MICRO-ROS

NUTTX 2019
JULY 16TH 2019

DR.-ING. INGO LÜTKEBOHLE, BOSCH CORPORATE RESEARCH

Partially supported by EU grant 780785

Micro-ROS
The people

The original Micro-ROS Team

Your presenter today

Ingo Lütkebohle

Senior Expert SW Platforms
Bosch Corporate Research
ingo.luetkebohle@de.bosch.com

BOSCH

Micro-ROS

It‘s all about robots: A small selection from the partners

Mara robot arm by Acutronic Robotics

EOD robot by PIAP

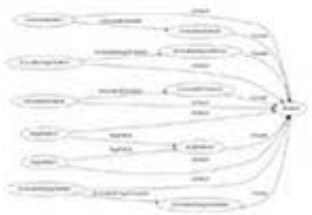
Indego lawn-mower by Bosch

BOSCH

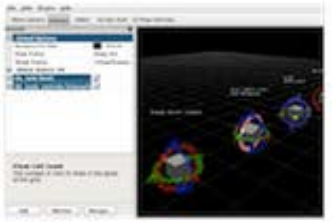
Micro-ROS
ROS: Robot Operating System

„Linux for Robotics“

=



+



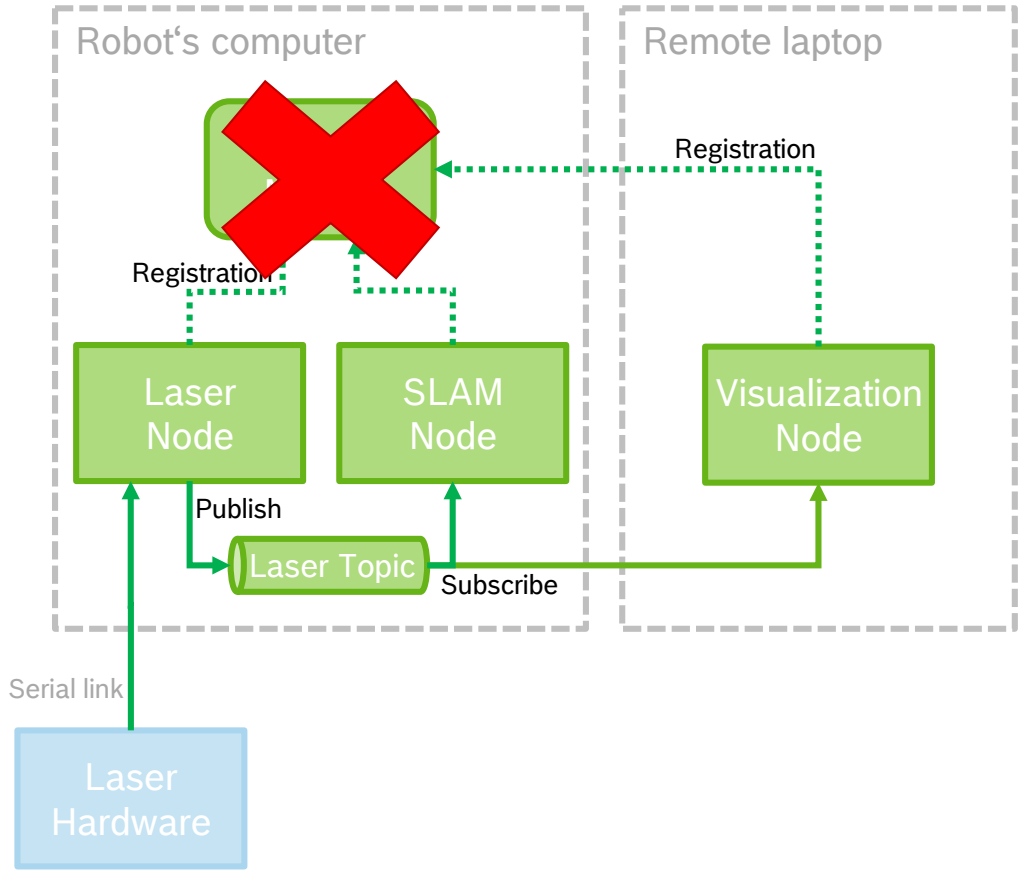
+

+

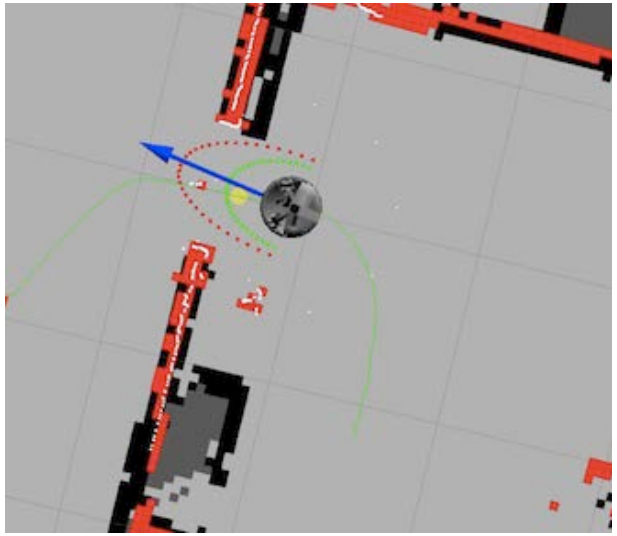
Plumbing	Tools	Capabilities	Ecosystem
- Process management	- Visualization	- Control	- Shared development
- Communication	- Simulation	- Perception	- Robot models
