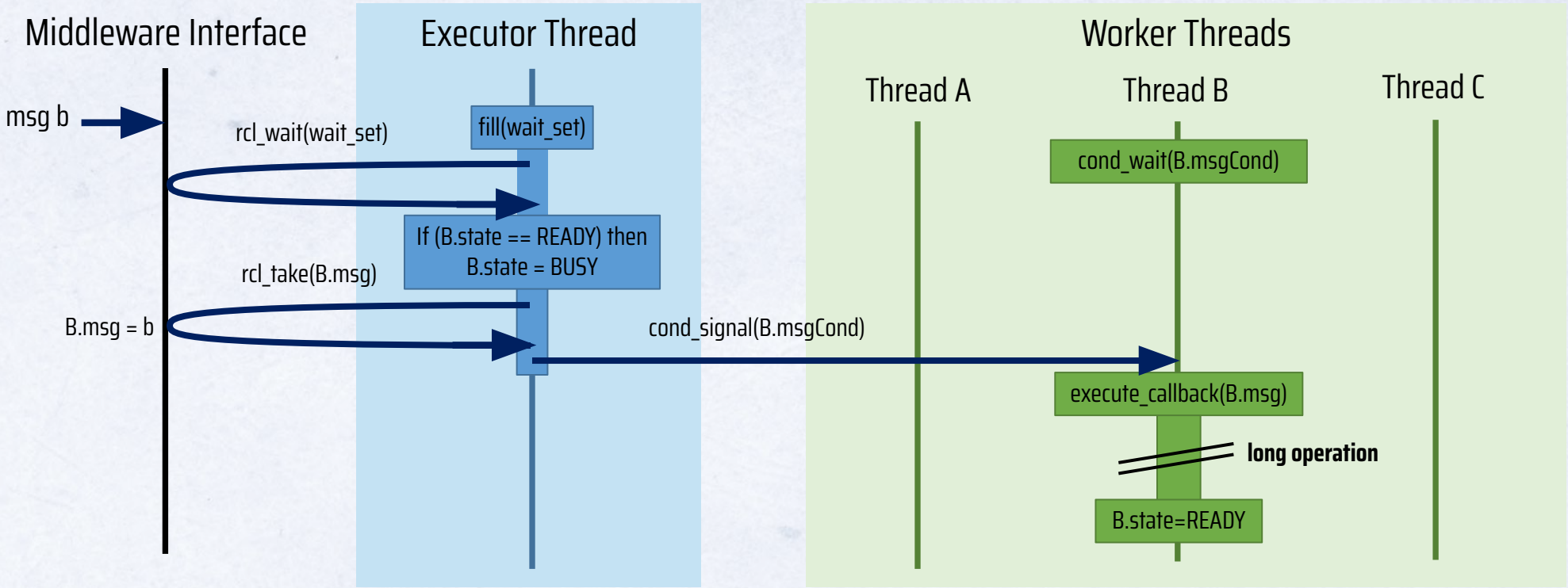
- Prioritized paths



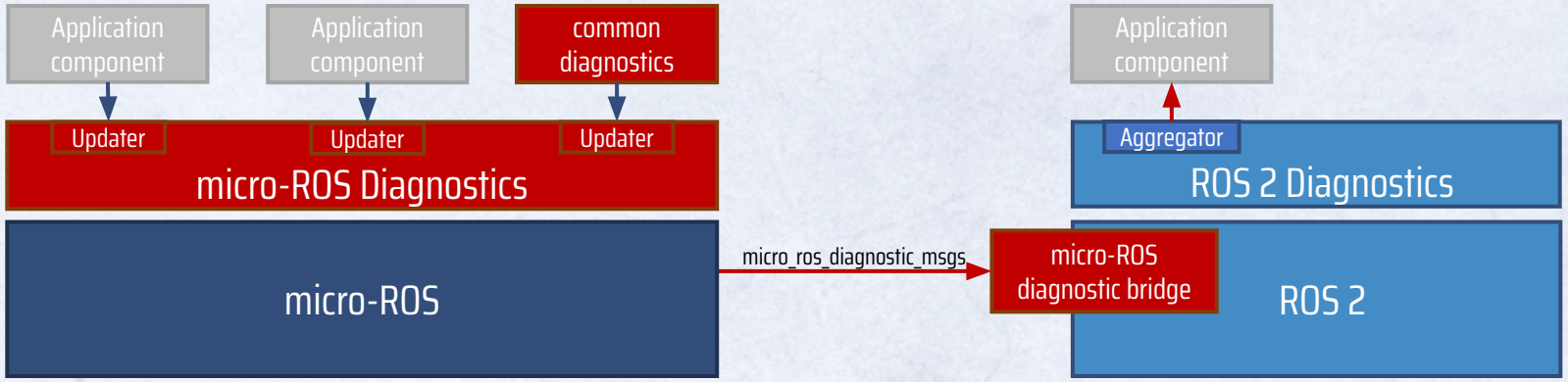
Executor Worker Concept

Flexible concept for use with rate-monotonic or reservation-based scheduling





Diagnostics



Important features

- Optimized message types
- Updater implementation for rcl+rclc

Source code at github.com/micro-ROS/micro_ros_diagnostics

- Contributions welcome. Reach out to us!

