ROS 2 in PX4:
Technical details of a seamless transition to XRCE-DDS and micro-ROS

PX4 Developer Summit 2021

15 September 2021

Pablo Garrido (Lead Embedded SW Engineer @ eProsima)
Nuno Marques (Senior Robotics SW Engineer @ Auterion)

AGENDA

micro-ROS Update 01
Pablo from eProsima

Micro XRCE-DDS and micro-ROS in PX4 02
Nuno from Auterion

Questions and Answers 03

micro-ROS Update

- What is micro-ROS?
- Layered Architecture
- RMW Architecture
- Portability capabilities
- NuttX 10.0 10.1 example

Why micro-ROS?

XRCE (µC) Embedded world

Robotics trend evolves towards interconnected systems of CPUs and **multisensor-actuator (that run on low resource boards µC)**

New inherent challenges

Memory limitations, real-time systems, energy consumption, wide range of vendors.
Lack of common standard development framework

micro-ROS mission

Common framework ROS 2 based which Mission is to bring ROS 2 nodes into the embedded world (µC)

Why micro-ROS?

A solution for creating ROS 2 nodes into embedded devices

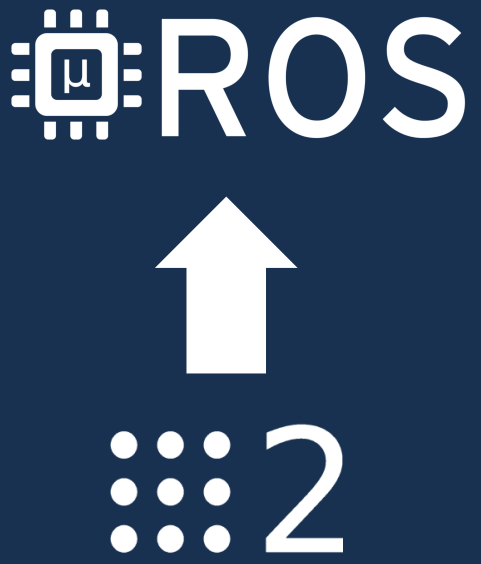
- **Accelerator** of application development via allowing the combination of CPUs and µC within any robotic system
- **Enabler** of affordable deployments (IoT, robotics, autonomous driving ,...)