- Device drivers	- Data recording	- Planning	- Documentation
- Data models	- Monitoring	- Manipulation	- Exchange
- Language-independence			- Market

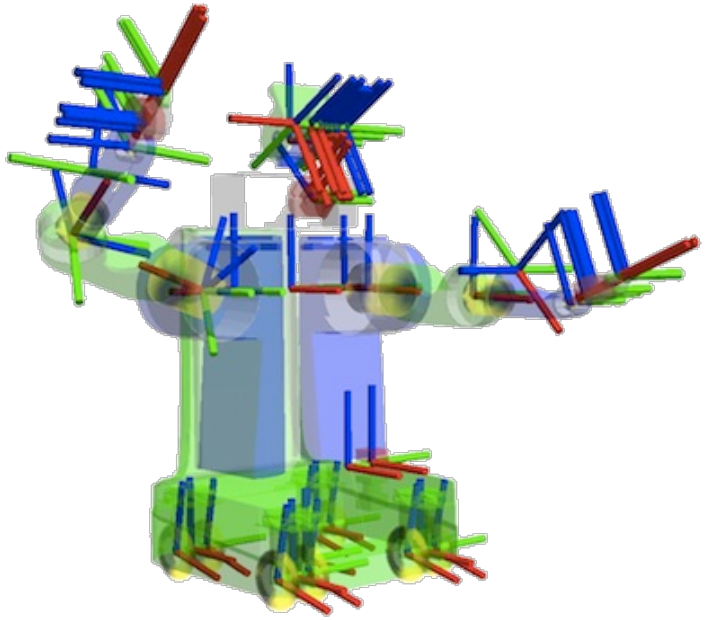
Micro-ROS
Architectural Principles of ROS (2)

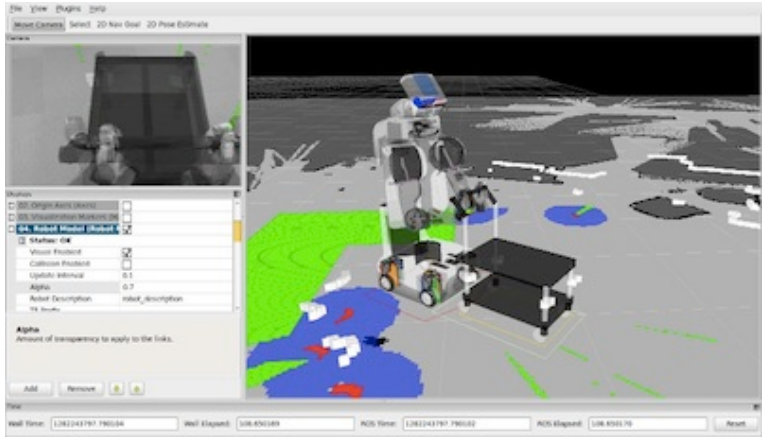
- Basic entity: *Nodes* that exchange messages
  - Default protocol is TCP, UDP also possible
  - *Nodelets* for in-process communication
- Can be distributed across machines
- Standard communication patterns
  - *Topics*: Publish-Subscribe(1 – n, uni-directional, async)
  - *Services*: Request-Response (n – 1, bi-directional, sync+async)
  - *Actions:* Advanced Request-Response (1-1, multi-state)
- Nodes comprised of *callables* (functions), which are *data-* or *time-triggered*
  - Implemented in C++, Python, ...
  - Run-to-completion