More New Features

Hardware support: **RasPi Pico · Teensy · Renesas EK RA6M5**

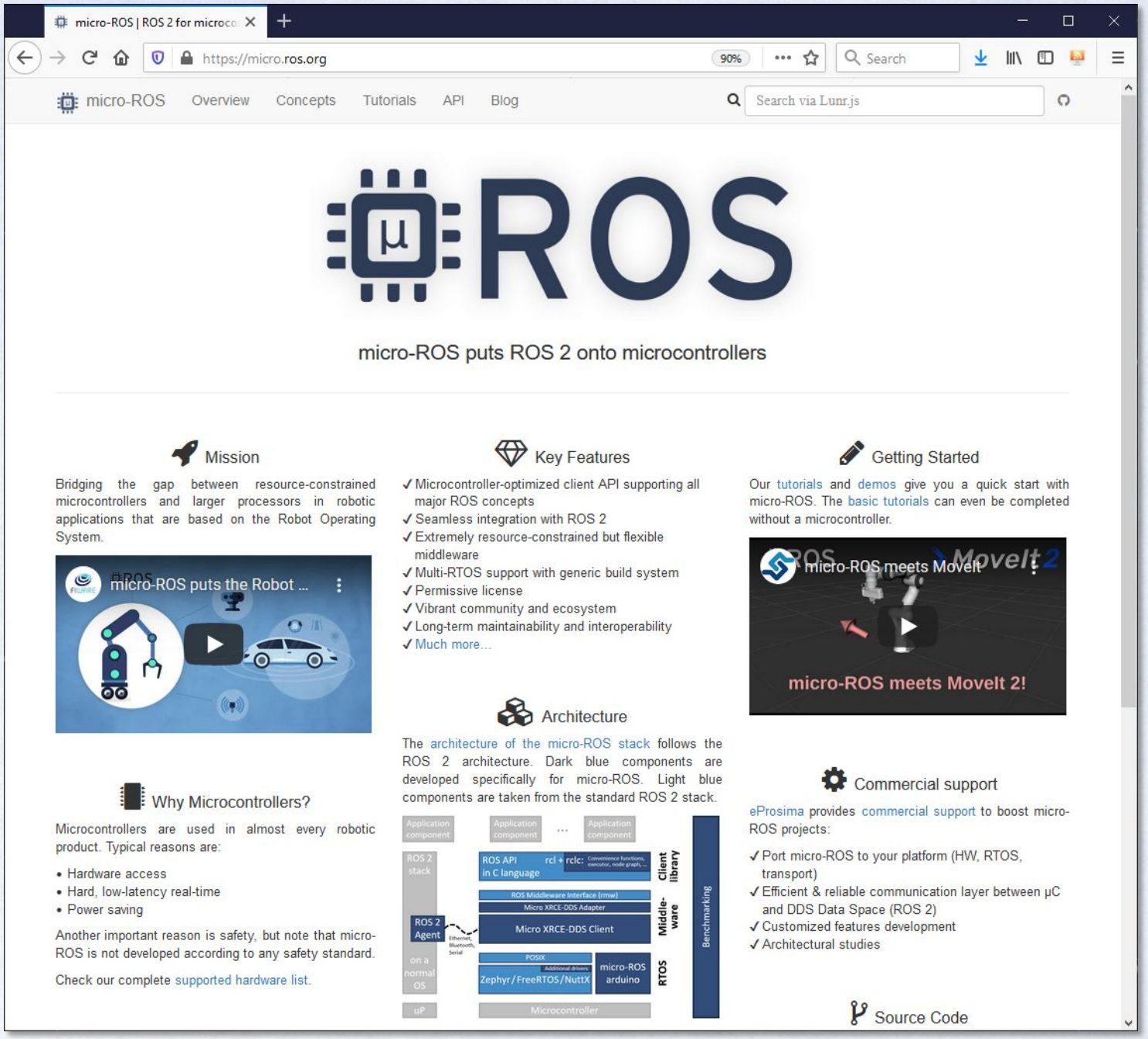
RTOS: **bare metal (Arduino) · Mbed OS · Azure RTOS ThreadX**

Middleware: **static memory pools · thread-safety · embedded RTPS RMW beta support**

Client library: **parameters · node lifecycle · ROS graph · services**

micro.ros.org
micro-ROS at GitHub
Features
SW and HW support
Advanced tutorials with examples
Articles
Demo videos

+ Join the Embedded WG!



The OFERA project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 780785