µROS

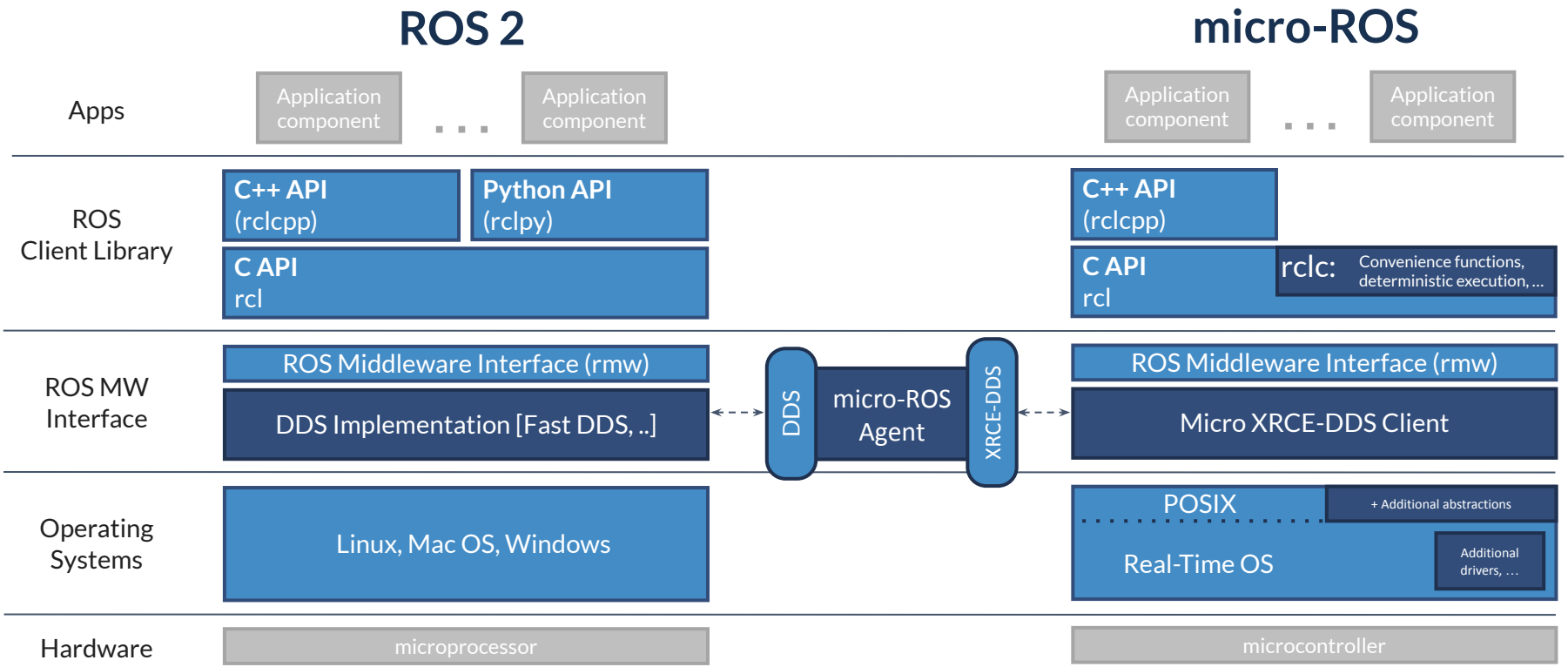
2

Highlights

- **Mirroring ROS2 for Embedded world**
  - Layer-compatible with ROS 2
  - Integrated into ROS 2 ecosystem
  - Allows to create a ROS 2 node with ~ all functionalities
  - Client-server logics (client fully dynamic memory free)
- **Widest range of use cases**
  - Middleware transports fully customizable
  - Runs on bare-metal, all RTOSs and all MCUs
  - Platform-versatile cross-compilation tools
- **Mature technology**
  - Benefits of full QoS support ROS 2
  - Now supporting **Foxy and Galactic and Rolling**
  - A growing community



micro-ROS layered architecture



Middleware architecture

Micro XRCE-DDS

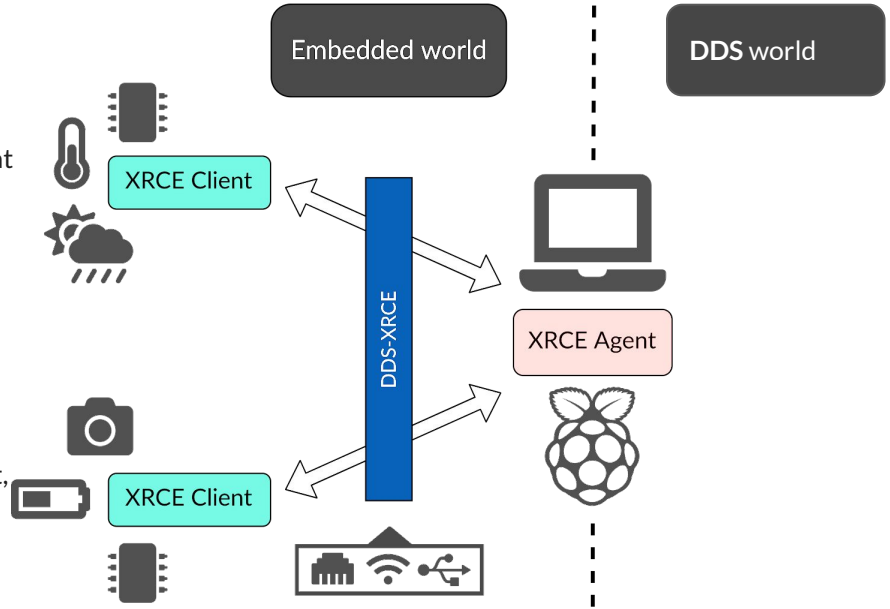
- **Wire-protocol over Client-Server architecture**

  XRCE Client on low-resource consumption devices

  XRCE Agent entity connected with DDS global data space that acts on behalf of Clients
- **Client fully static and dynamic memory free**

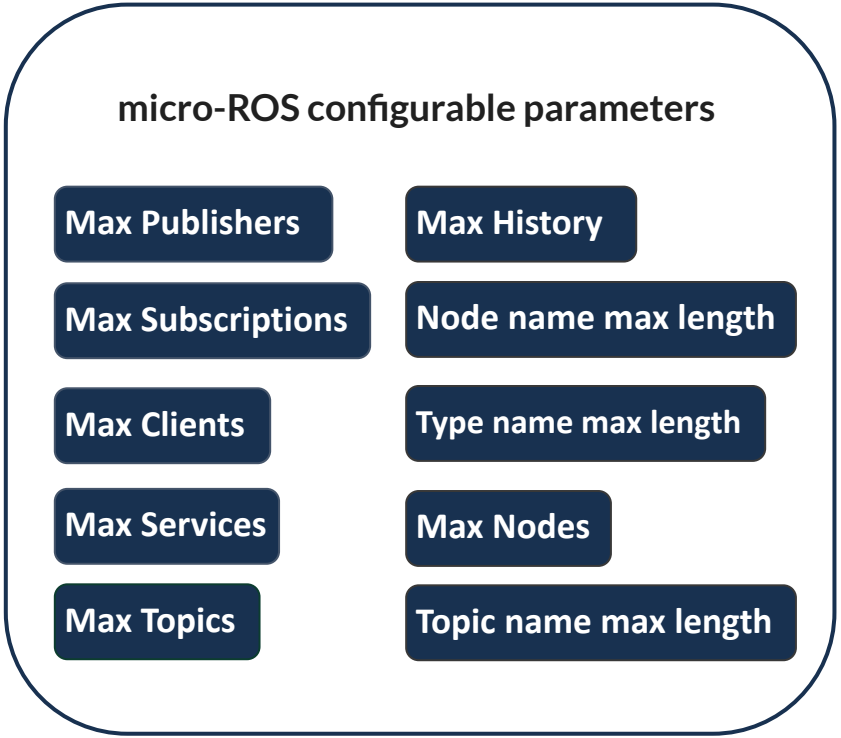
  75 KB of Flash memory and 3 KB of RAM
- **Real-Time and Deterministic - critical applications**
- **Transport-agnostic, customized by the user**