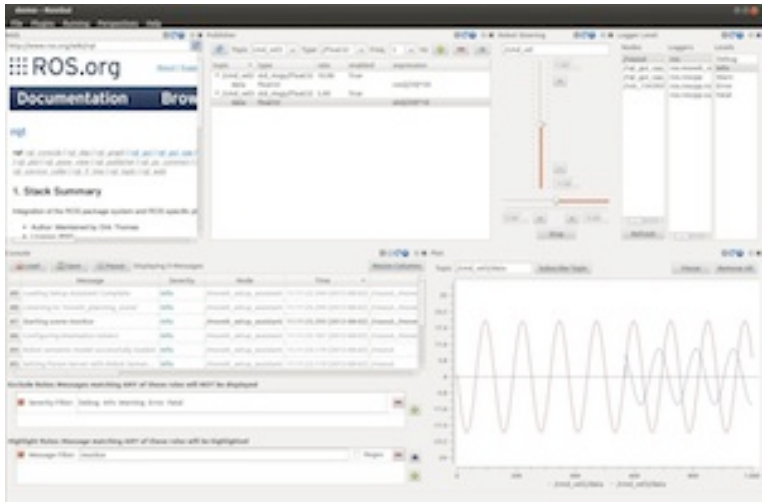


Micro-ROS
Selected Core Features









http://www.ros.org/core-components/

BOSCH

Current Status of ROS 2 - Hands-on Feature Overview

Historic Milestones

2007 Pre-cursors of ROS created at Stanford University

2008 Willow Garage, Inc takes over ROS development and starts internship program for PhD students

2010 ROS 1.0 released and eleven PR2 robots donated to universities/institutions (including Bosch)

2013 Maintenance moves to Open Source Robotics Foundation (OSRF)

ROS Industrial Consortium founded

2014 Willow Garage shuts down, but has seven spin-offs
ROS Industrial Consortium Europe (RIC-EU) founded

2015 OSRF starts design of ROS 2

2016 ROS Kinetic, the 10th release, is launched

2017 ROS 2 V1.0 (named Ardent Apalone) released

Colleagues with PR2 robot at the Bosch RTC in Palo Alto

https://spectrum.ieee.org/automaton/robotics/robotics-software/the-origin-story-of-ros-the-linux-of-robotics

Current Status of ROS 2 - Hands-on Feat
Releases – and How to Get ROS

- Annual release
- Long-term releases together with Ubuntu LTS

- Debian packages at **http://packages.ros.org/ros/ubuntu**

- Installation
  - Instructions at **http://wiki.ros.org/ROS/Installation**
  - Full installation: `sudo apt install ros-[distro]-desktop-full`
  - Single packages: `sudo apt install ros-[distro]-[packagename]`

Distro	Release	Poster	EOL
Melodic Morenia	*May 2018*		*May 2023*
Lunar Loggerhead	**23 May 2017**		**May 2019**
Kinetic Kame	**23 May 2016**		**May 2021**
Jade Turtle	23 May 2015		May 2017
Indigo Igloo	22 Jul 2014		April 2019
…	…	…	…

BOSCH

“In the future it should be possible to implement the ROS protocol directly on the devices embedded system”

ROS2 Design Wiki “Stories”

Micro-ROS

Robots are networks of devices

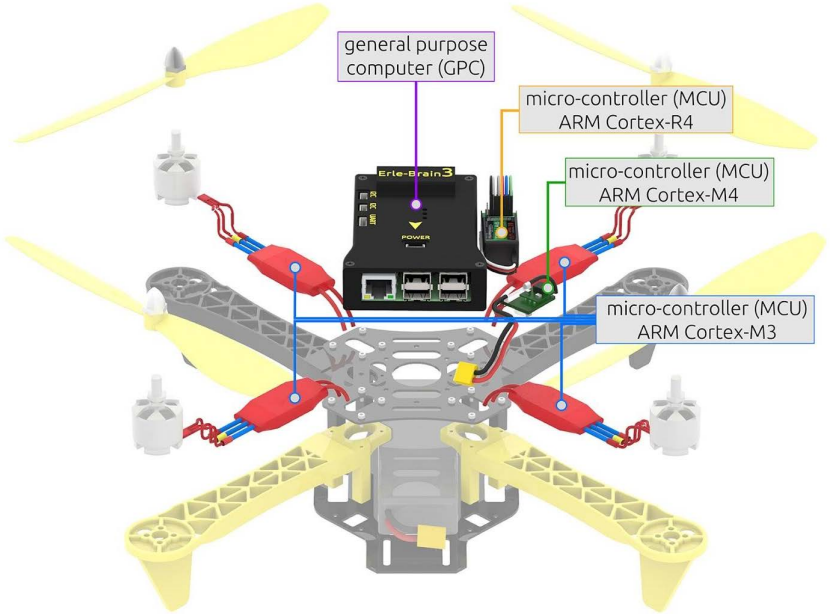


Image source: Erle Robotics, taken from OFERA proposal.

