micro-ROS: bringing ROS 2 to MCUs

Francesca Finocchiaro - eProsima

November 12th, 2020

Overview

Who are we?

OFERA

µROS

EPROSIMA
The Middleware Experts

FIWARE
FOUNDATION

BOSCH

PIAP

funded by European Commission

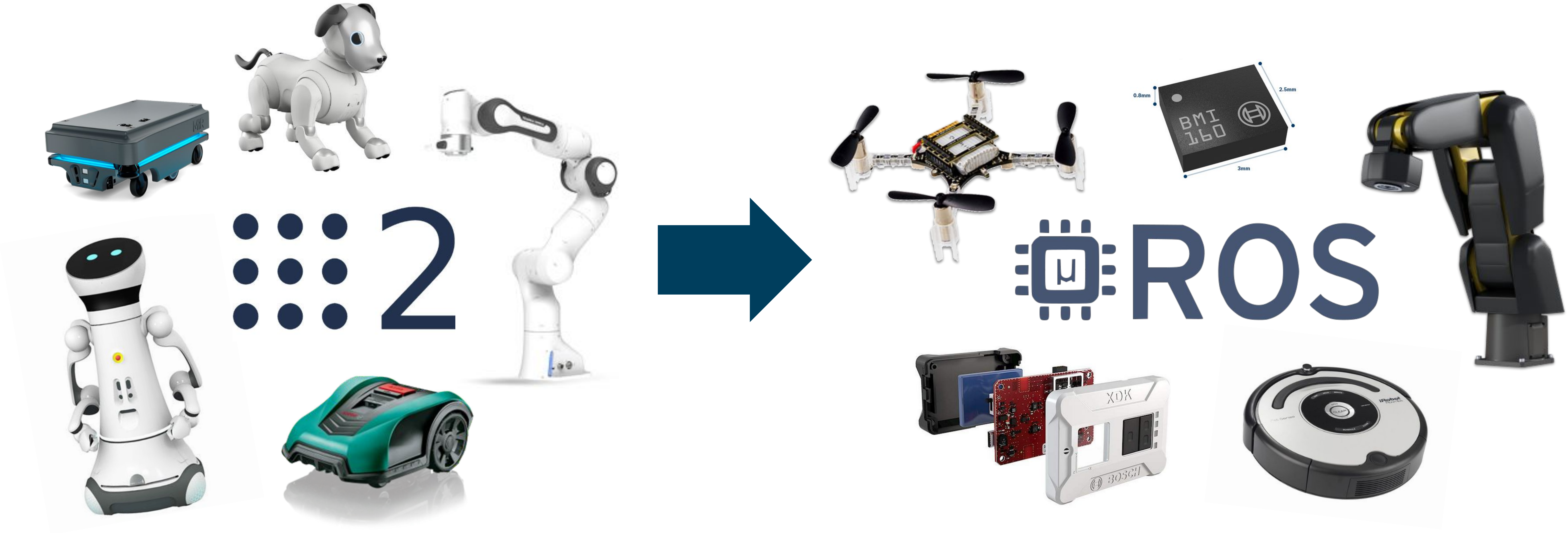
***Open-source** project,*
now benefiting from a huge
participation from a growing
community!

https://micro-ros.github.io/
https://www.eprosima.com/
francescafinocchiaro@eprosima.com

Why micro-ROS?

micro-ROS: puts ROS 2 onto microcontrollers!

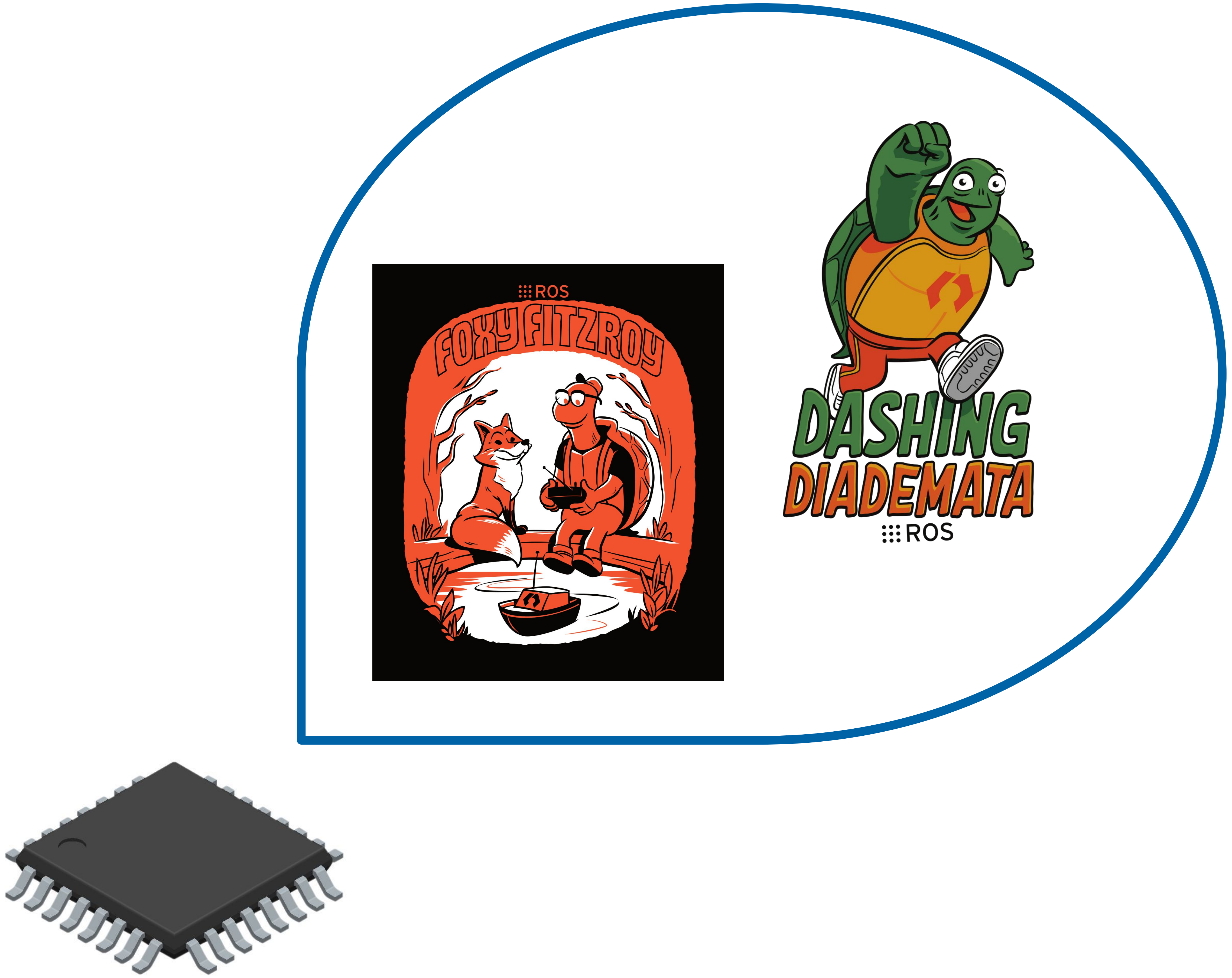
A solution for creating ROS 2 nodes into embedded devices



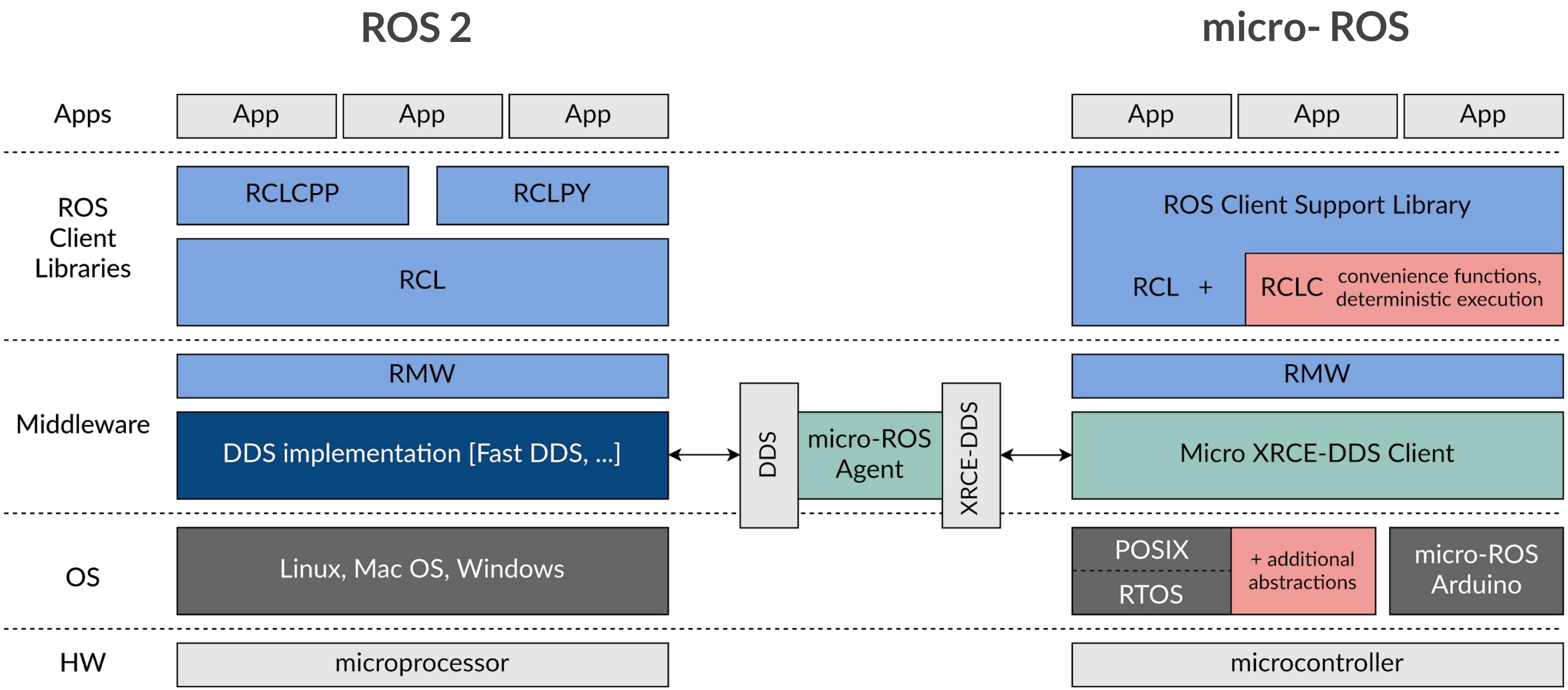
Why micro-ROS?