  Built-in support for serial transports, TCP, UDP over Ethernet, Wi-Fi, and 6LoWPAN, and Bluetooth



Memory optimization

- Implemented using Micro XRCE-DDS middleware in lower layers
- Allows smart configuration of memory resources (micro-ROS)
  - Static configuration
  - Parameter level



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

- **Full support for STM32F7/H7/F4/F3**
- **General support for ARM Cortex M7/4/3**
- Teensy 3.2 / 3.5 / 4.1 / 4.2
- ESP-IDF v4.3 & ESP32-S2/C3
- Raspberry RP2040
- OpenCR support, STM32CubeMX/IDE
- Olimex LTD STM32-E407,Crazyflie 2.1 drone
- **EK RA6M5 + e2studio**

REFERENCE RTOS

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

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

PX4 autopilot µROS

AGENDA

micro-ROS Update 01
Pablo from eProsima

Micro XRCE-DDS and micro-ROS in PX4 02
Nuno from Auterion

Questions and Answers 03

Micro XRCE-DDS and micro-ROS in PX4

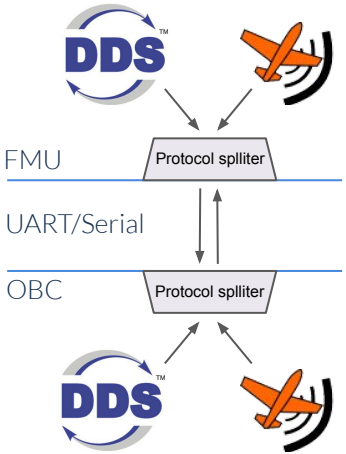
- micro-RTPS update
- micro-RTPS vs Micro XRCE-DDS
  - Why does it matter to PX4?
- Migration: what has to happen?
  - Roadmap

micro-RTPS update

- Latest features and fixes
  - Support for ROS 2 Galactic
  - Support for default ROS 2 DDS participants configurations
  - Support for ***`ROS_LOCALHOST_ONLY`*** and Shared Memory transport
  - In SITL, the micro-RTPS client now starts automatically
  - Timesync with ROS-defined time (including simulation time)
  - Topic naming following ROS topic pattern conventions
  - Support data polling on the client to data stream priority

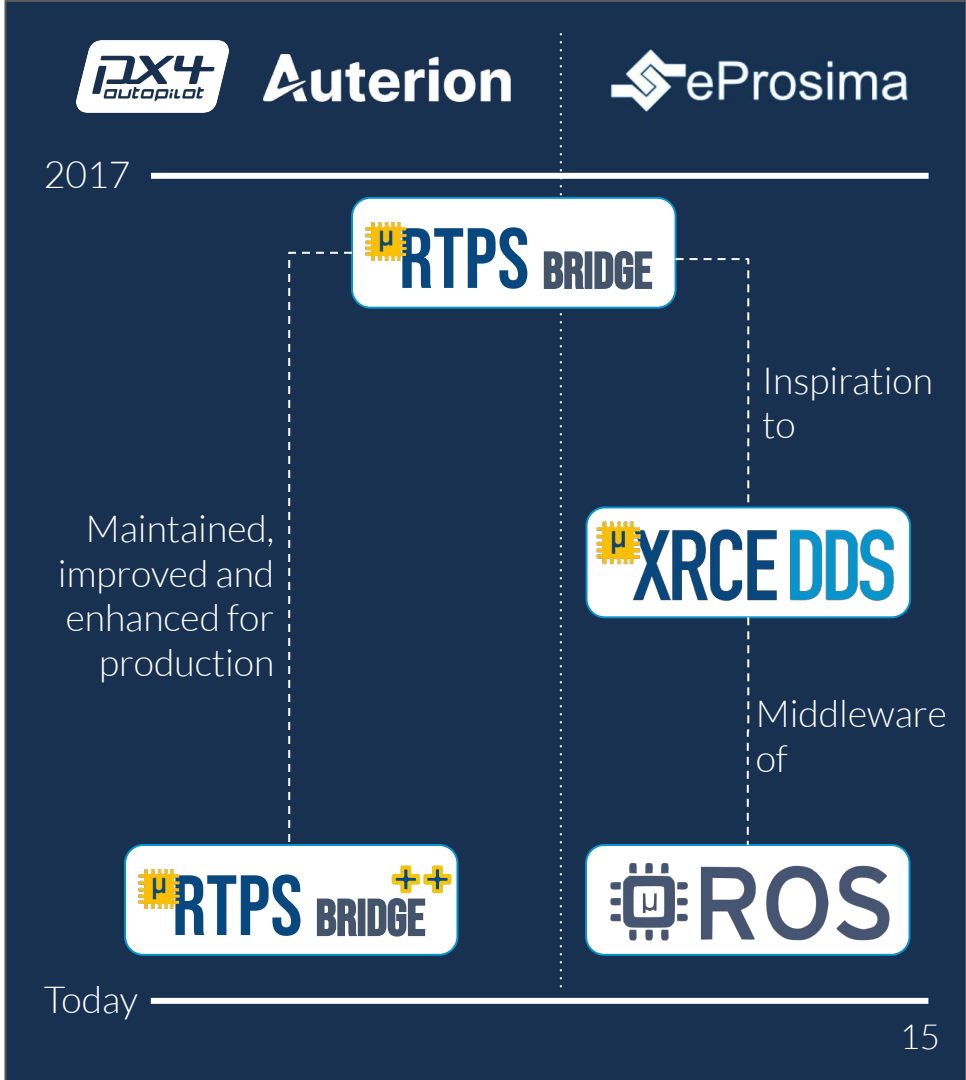
micro-RTPS update

- Latest features and fixes
  - Fully functional RTPS/MAVLink protocol splitter (UART)
  - Bridge RTPS IDs set automatically
  - Improved UX and workflow for updating bridge messages
  - Faster agent code generation
  - UART setup flags optimization
  - Client bandwidth usage overall limiter
  - Improved and extended documentation