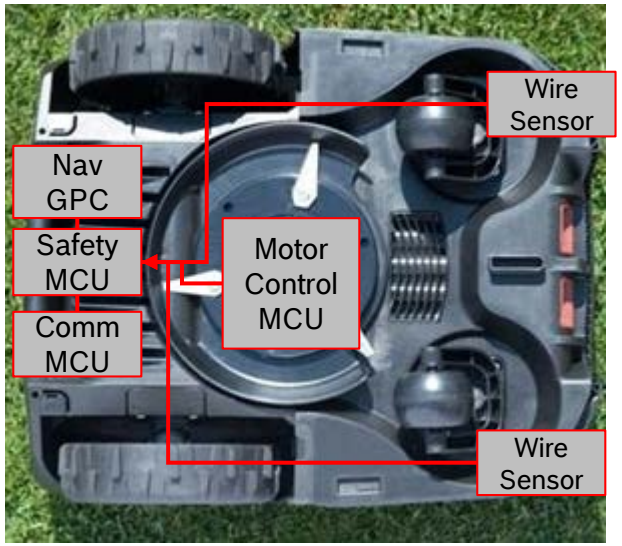
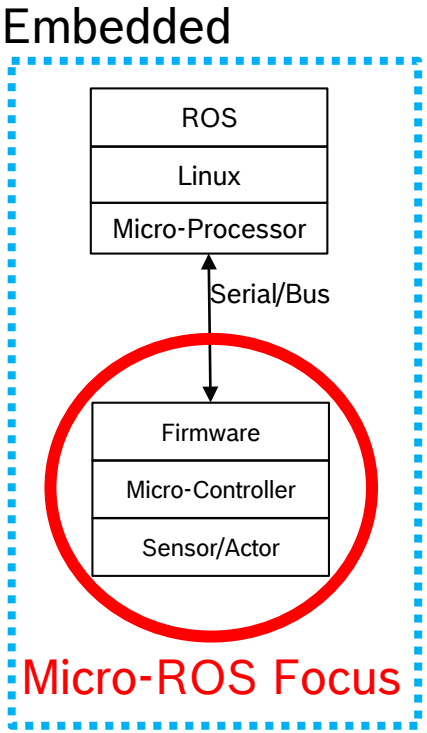


Image source: Bosch PowerTools GmbH, All rights reserved

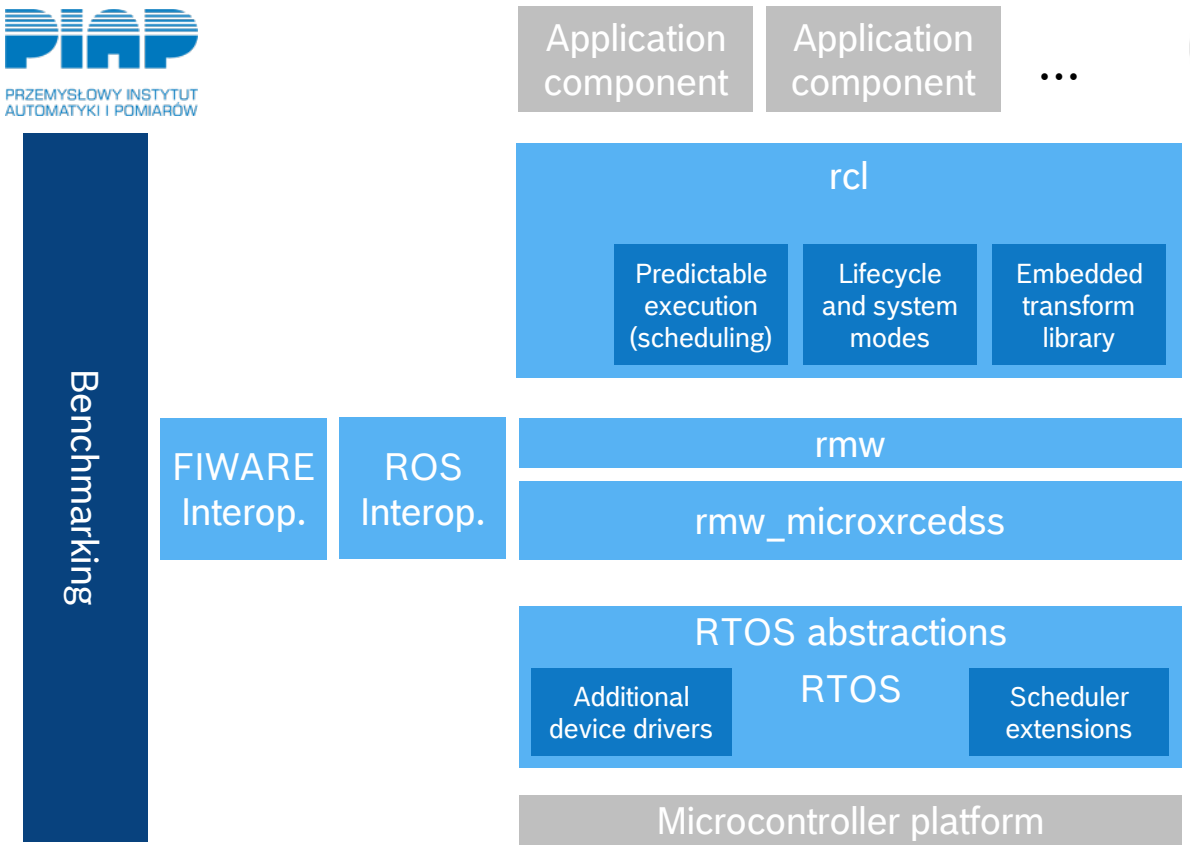


Micro-ROS
Open Framework for Embedded Robot Applications (OFERA)