Highlights

- Layer-compatible with ROS 2
- Integrated into ROS 2 ecosystem
- Allows to create a ROS 2 node with ~ all functionalities
- *Client/server* logics
- Middleware transports fully customizable
- Runs on different RTOSes and MCUs
- Platform-versatile cross-compilation tools
- Benefits of full QoS support
- Now supporting ***Foxy***
- A growing community!



micro-ROS architecture



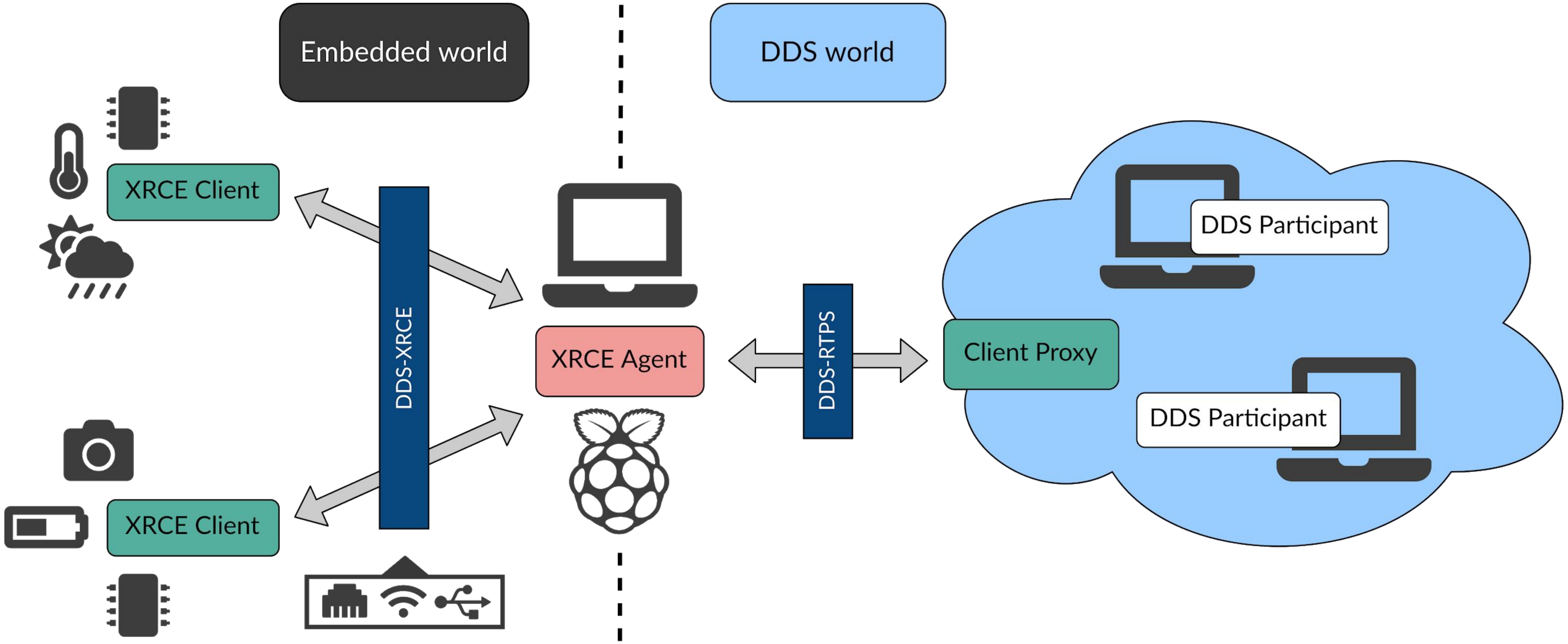
Middleware architecture

Micro XRCE-DDS: DDS for e**X**tremely **R**esource-**C**onstrained **E**nvironments.

Clients - XRCE entities on low-resource consumption devices.
Agent - XRCE entity connected with DDS global data space. Acts on behalf of Clients in the DDS world.

Main features:

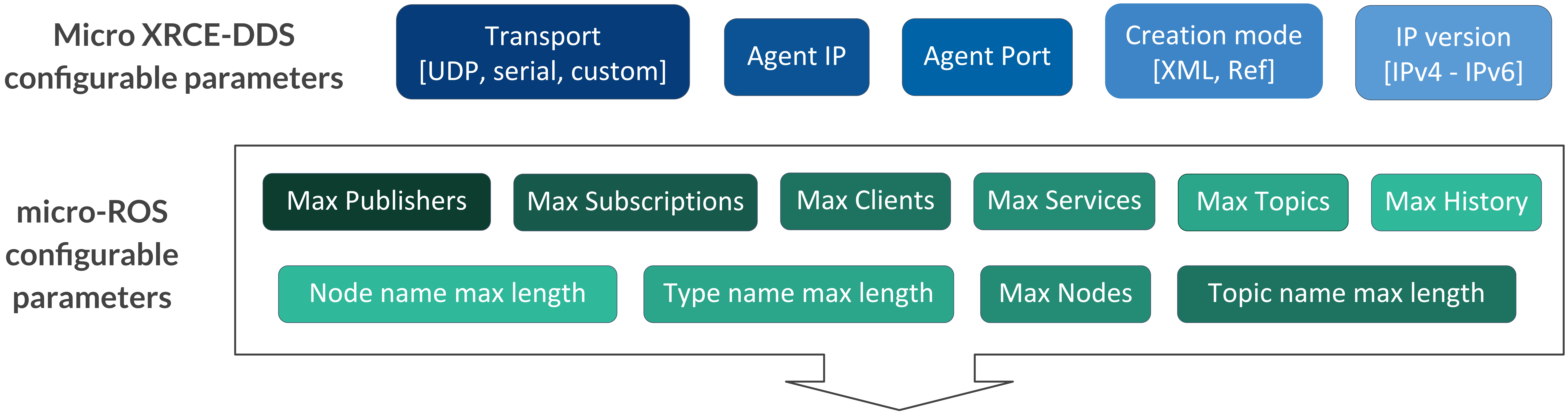
- *Client-server architecture*
- *Request-response pattern*
- *Connection oriented*



eProsima
The Middleware Experts

RMW

- Implemented using Micro XRCE-DDS middleware in lower layers
  - Allows static configuration of memory resources



Micro XRCE-DDS configurable parameters

Transport [UDP, serial, custom]

Agent IP

Agent Port

Creation mode [XML, Ref]

IP version [IPv4 - IPv6]

micro-ROS configurable parameters

Max Publishers

Max Subscriptions

Max Clients

Max Services

Max Topics

Max History

Node name max length

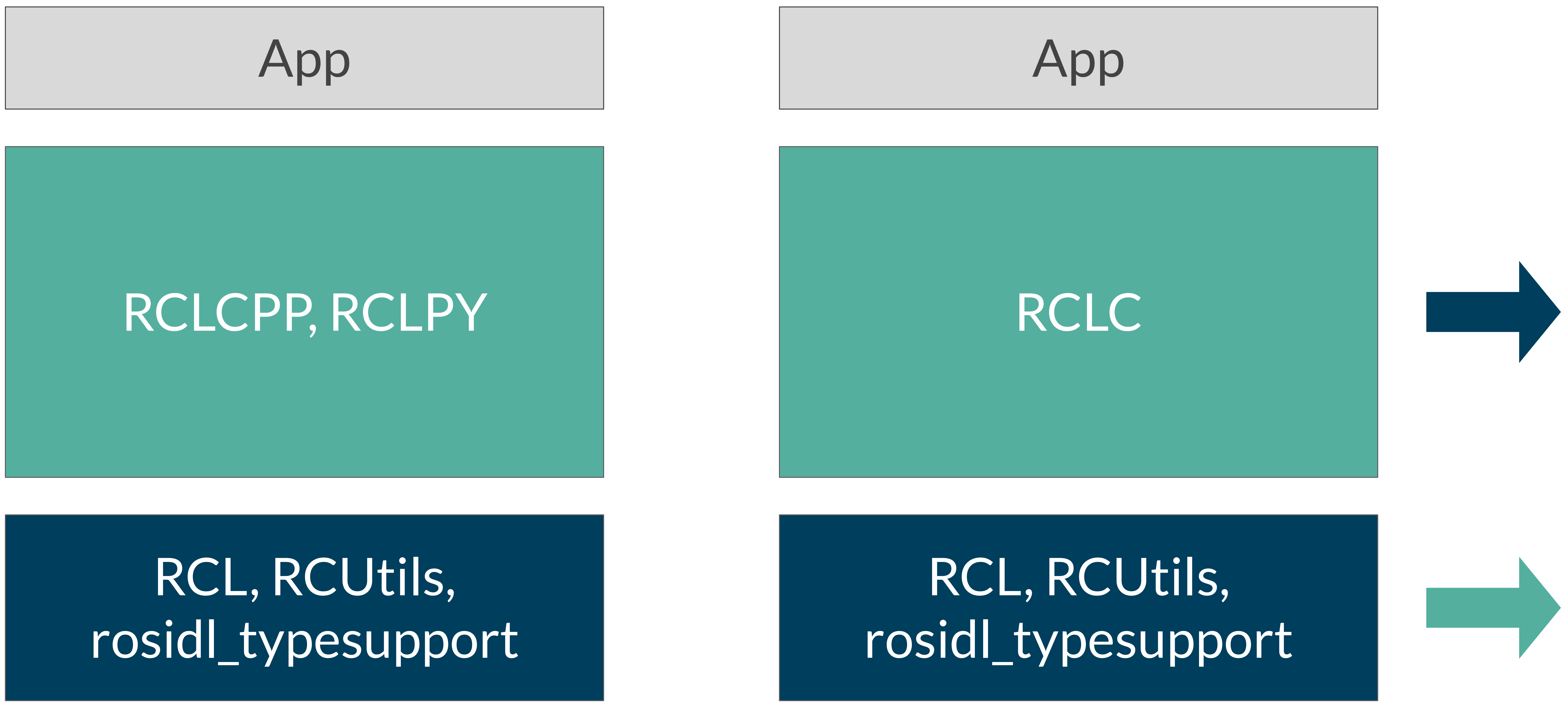
Type name max length

Max Nodes

Topic name max length

Configurability of these parameters allows preconfiguring the size of the library and tuning the size of the buffer to the memory needed

ROS Client Support Libraries



App

RCLCPP, RCLPY

RCL, RCUtils, rosidl_typesupport

App

RCLC

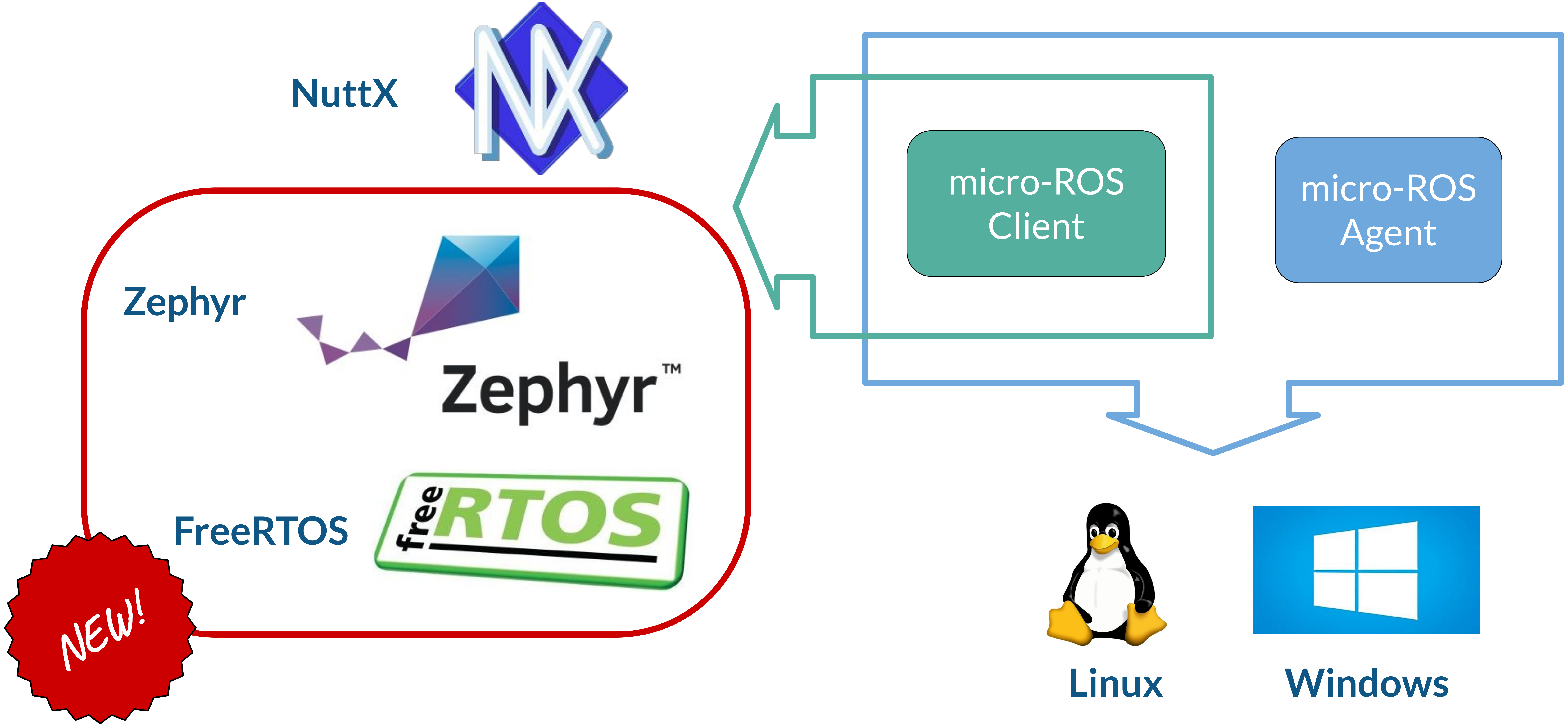
RCL, RCUtils, rosidl_typesupport

C99 library:
provides utility functions for creating ***nodes***, ***publishers***, ***subscribers*** & ***redesigned executor*** [deterministic and LET semantics, dynamic memory allocation only at startup, domain-specific scheduling]