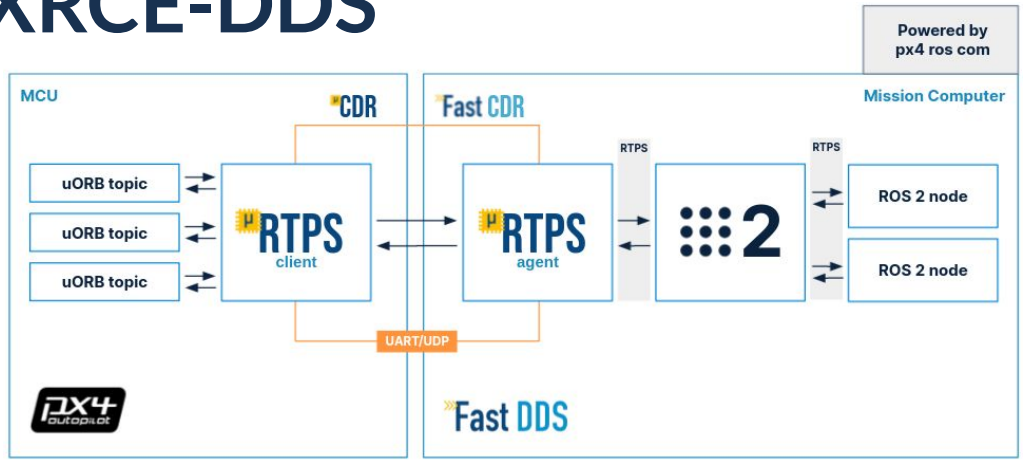
micro-RTPS vs micro XRCE-DDS

- micro-RTPS was added in 2017 as a proof-of-concept to bridge PX4 with DDS
- The bridge was maintained and improved by the PX4 community, with the development efforts driven by Auterion
- eProsima progressed to support OMG DDS-XRCE, creating micro XRCE-DDS, which serves as the backbone to micro-ROS

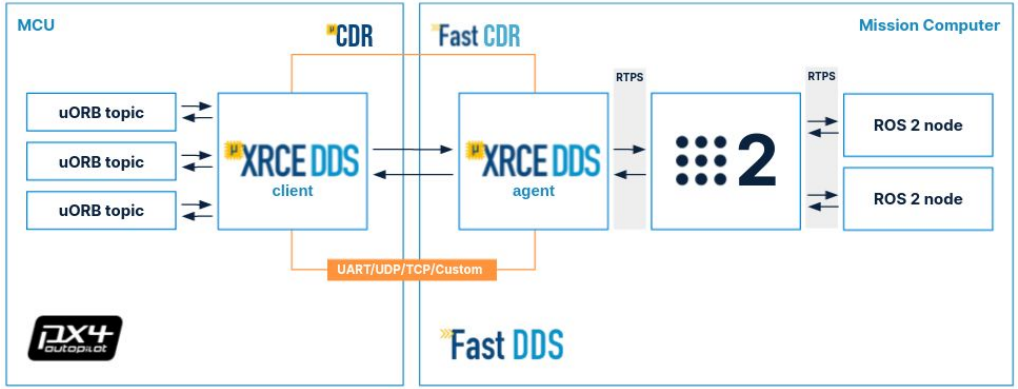


micro-RTPS vs micro XRCE-DDS

µRTPS BRIDGE



µXRCE DDS



micro-RTPS vs micro XRCE-DDS

	µRTPS BRIDGE	µXRCE DDS
Serialization	micro-CDR + Fast CDR	micro-CDR + Fast CDR
Transport	UDP / Serial	UDP / TCP / Serial / Custom
Code Generation	*FastDDSGen* to generate the agent using the FastDDS API	*microXRCEDDSGen* to generate the client using the XRCE-DDS API
Timesync	Through uORB message exchange, using ROS or monotonic time	Native, using NTP, but allows custom callbacks

micro-RTPS vs micro XRCE-DDS

	µRTPS BRIDGE	µXRCE DDS
Client	• There is no concept of publishing or subscribing to the DDS Global Data Space - only write or read data from the underlying link wire transport protocol • PX4 module that serves as a parser between the uORB data and the microRTPS wire protocol	• Can either publish and subscribe directly to data Topics in the DDS Global Data Space • Allow the creation of Entities on the Agent which can act on behalf of the Clients in the DDS world - *Proxy Client* • Implemented as a library
Agent	• Data parser that serializes and deserializes data to / from the link protocol and publishes / subscribes to the DDS Data Space • Statically defined according to the agent Publisher / Subscriber templates - no dynamic entity creation	• Server that acts as an intermediary between the client and the DDS Global Data Space • Receives messages containing operations from the Clients, and keeps track of the Clients and of the entities they create