OFERA will extend ROS2 to allow its use in MCUs

https://ofera.eu/

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



Micro-ROS
Situation

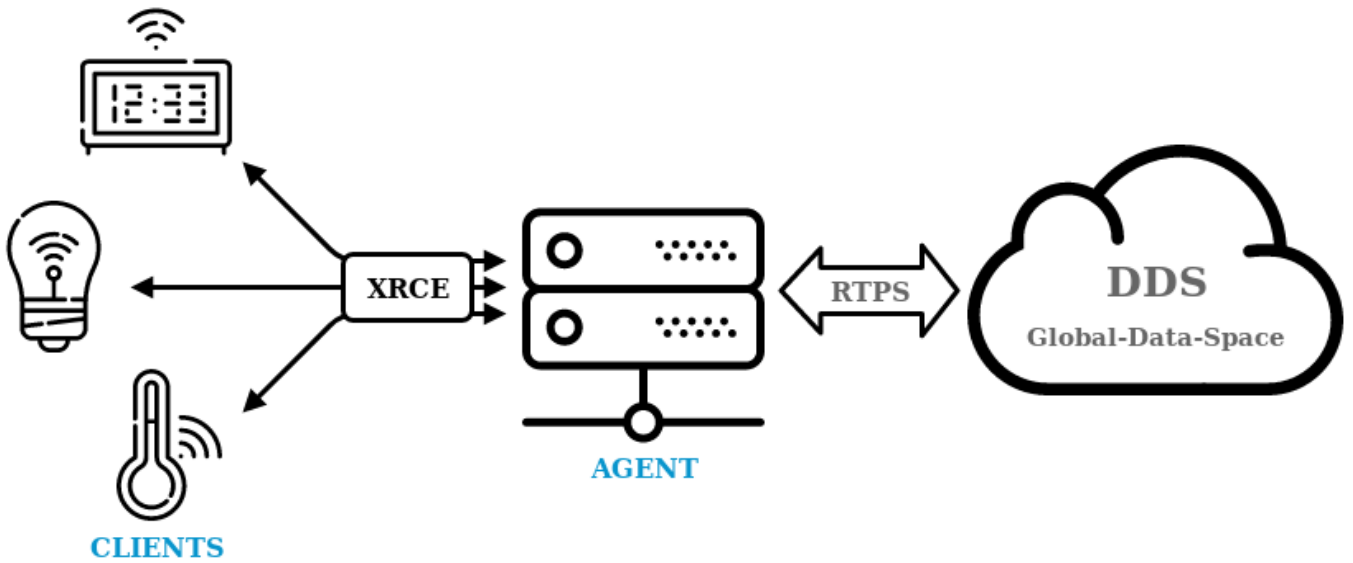
- Robot application development happens on Linux and Windows

- ROS+Linux is a powerful combo
  - Excellent libraries for perception, planning, networking, etc
  - Unified developer eco-system: One kernel, most devices
  - It's what we all have on our desks
- But…
  1. Peripheral access…Hard, low-latency RT… Power saving … Safety
  2. Open Source Firmware! → composable modules

Micro-ROS
NuttX

- OFERA has chosen NuttX as the default RTOS
  - Primary reason: POSIX-style API makes porting easy
  - Secondary reason: Linux-like in many respects
- Contenders
  - Prior work has also used the RIOT OS
  - Zephyr is interesting for safety-rated applications
  - FreeRTOS has a large userbase

Micro-ROS
New DDS-XRCE Standard



- DDS-XRCE for e**X**tremely **R**esource **C**onstrained **E**nvironments
  … brings DDS on MCUs
- Client-server approach
  - Power-saving
  - Disconnected use

Open-source at github.com/eProsima/Micro-XRCE-DDS

Micro-ROS
Side-by-Side Comparison

	ROS2	Micro-ROS	XRCE++
Hardware	X86, ARM Cortex-A, …	ARM Cortex-M, ….	
Resources	>512MB RAM, >8G Disk	~100K RAM, ~1MB Flash	~10k RAM
Communications	GBit/s: Ethernet, 802.11 WiFi	Serial, WPAN – 250k to 1MBit/s	←
Operating System	Linux, Windows, MacOS	RTOS (NuttX by default)	any
Middleware	DDS variant (by default)	XRCE-DDS (by default)	XRCE-DDS
Middleware Abstraction	RMW	RMW	-
Client Support Library	RCL	RCL	-
Execution Layer	RCLCPP / RCLPY / …	**RCL** + RCLCPP	Micro-ROS
Executors	Generic	Micro-ROS custom	Micro-ROS