Same as in ROS 2
(many functionalities not used)

Supported platforms

Supported RTOSes



NuttX

Zephyr

FreeRTOS

NEW!

micro-ROS Client

micro-ROS Agent

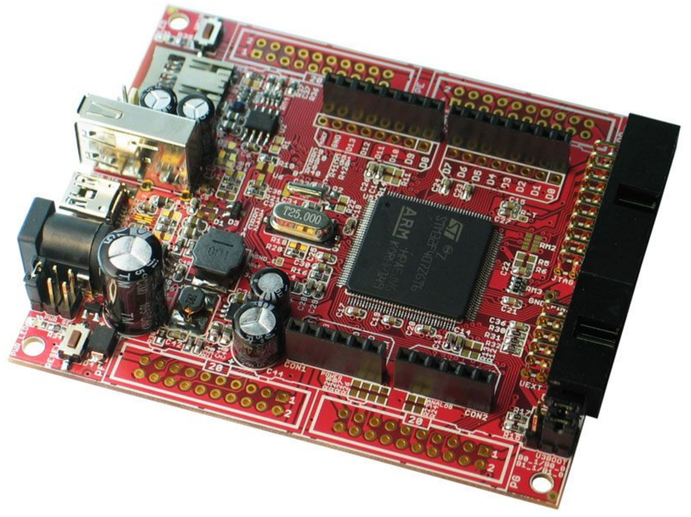
Linux

Windows

Supported HW

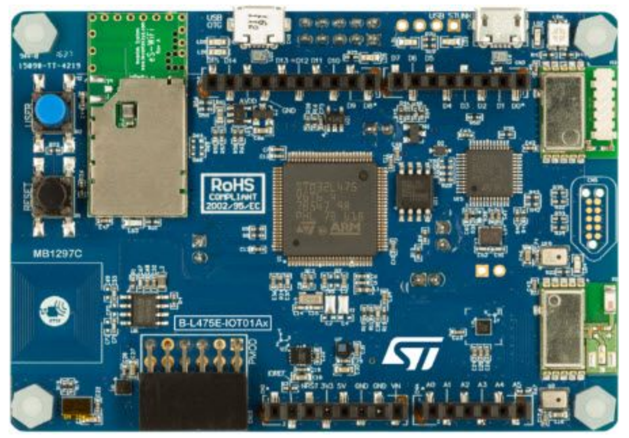
Officially supported HW...

Olimex LTD
STM32-E407



Crazyflie 2.1 drone

STM32L4
Discovery kit IoT



ESP32-DevKitC-32E

Target: mid-range microcontrollers.

Currently supported:

- ARM-M4/M7 MCUs (STM32, i.MX RT ...)
- Xtensa MCUs (ESP32)

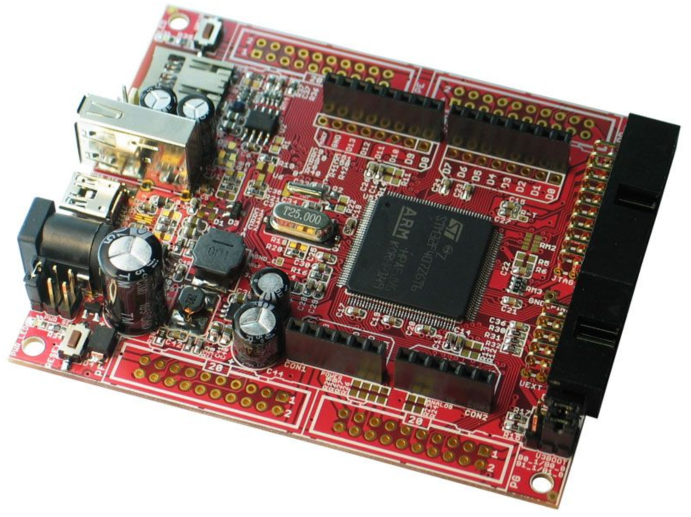
Typical features:

- ~ 1MB of flash memory
- ~ 200 KB of RAM memory
- < 500 mA consumption
- General purpose input/output pins (GPIO)
- Communication peripherals: USB, Ethernet, SPI, UART, I2C, CAN, etc

Supported HW

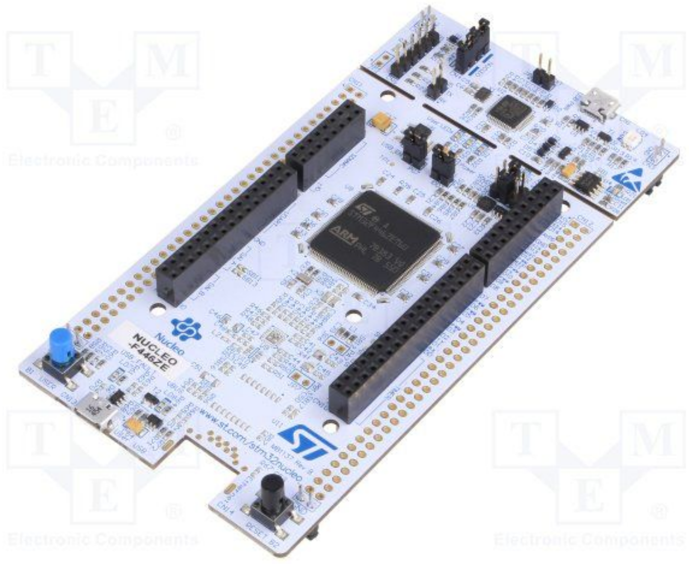
… + community-supported HW!