PX4 autopilot ROS

micro-RTPS vs micro XRCE-DDS

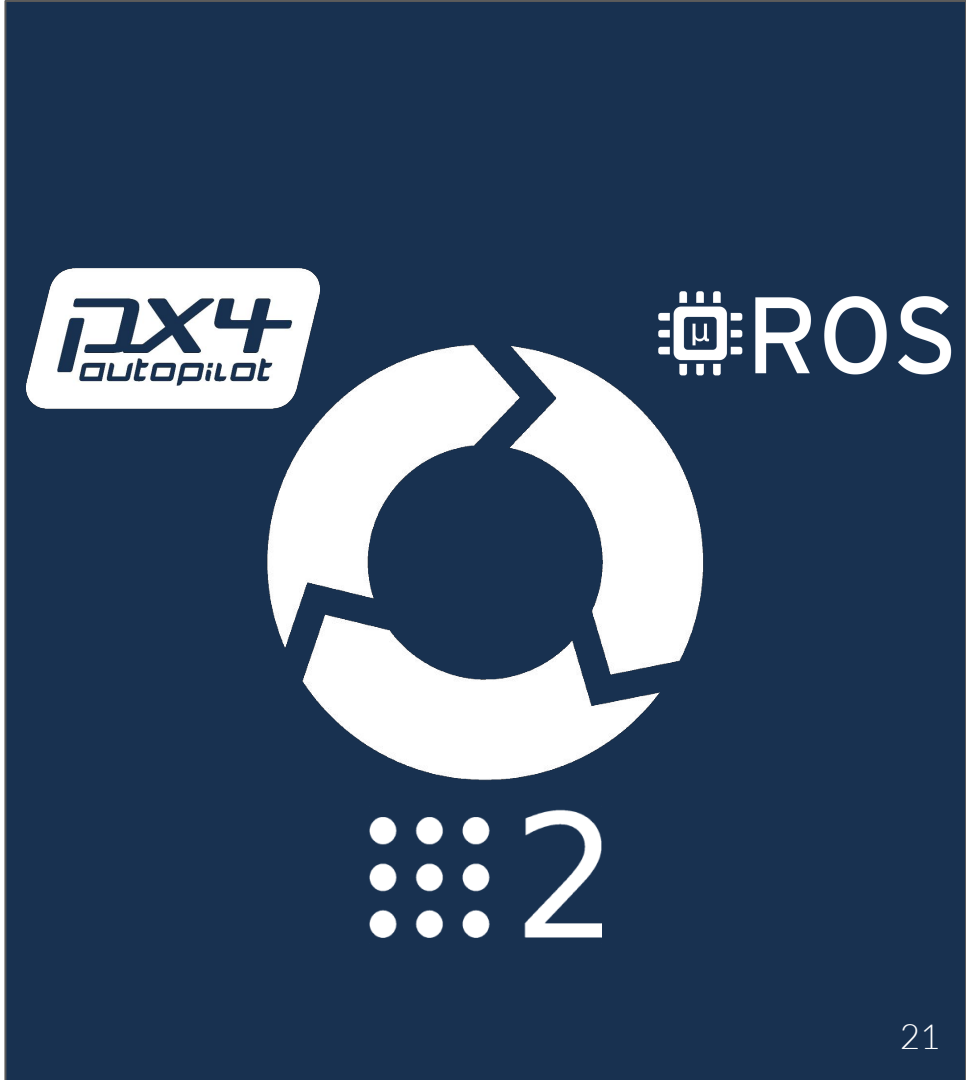
	µRTPS BRIDGE	µXRCE DDS
Streams	Only best effort, with no fragmentation handling	Best-effort or Reliable with fragmentation (HDLC) handling
Profiling	No profiling mechanism	Profiles allow to enable or disable features that influence the Client library size
Discovery	No discovery mechanism (1:1 Client-Agent communication only)	UDP multicast or unicast discovery of Agents

Why does it matter to PX4?

- Besides the advantages referenced previously...
- Reduction of the
  - Number of templates
  - Custom repos required to enable the pipeline
- We only care about how to generate and setup the Client
  - The Client will handle agent creation on the Agent
  - The Agent code doesn't have to be created anymore (using Fast-RTPS-Gen)
  - All the templates and code can be self-contained in the PX4-Autopilot repo

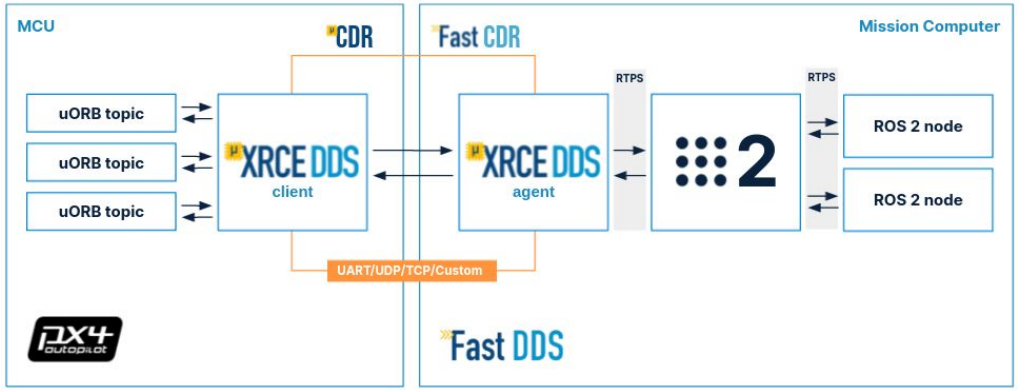
Why does it matter to PX4?