CURRENT WORK

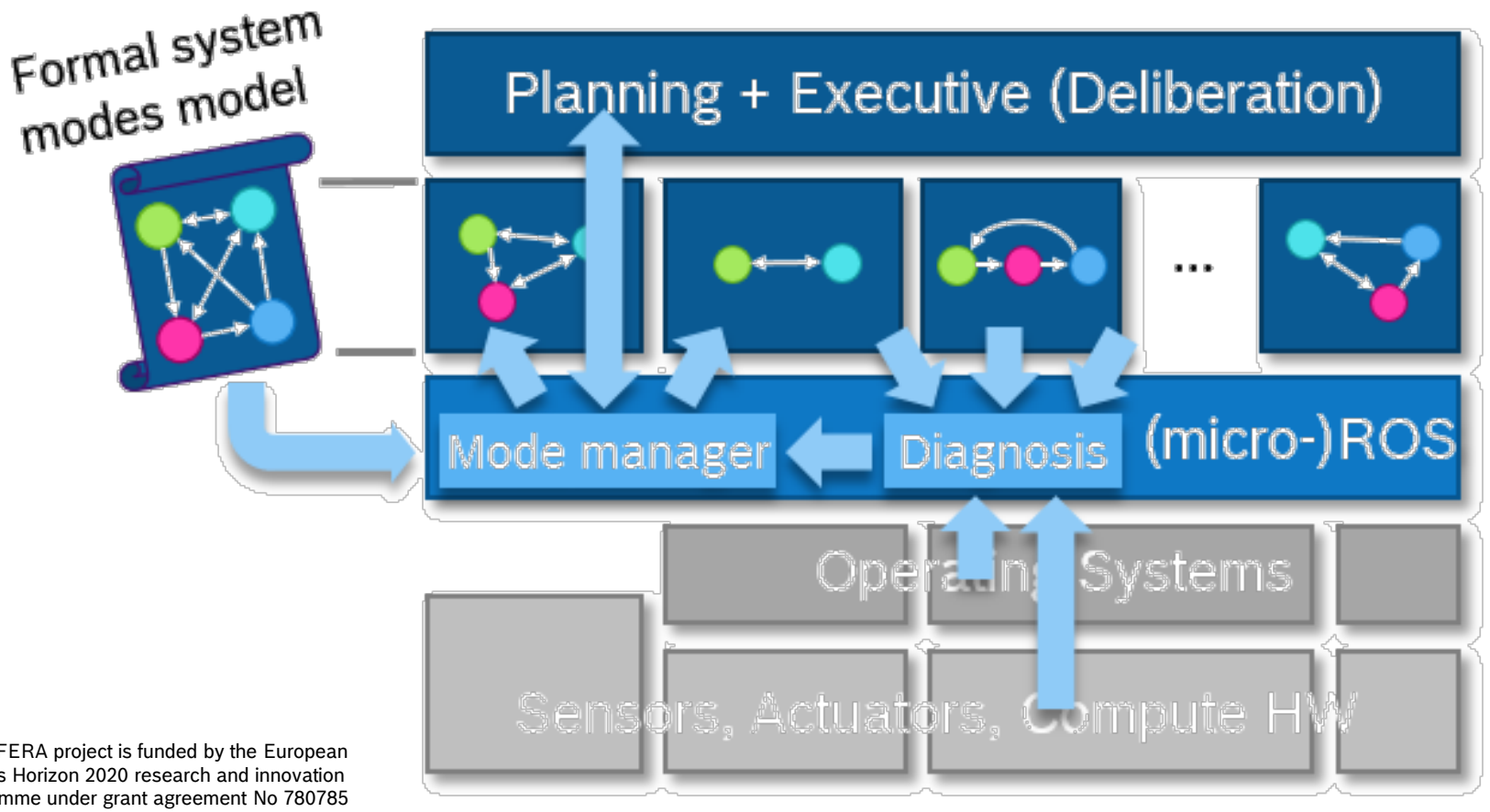
BOSCH

Micro-ROS
Recap: Composable firmware

- Nowadays, firmware provided by vendor
  - Unforeseen features? Bad luck…
- Vision: Add new features to existing firmware
  - ROS2 way: Just add nodes
- Challenge: Interference
  - Need to make sure existing stuff still works!

- Micro-ROS Approach:
  - System Modes
  - Domain-specific scheduling, towards providing guarantees