Olimex LTD
STM32-E407

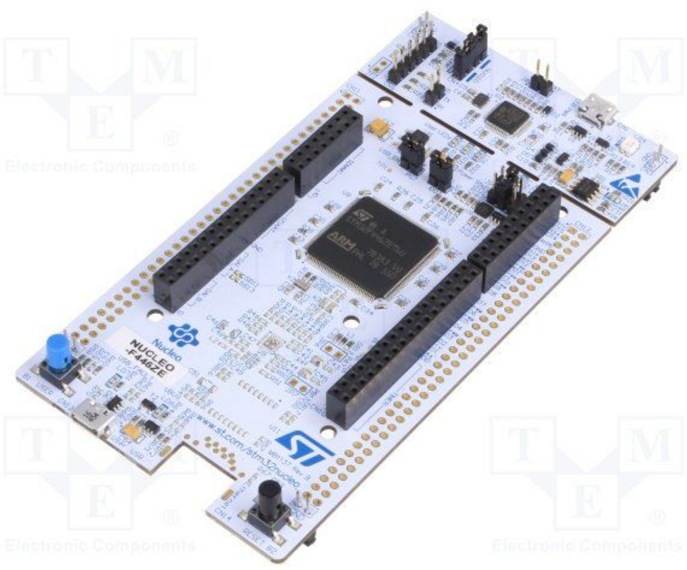


Crazyflie 2.1 drone

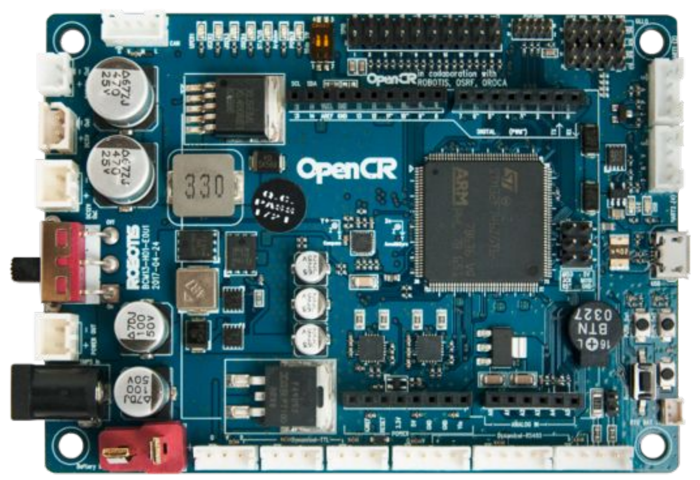
ST Nucleo F446ZE



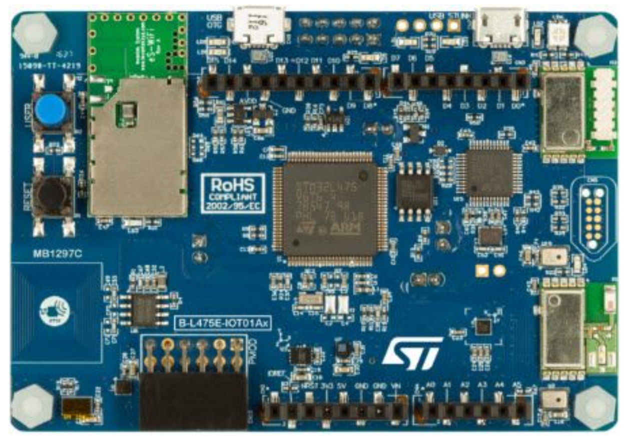
ST Nucleo H743ZI



OpenCR 1.0

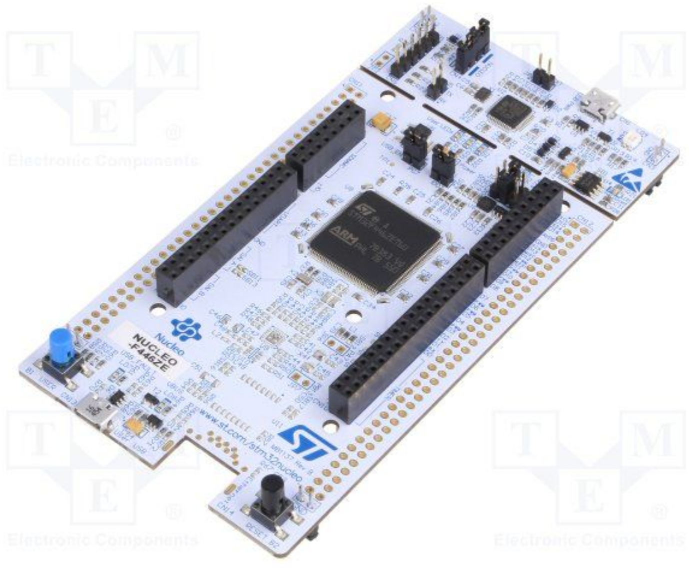


STM32L4
Discovery kit IoT



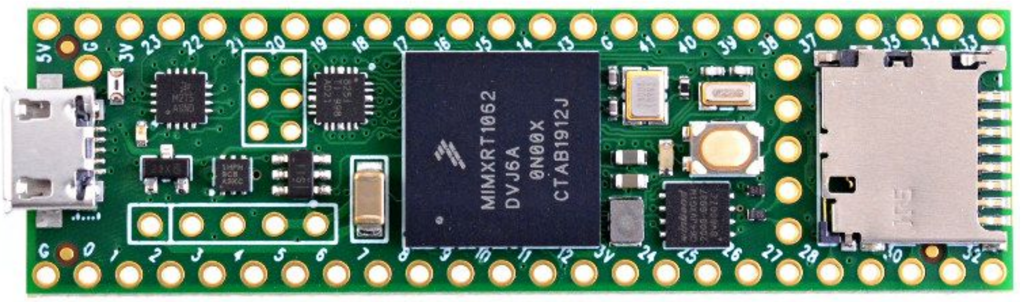
ESP32-DevKitC-32E

ST Nucleo F746ZG



Teensy 3.2

Teensy 4.1

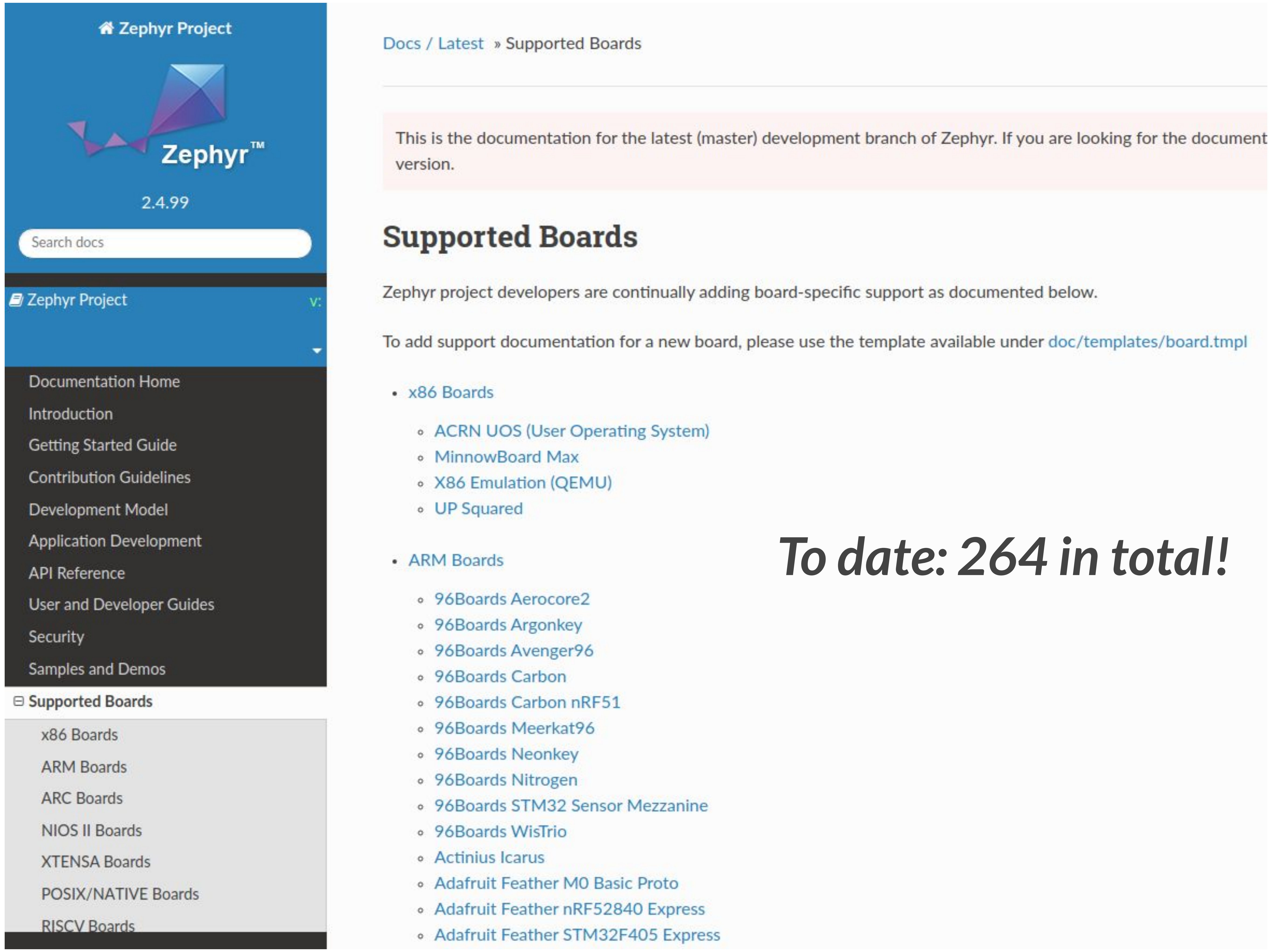


Ease of porting new platforms

Porting new boards with Zephyr RTOS is super-easy thanks to the huge amount of boards already supported by The Zephyr Project!

Compatibilities to be aware of:

- *Memory resources*
- *Transports*



Docs / Latest » Supported Boards

This is the documentation for the latest (master) development branch of Zephyr. If you are looking for the document version.

Supported Boards

Zephyr project developers are continually adding board-specific support as documented below.

To add support documentation for a new board, please use the template available under doc/templates/board.tmpl

- x86 Boards
  - ACRN UOS (User Operating System)
  - MinnowBoard Max
  - X86 Emulation (QEMU)
  - UP Squared
- ARM Boards
  - 96Boards Aerocore2
  - 96Boards Argonkey
  - 96Boards Avenger96
  - 96Boards Carbon
  - 96Boards Carbon nRF51
  - 96Boards Meerkat96
  - 96Boards Neonkey
  - 96Boards Nitrogen
  - 96Boards STM32 Sensor Mezzanine
  - 96Boards WisTrio
  - Actinius Icarus
  - Adafruit Feather M0 Basic Proto
  - Adafruit Feather nRF52840 Express
  - Adafruit Feather STM32F405 Express

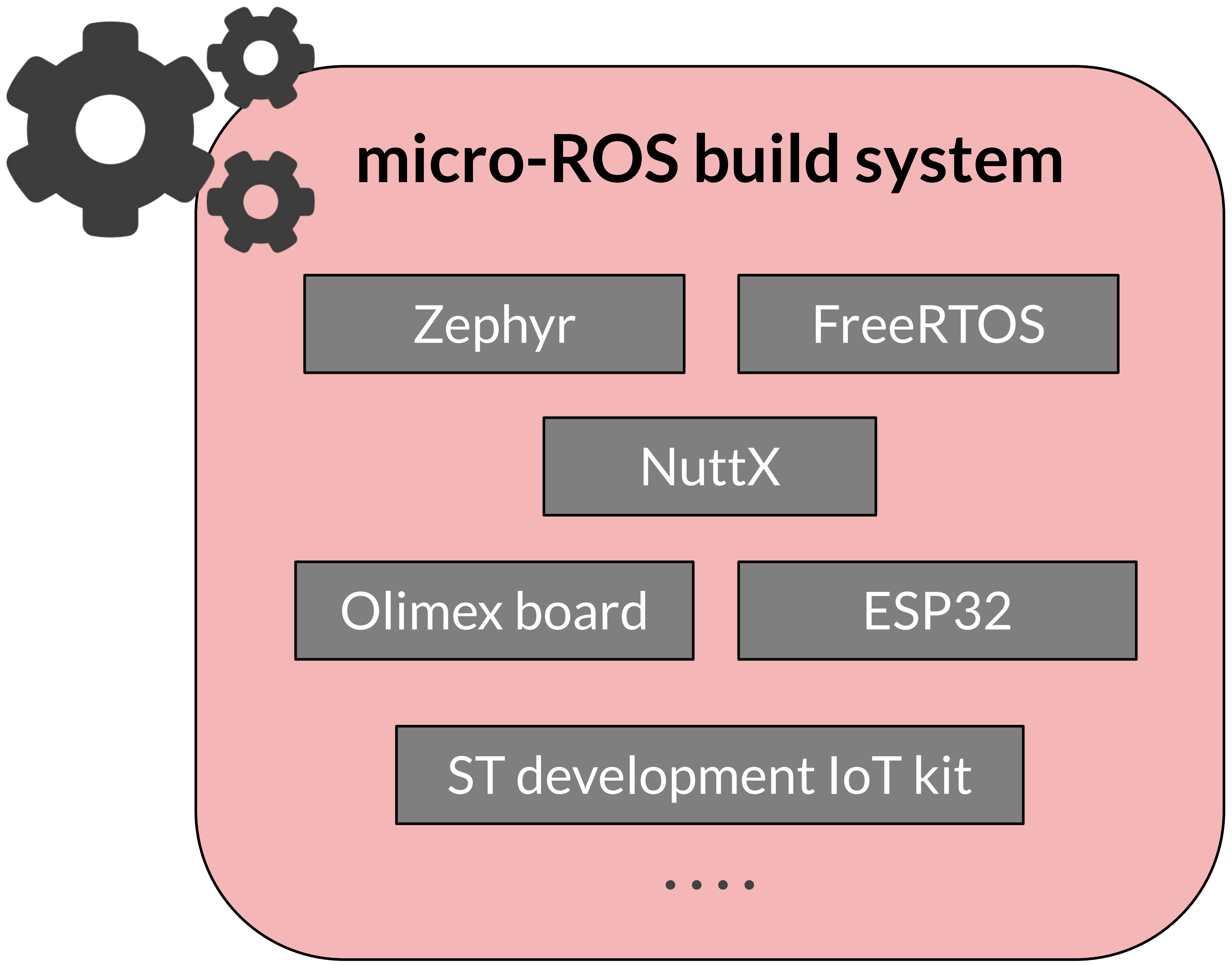
To date: 264 in total!