- And at last, but not least
  - We share the same middleware used by micro-ROS, which is well supported, maintained and gaining a lot of traction in the industry
  - We keep the native support to ROS 2
  - The users/developers do not have to change anything on their ROS 2 apps

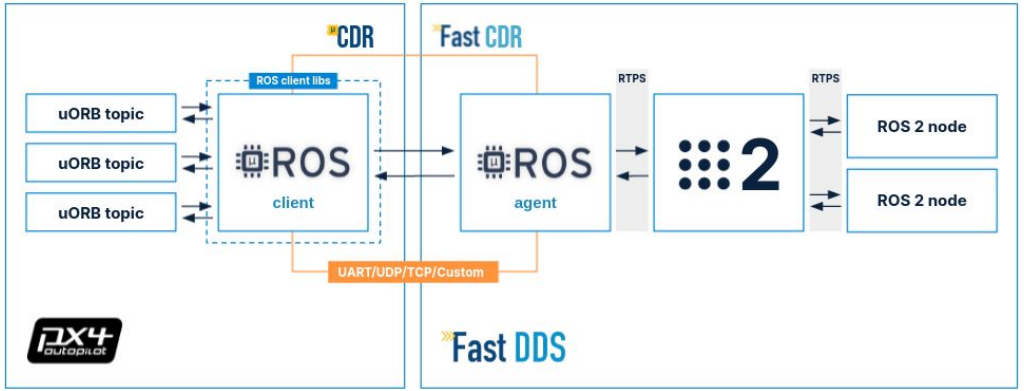


micro XRCE-DDS vs micro-ROS

XRCE DDS

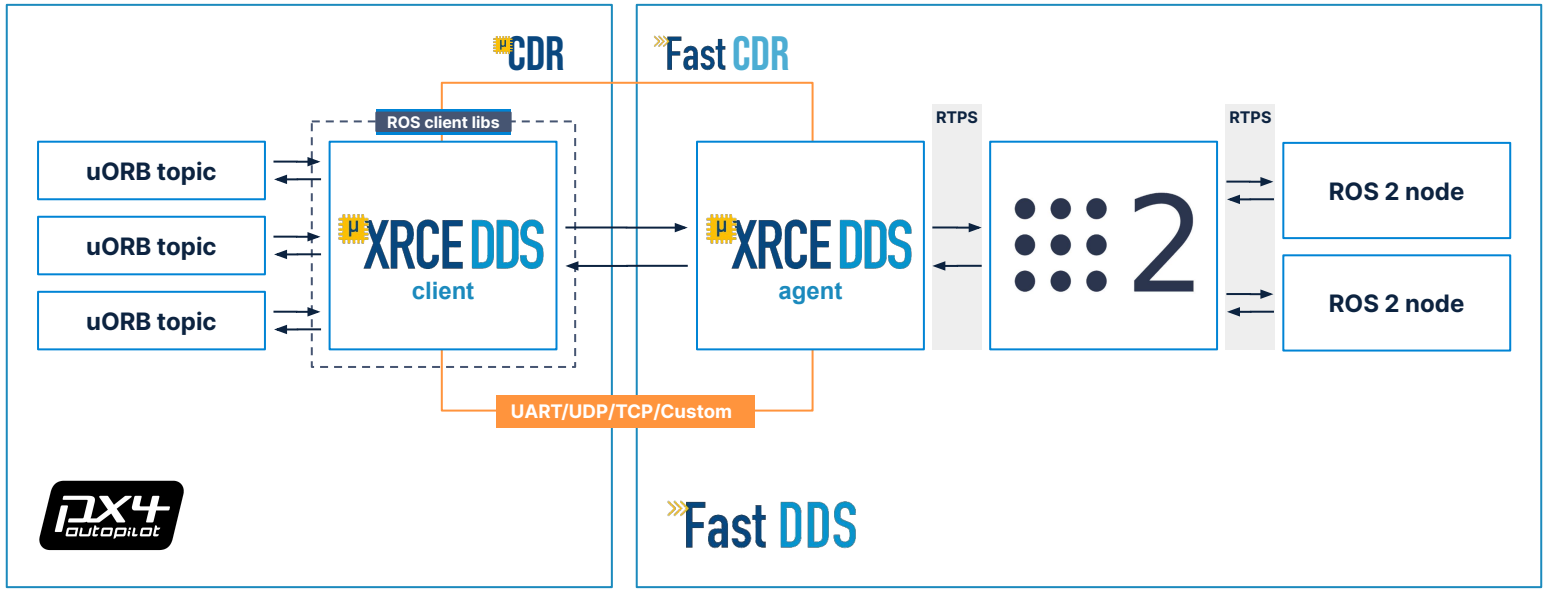


ROS



px4 autopilot ROS