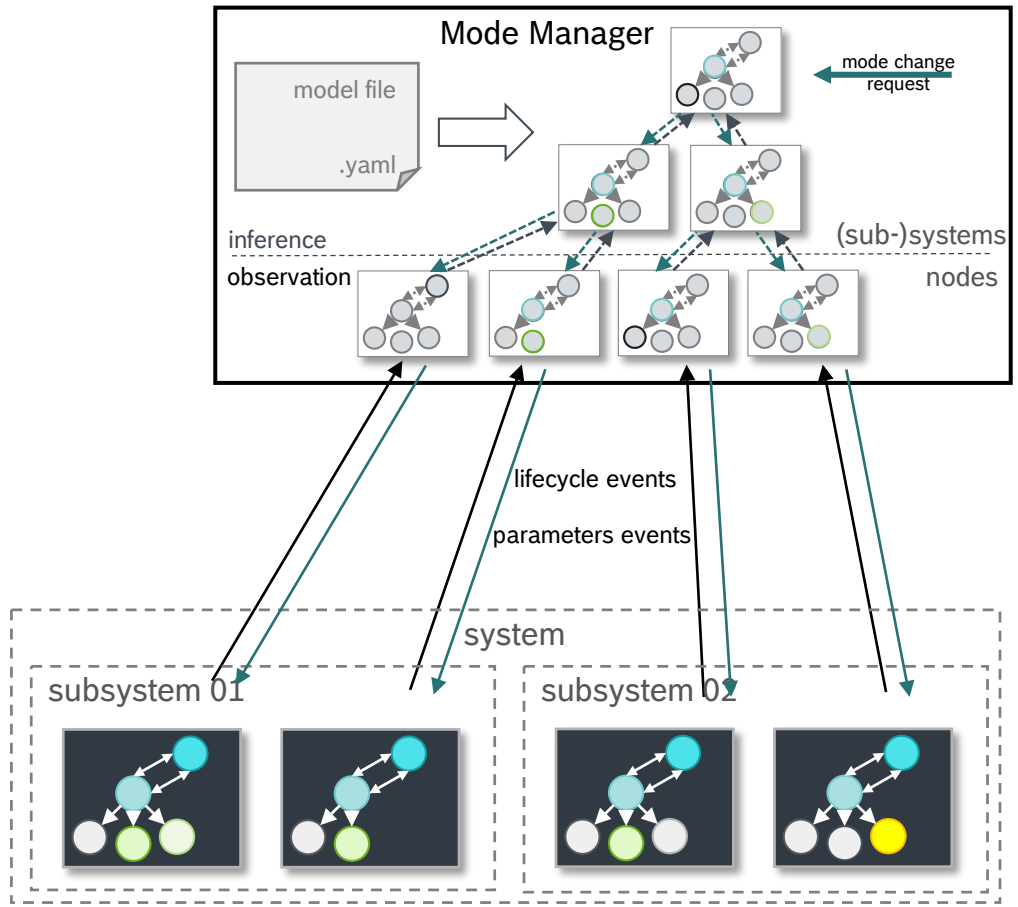
Micro-ROS
Towards explicit architecture



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

Micro-ROS
System Modes

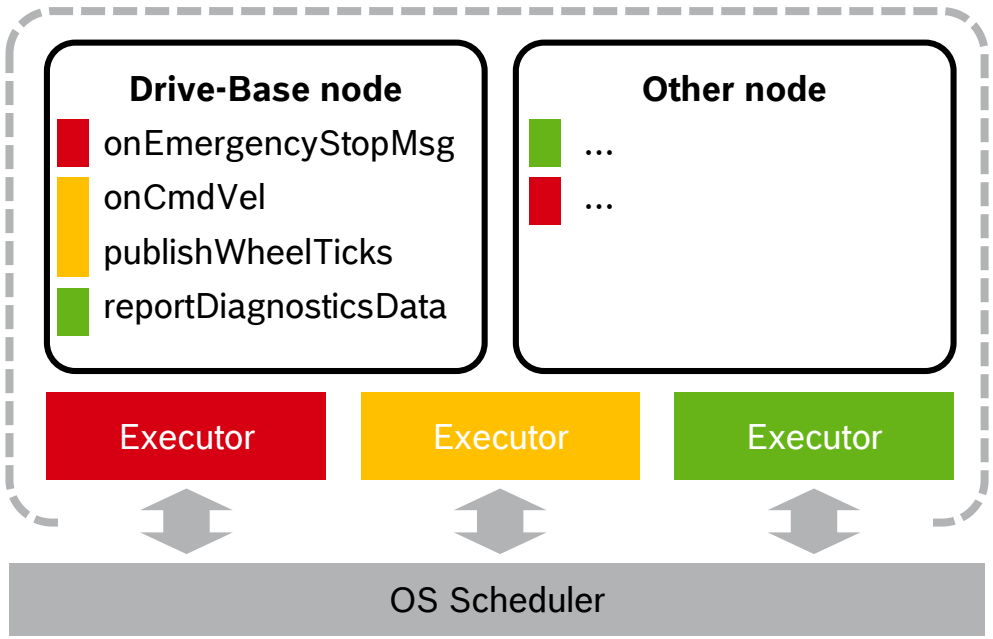
- Introduces **(sub-)systems** hierarchy to ROS 2
- Abstraction for hierarchical configuration, called **system modes**
- **Mode manager** manages consistent, system-wide configuration
- See **microros.github.io/system_modes/**