Recent integrations, developments and WIPs

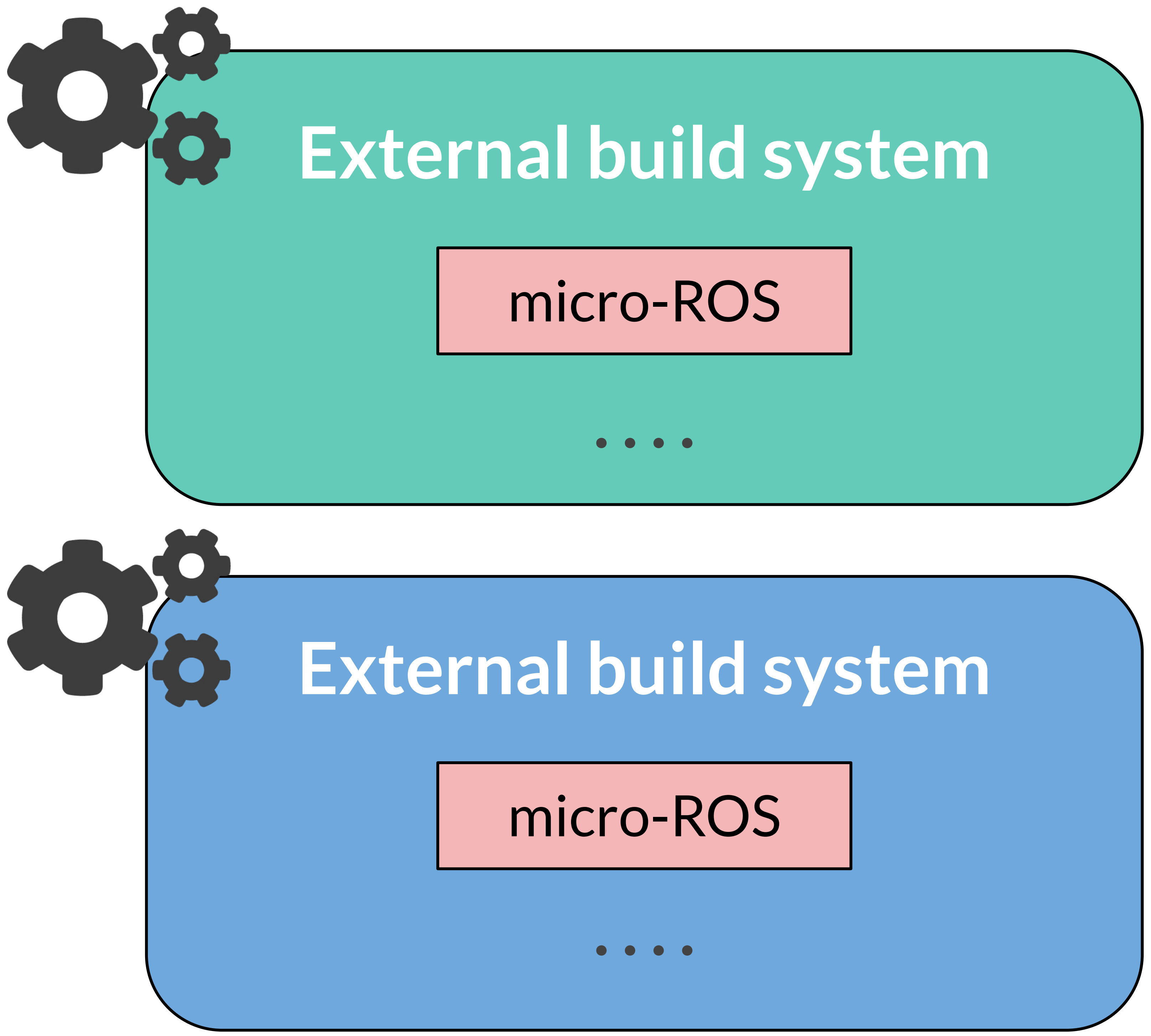
Giving the build system a twist

The micro-ROS build system: now a two-tales story

Classic approach



New complementary approach



Achieved by generating standalone micro-ROS library & headers

Giving the build system a twist

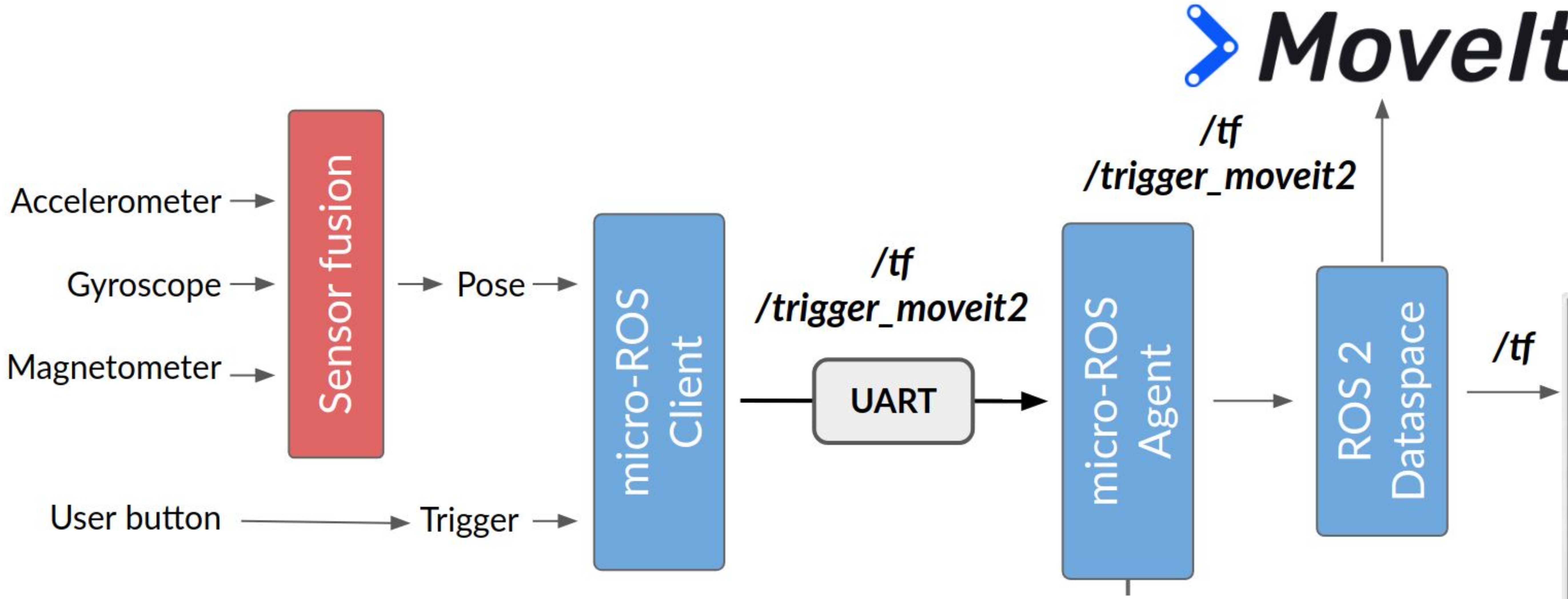
micro-ROS as an
ESP-IDF component

micro-ROS as a
Zephyr module

micro-ROS into
Arduino IDE

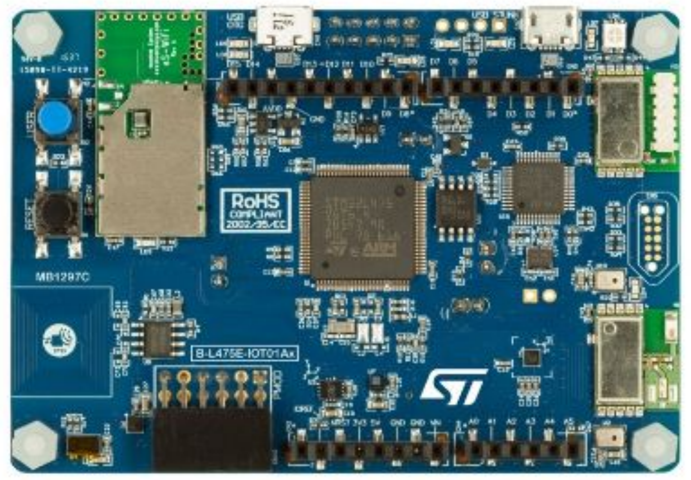
... and more are to come!

micro-ROS meets MoveIt 2!