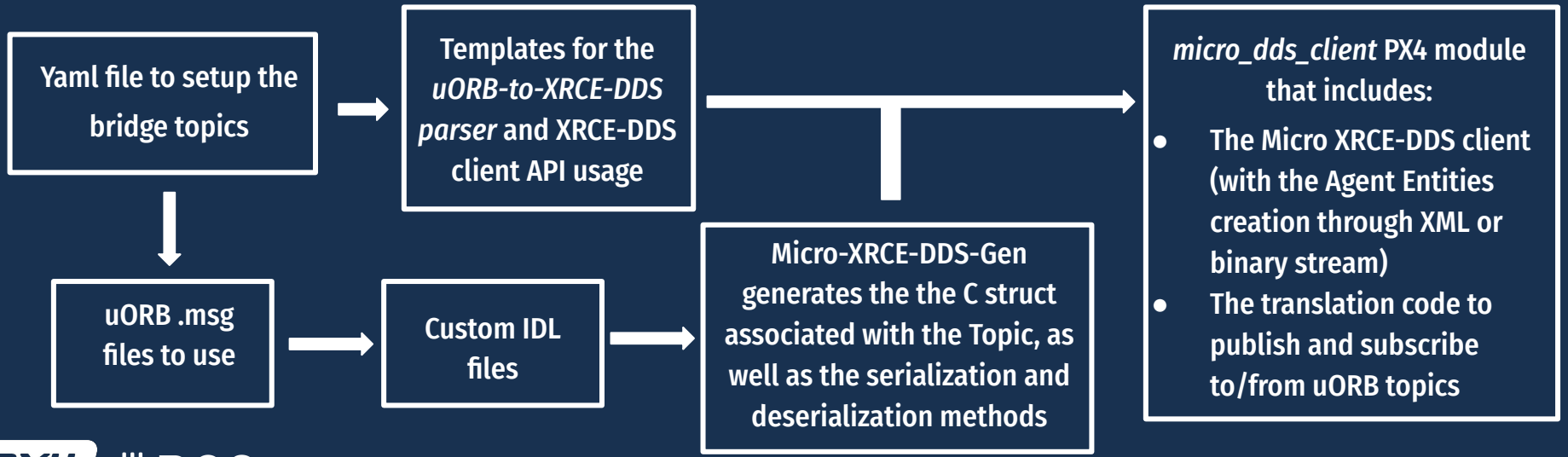
micro XRCE-DDS vs micro-ROS



Migration: what has to happen?

2 possible valid approaches:

1. Use Micro-XRCE-DDS-Gen to generate the client code

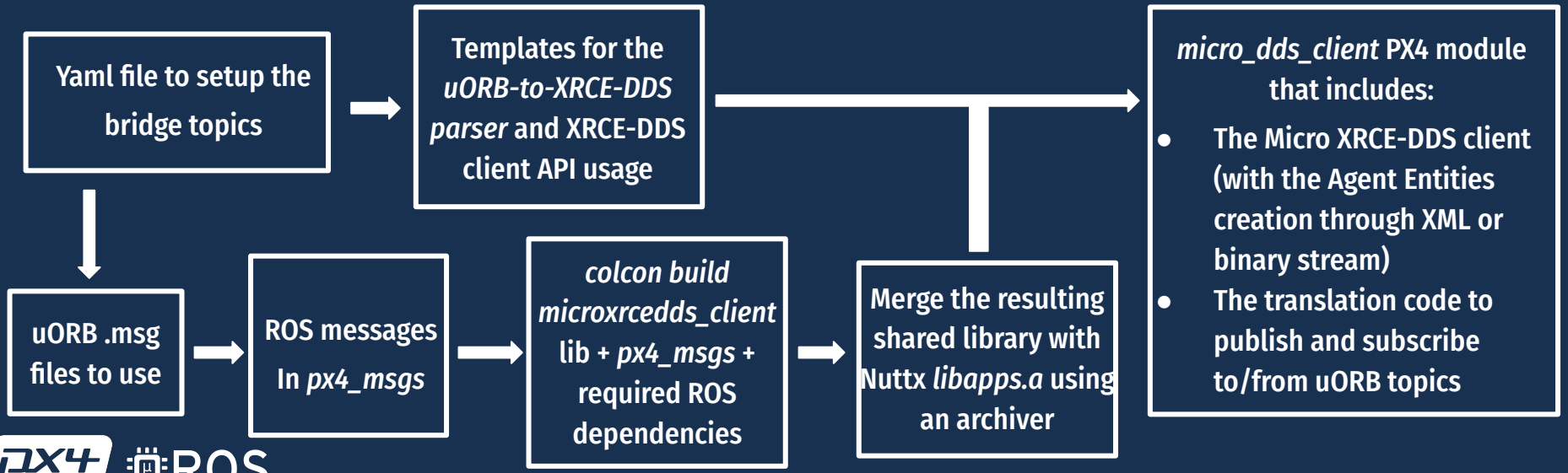


Migration: what has to happen?

XRCE DDS

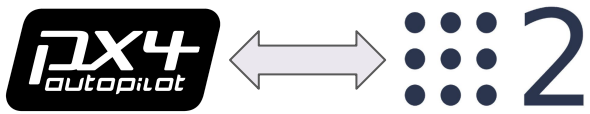
2 possible valid approaches:

2. Use the ROS 2 typesupport and IDL generator + colcon build



PX4 autopilot ROS

Migration: what has to happen?