Plan and move arm to a pose on trigger

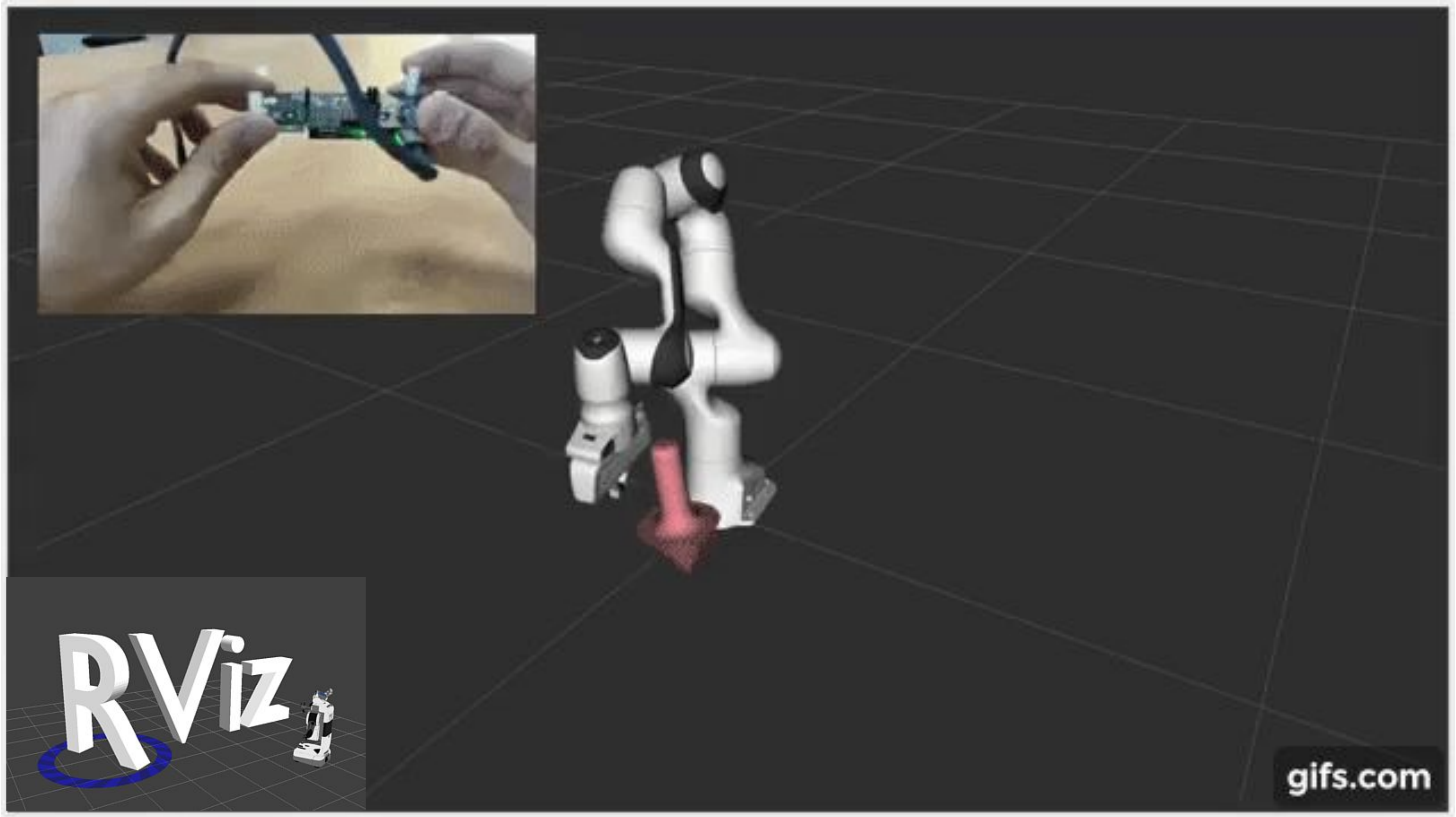
STM32L4 Discovery kit IoT



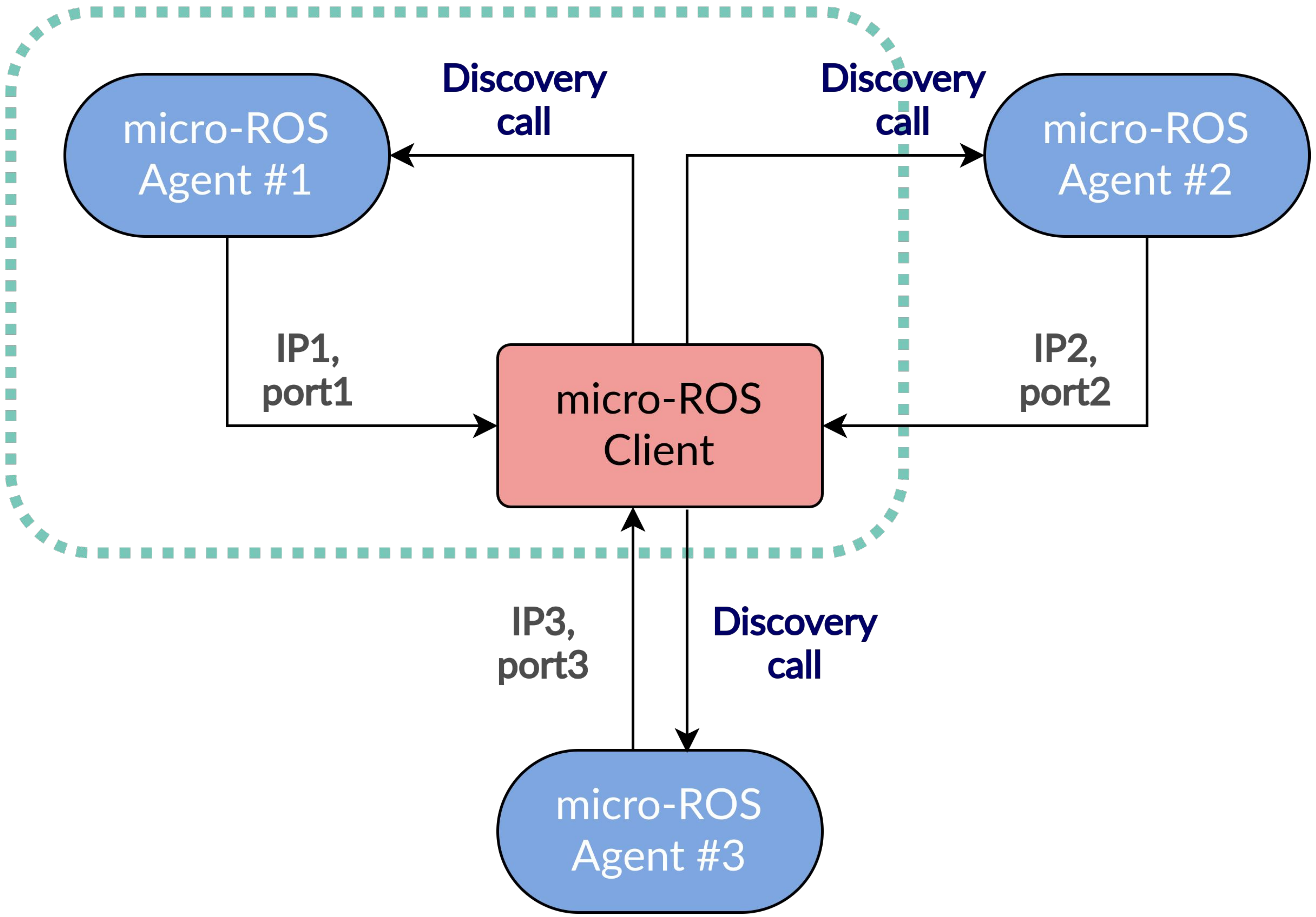
Zephyr™ Project

ROS

2



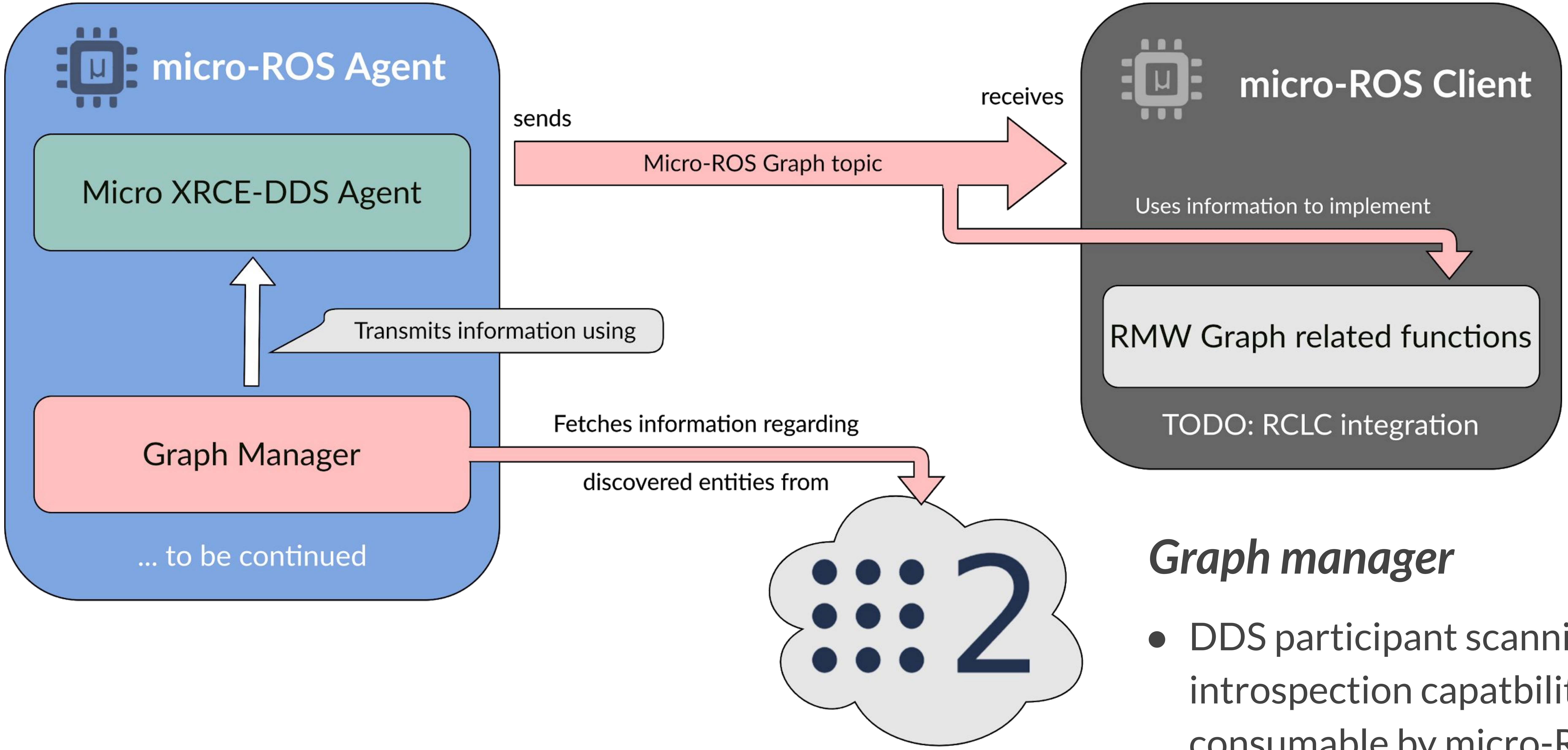
Agent discovery



Discovery mechanism flow

- Clients discovery call by multicast on UDP
- Reachable Agents respond providing IP & port
- Clients match with first available Agent

Graph support



Graph manager

- DDS participant scanning the network: provides introspection capatbilities to user. ROS 2 topology consumable by micro-ROS
- micro-ROS topology info available to ROS 2

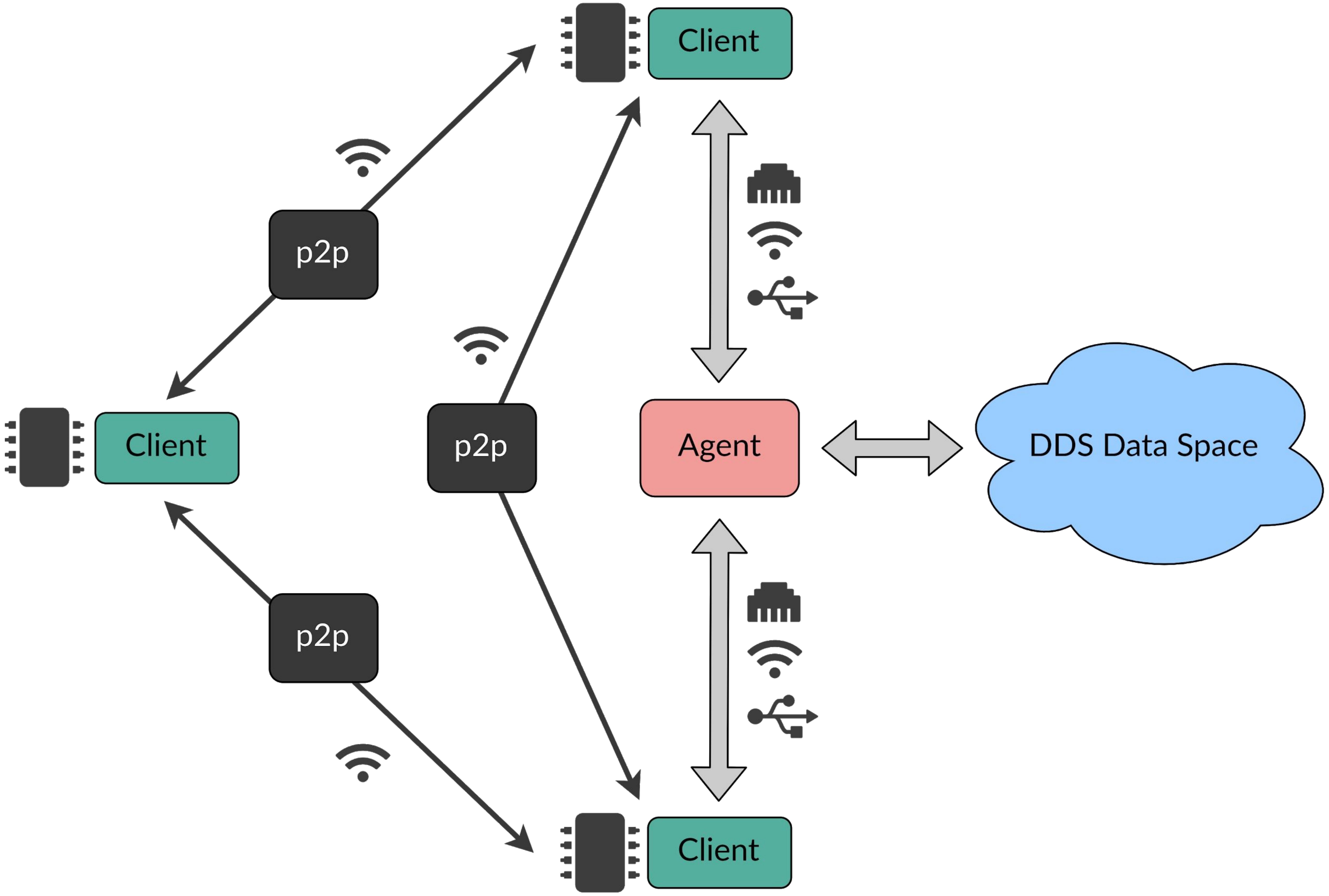
WIP: P2P functionality

P2P prototype

- Clients send info about themselves on broadcast
- Clients can choose whether to connect via the Agent or by P2P [WIP]
- At present, P2P offers limited set of functionalities
- Tried on:

ESP32

freeRTOS

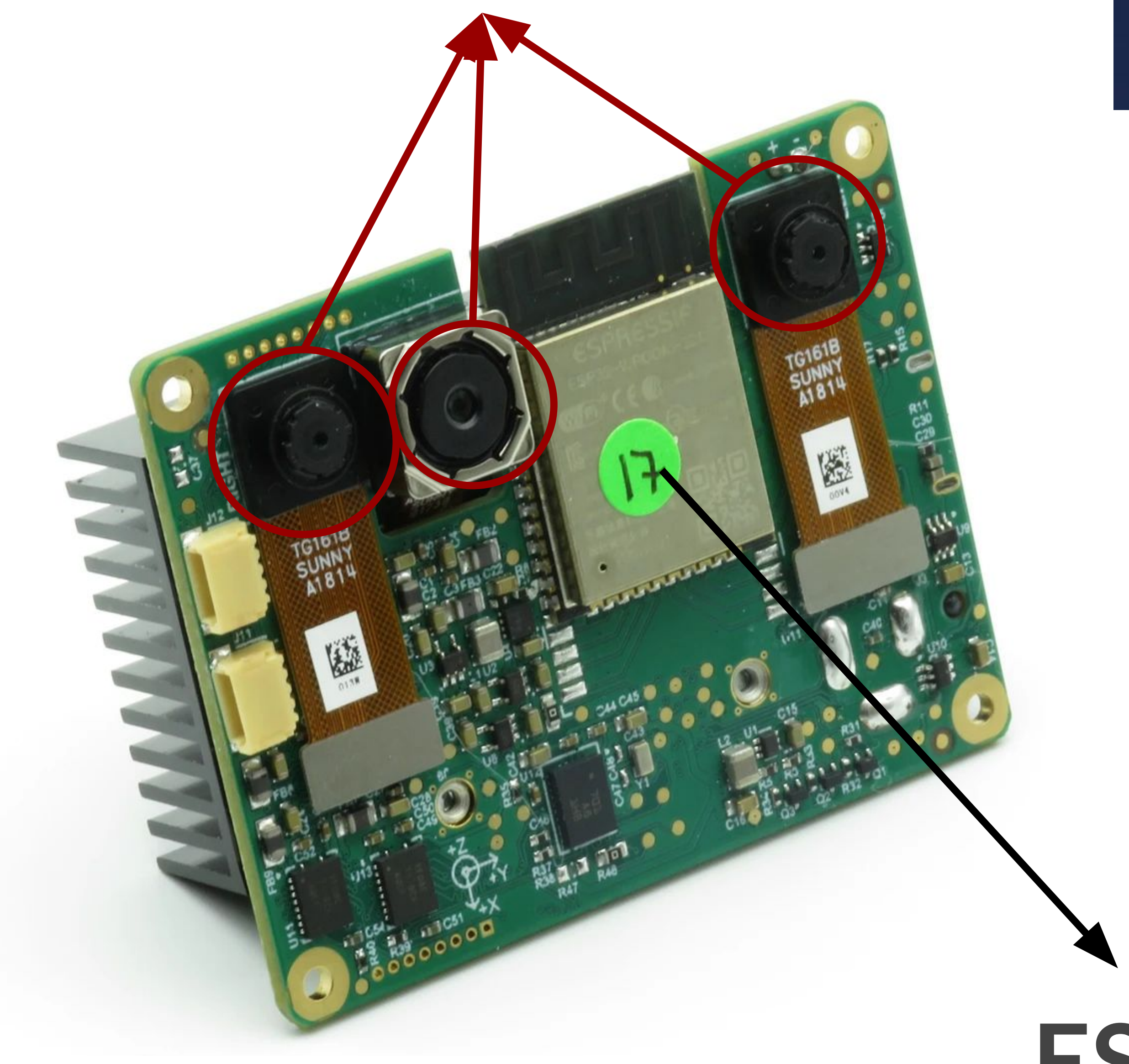


WIP: micro-ROS goes AIoT!

DEPTH ai

Open-source platform - custom hardware, firmware, software & AI training - that combines ***neural inference***, ***depth vision***, and ***feature tracking***.

Enables for ***embedded artificial intelligence*** and ***spatial AI/CV***.



freeRTOS

LUXonis

+

µROS

WIP

micro-ROS on DepthAI via ESP32 support: combining embedded artificial intelligence with ROS 2 ecosystem!

Thanks for your attention!