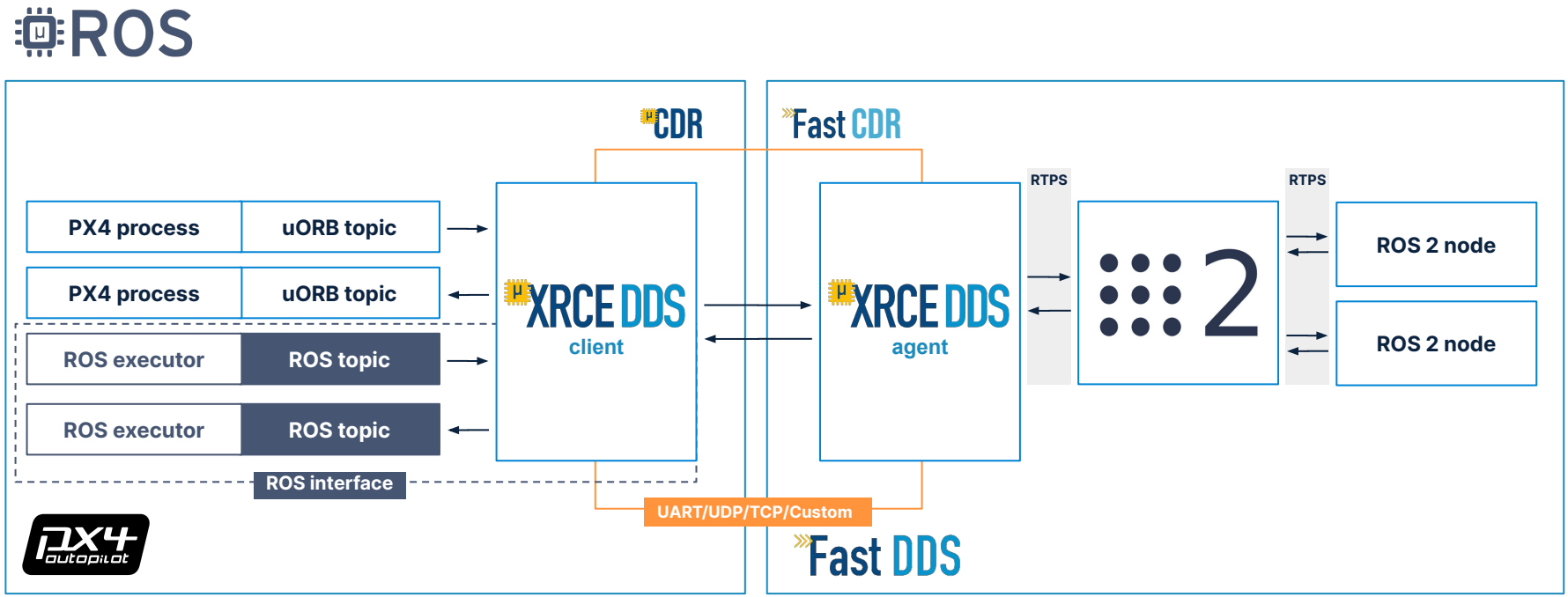
- Micro XRCE-DDS Agent and *px4_ros_com*
  - The Micro XRCE-DDS Agent lives in its separate repo
    - Can be cloned and used directly on the Mission Computer
  - *px4_ros_com* will not generate the micro-RTPS agent anymore. Instead will provide:
    - Code examples on how to interface with the topics provided by the Agent (as it is already doing, so nothing changes)
    - An high-level API/library to interface with the PX4 internals

PX4 autopilot ROS

Migration: what has to happen?

- What about micro-ROS?
  - A CMake option will enable building the entire set of dependencies to generate micro-ROS as a shared library to be added to the target board
  - The process above will be facilitated if the 2nd option presented previously is selected
  - Independently of the approach, having micro-ROS available in the FCU would be optional, as the user might not want to implement ROS executors

Migration: what has to happen?



Roadmap

July 2021

Leverage adoption - Improvements on documentation and examples repository

Leverage more examples and document them to facilitate adoption

Sep 2021

Industry/Production-ready robustness

Automatic RTPS ID assignment at compile time, bandwidth limiters, rate per topic, parallel code generation with FastRTPSGen, message definition hash check on boot, performance tests, ...

Dec 2021

Leverage adoption - ROS2 SDK for the native interface

In the process of continuously driving the adoption of ROS 2 through our native interface, we plan on providing higher level APIs so the interface with the flight controller becomes easier

Jan 2022

Start migration to micro XRCE-DDS

It will also depend on how much traction and adoption we have right for our native interface through the micro-RTPS bridge

Mar 2022

micro XRCE-DDS as the default native interface between ROS2/DDS and PX4

Examples and documentation in place.

Apr 2022

Micro ROS and micro XRCE-DDS in PX4

Micro ROS though would be optional and enabled at build time. Micro XRCE-DDS will remain the default.

Questions and Answers

SUMMARY

micro-ROS Update 01
Pablo from eProsima

Micro XRCE-DDS and micro-ROS in PX4 02
Nuno from Auterion

Questions and Answers 03

ROS 2 in PX4:
Technical details of a seamless transition to XRCE-DDS and micro-ROS

PX4 Developer Summit 2021

15 September 2021

Pablo Garrido (Lead Embedded SW Engineer @ eProsima)
Nuno Marques (Senior Robotics SW Engineer @ Auterion)