Q&A time

QoS

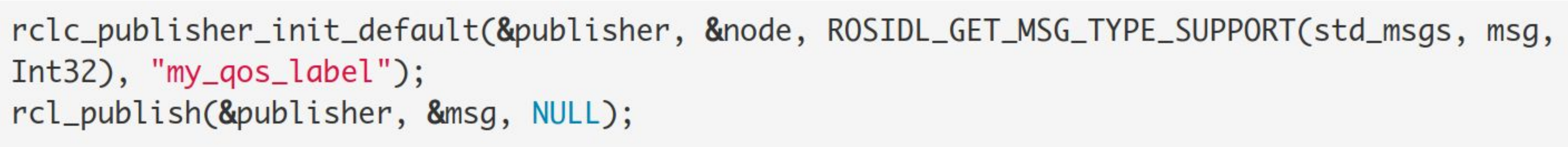
Two possibilities for entities creation:

- ***By XML (on Client) - default***
- ***By reference (on Agent) - allows full use of QoS***

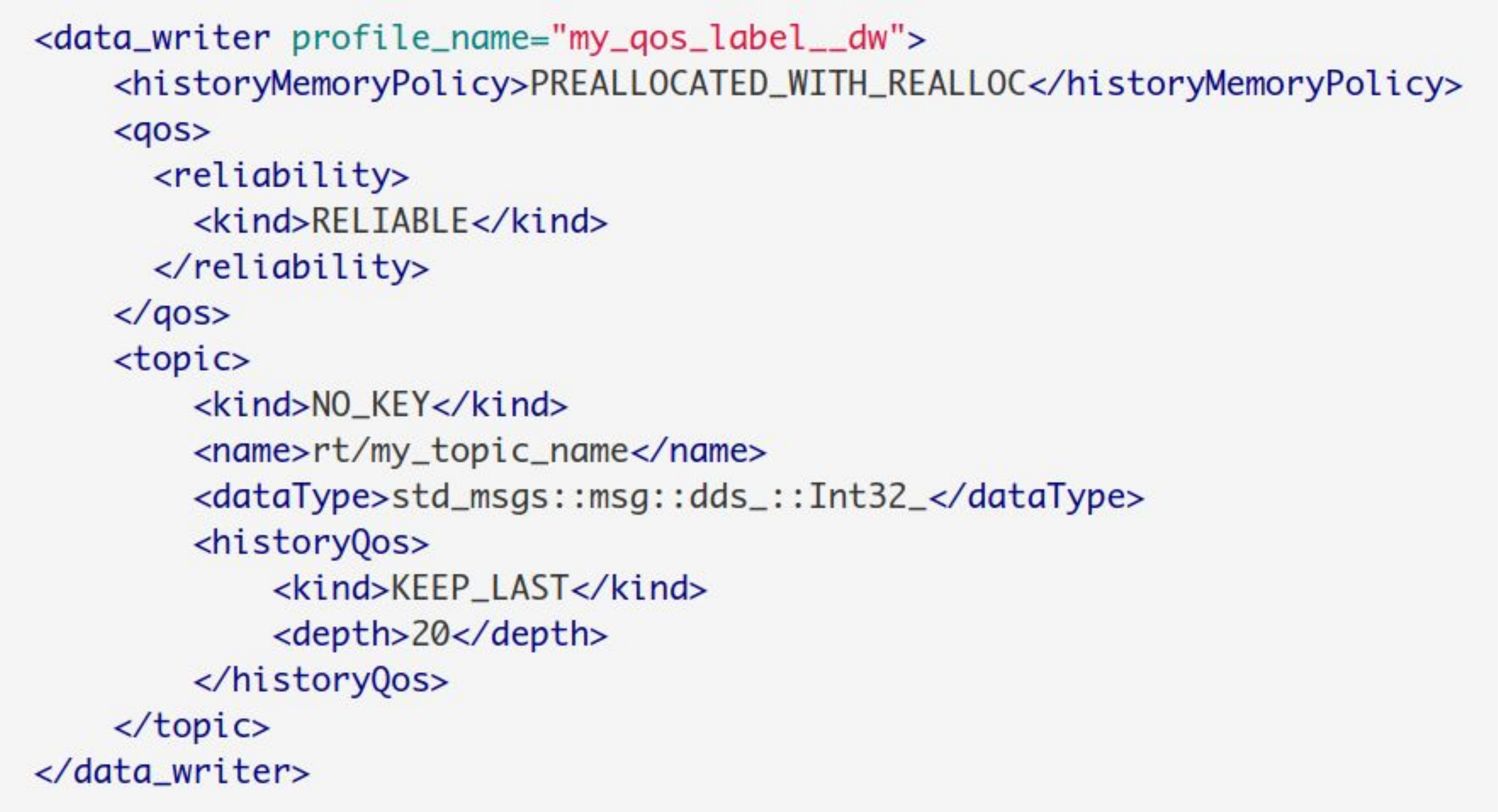
Users can write custom QoS on the Agent's side. Each entity has its own label and the Client creates the entities using this reference label.

Advantages of using creation by reference:

- Reduces memory consumption of micro-ROS Client inside the MCU.
- Full set of DDS QoS available



```
rclc_publisher_init_default(&publisher, &node, ROSIDL_GET_MSG_TYPE_SUPPORT(std_msgs, msg,
Int32), "my_qos_label");
rcl_publish(&publisher, &msg, NULL);
```



```
<data_writer profile_name="my_qos_label__dw">
    <historyMemoryPolicy>PREALLOCATED_WITH_REALLOC</historyMemoryPolicy>
    <qos>
      <reliability>
        <kind>RELIABLE</kind>
      </reliability>
    </qos>
    <topic>
        <kind>NO_KEY</kind>
        <name>rt/my_topic_name</name>
        <dataType>std_msgs::msg::dds_::Int32_</dataType>
        <historyQos>
            <kind>KEEP_LAST</kind>
            <depth>20</depth>
        </historyQos>
    </topic>
</data_writer>
```

Memory profiling

Transport: UDP

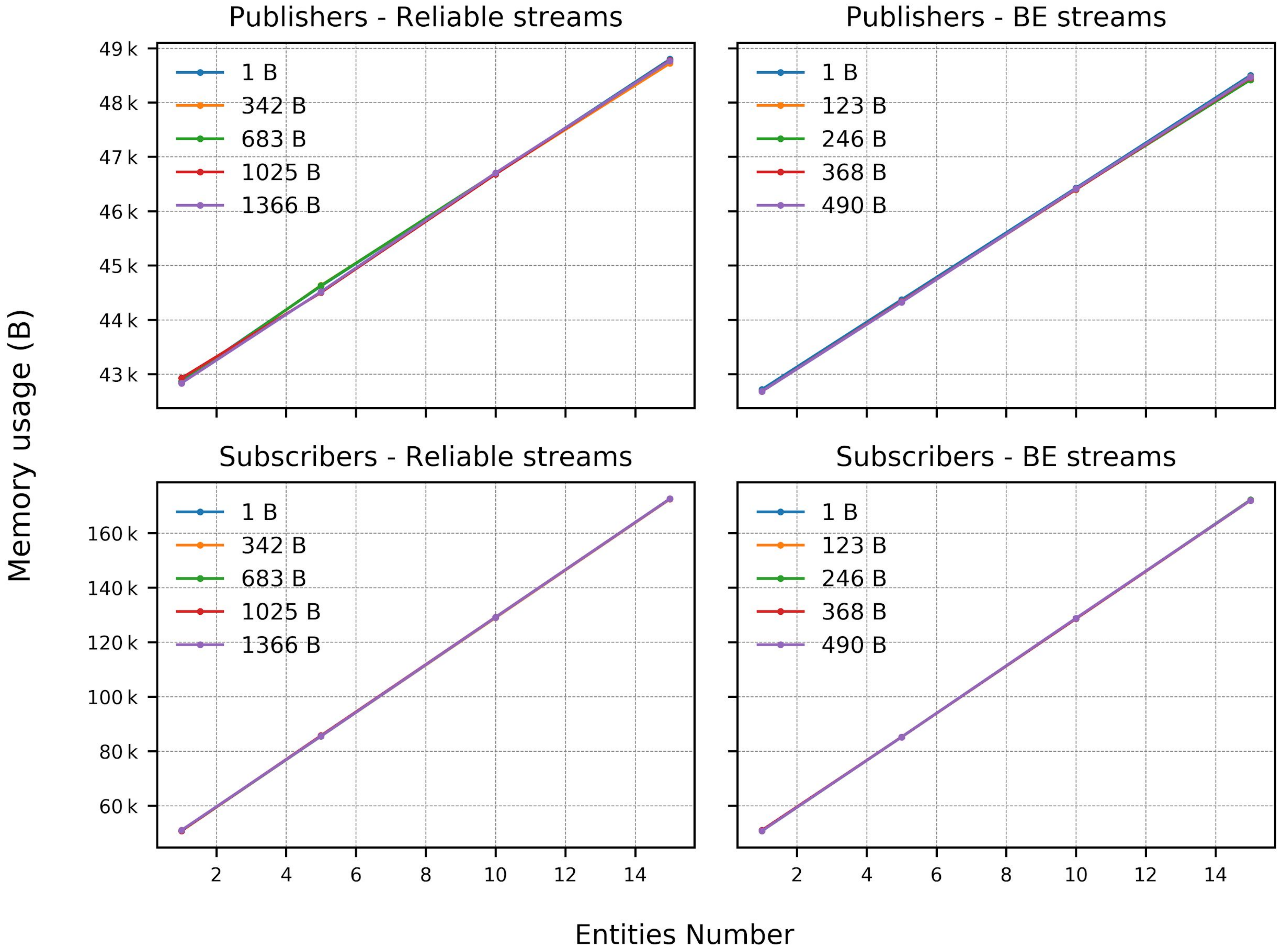
Creation: by XML

RMW history = 4

MTU = 512 B

XRCE history = 4

- *Total memory consumed by 1 pub ~ 400 B*
- *Total memory consumed by 1 sub ~ 8700 B*



Memory profiling

Transport: UDP

Creation: by XML

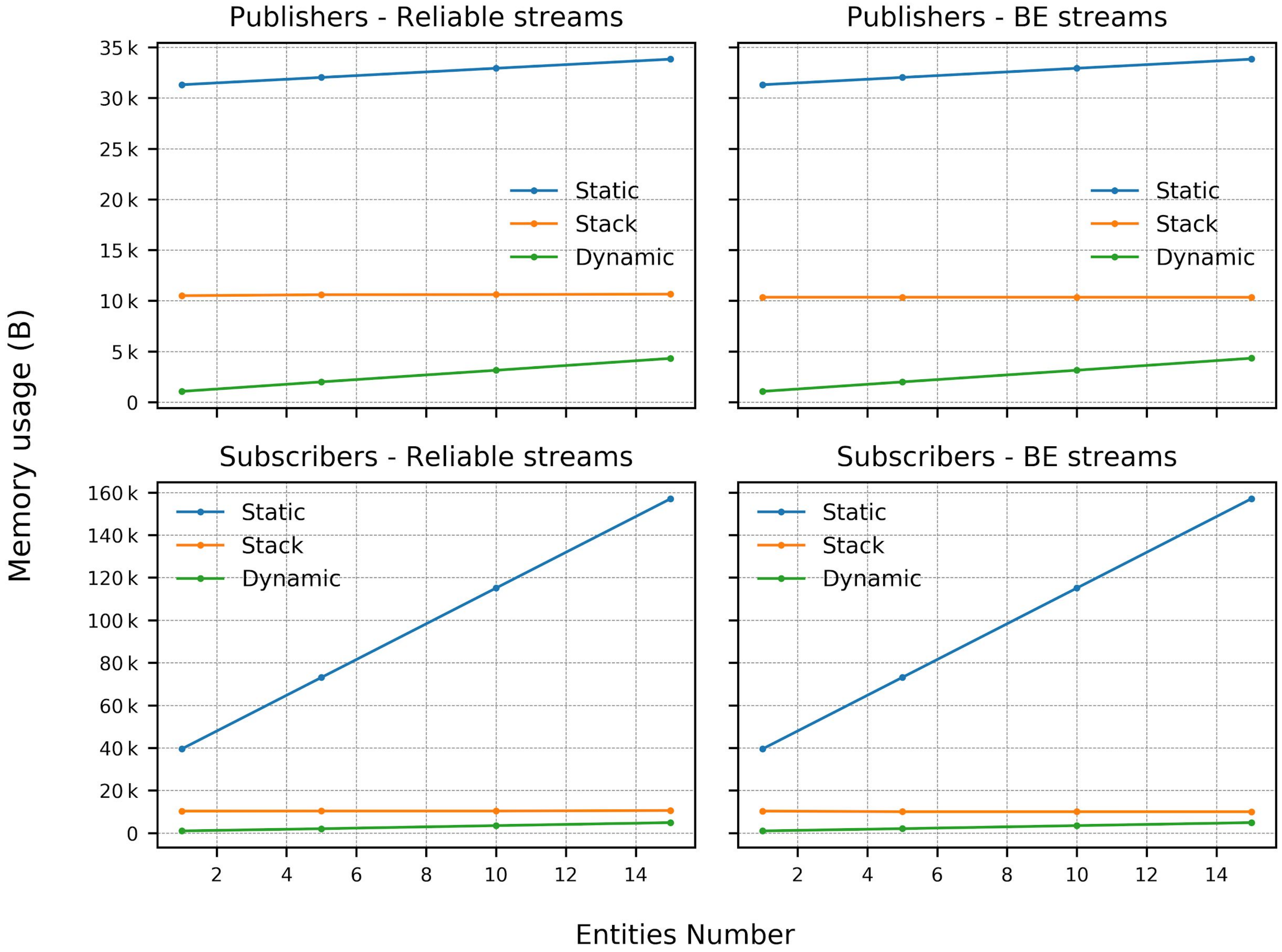
RMW history = 4

MTU = 512 B

XRCE history = 4

Overall memory:

- *Static*
- *Stack*
- *Dynamic*



Memory profiling

Transport: UDP

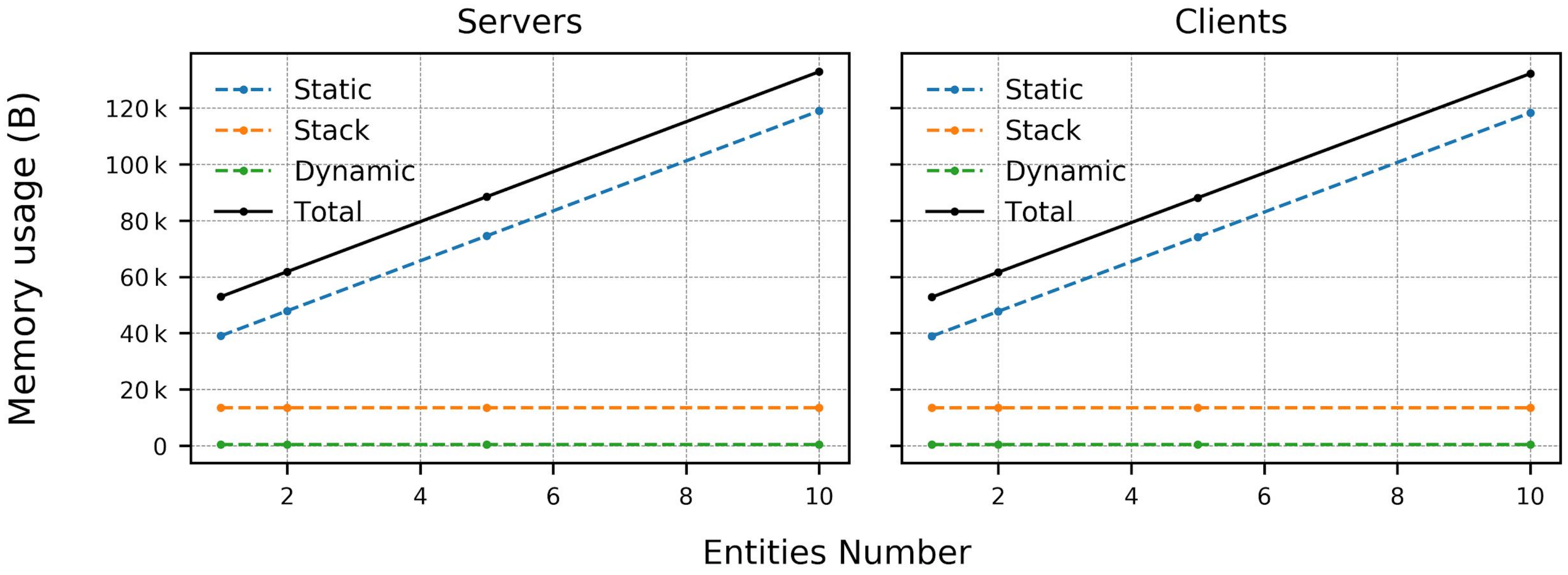
Creation: by XML

Comm stream: Reliable

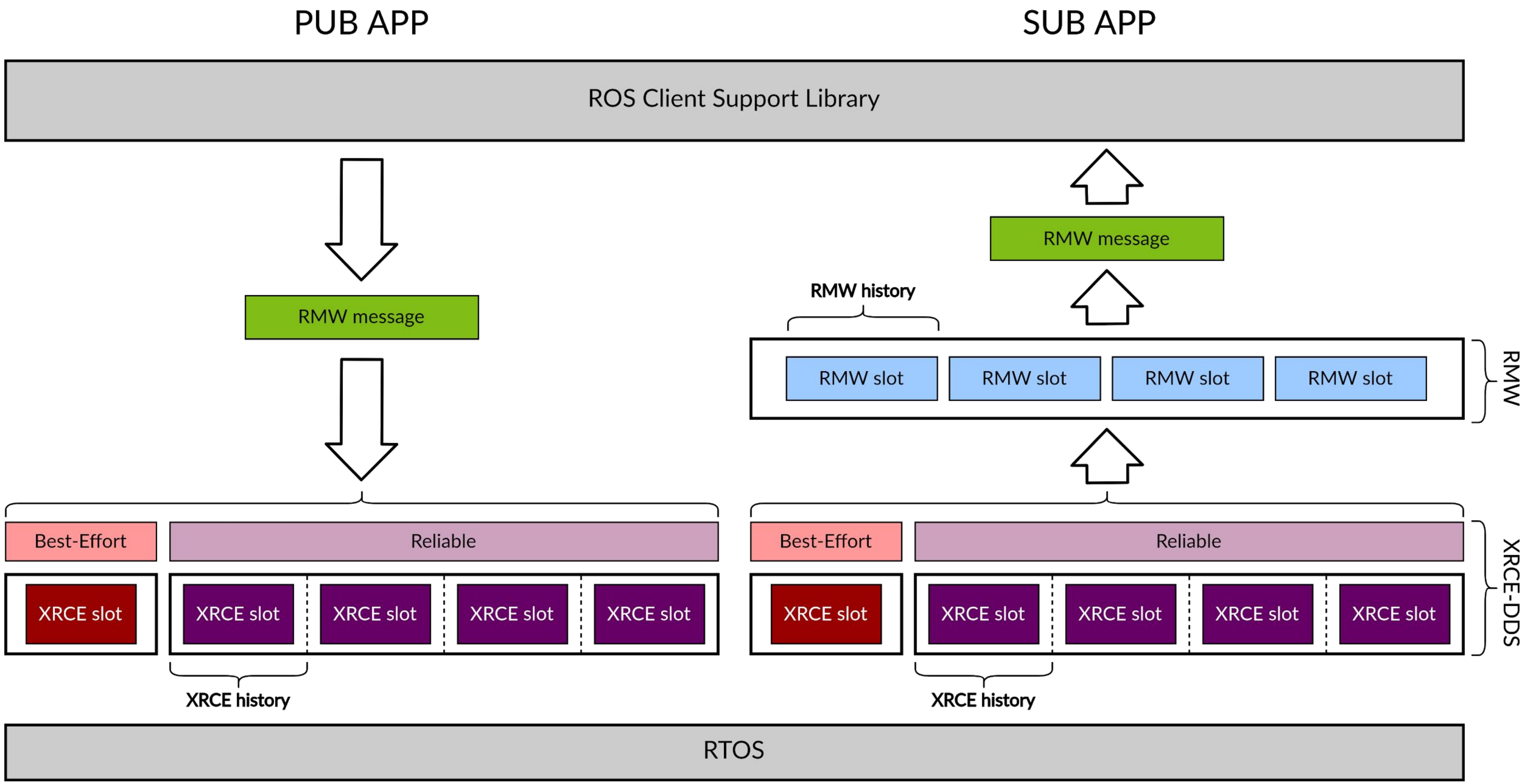
RMW history = 4

MTU = 512 B

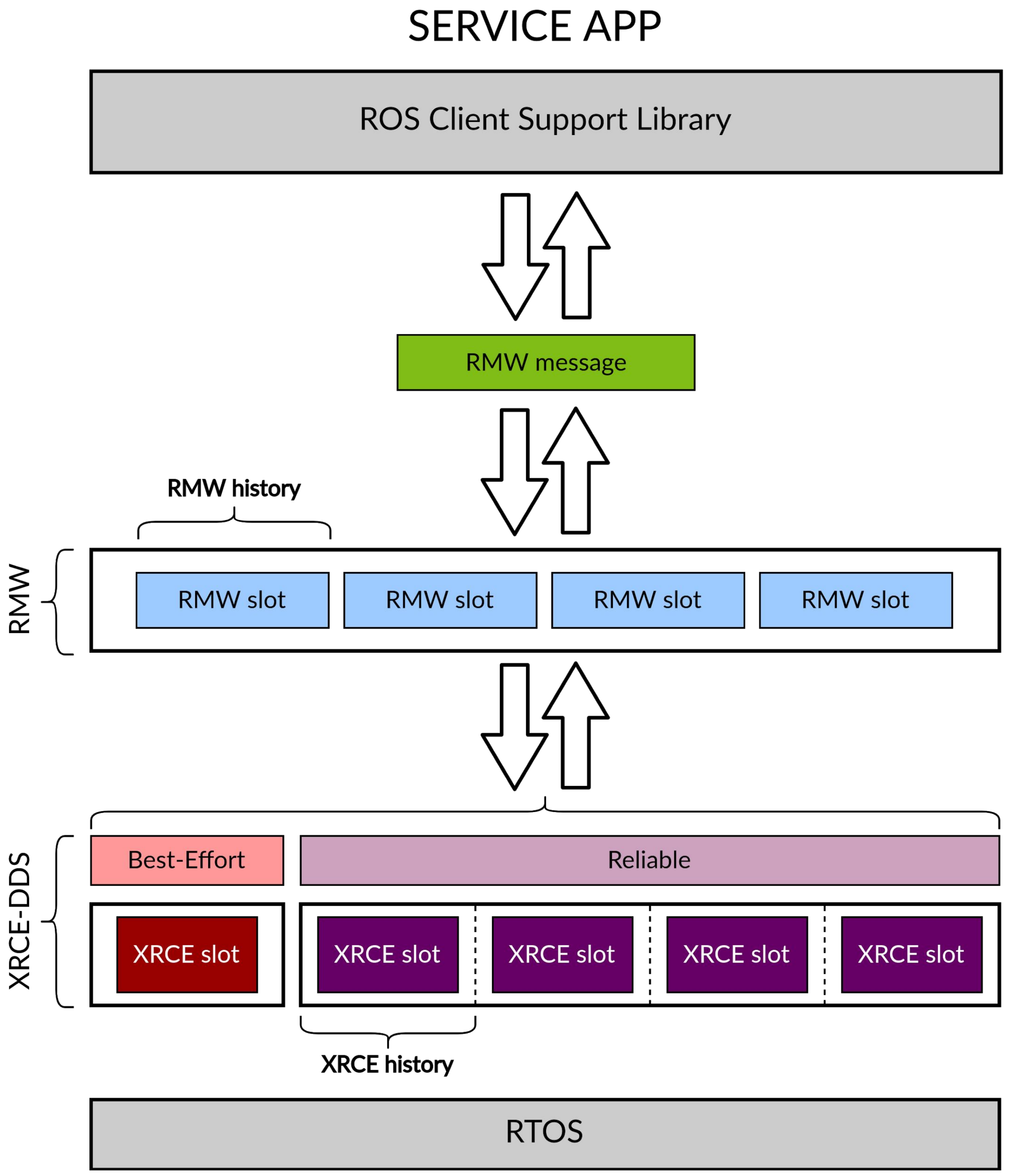
XRCE history = 4



Memory management: pub/sub



Memory management: services


SERVICE APP
ROS Client Support Library
RMW message
RMW history
RMW
RMW slot
RMW slot
RMW slot
RMW slot
XRCE-DDS
Best-Effort
Reliable
XRCE slot
XRCE slot
XRCE slot
XRCE slot
XRCE slot
XRCE history
RTOS