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

Micro-ROS
Predictable Execution

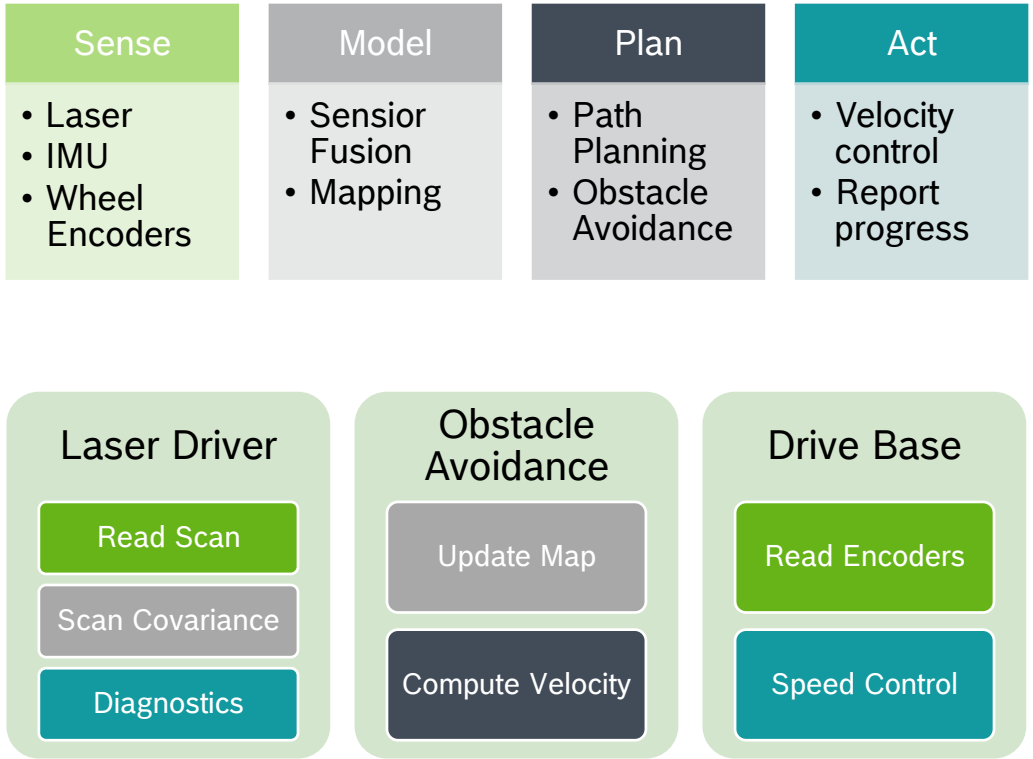
- First approach enables **multiple executors** per operating system process
- Executors can be configured individually using standard scheduling mechanisms
- Open-sourced prototype for ROS 2
- See **microros.github.io/real-time_executor/**



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

Micro-ROS
Domain-specific scheduling

- Current real-time schedulers typically use priorities
  - Not composable!
  - Not domain-appropriate
- Micro-ROS Approach: Domain-specific schedulers
  - E.g., stage-based approach with „Sense-Plan-Act“
  - Or more stages…
  - Assign callbacks to stage using callback groups
  - Derive within-group order from communication links
- Provide „budgets“ by group



BOSCH

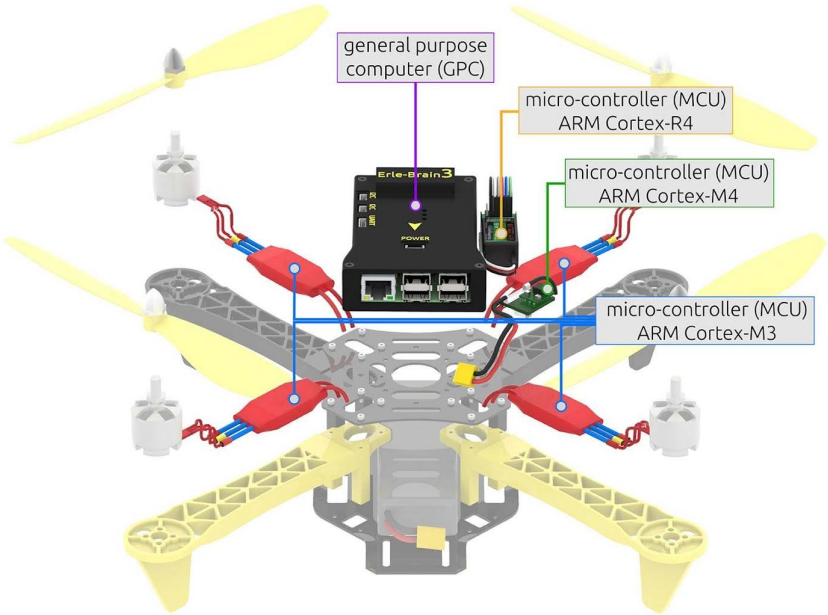
Micro-ROS
NuttX Feedback: My personal journey

- I had no NuttX, or any RTOS, experience before micro-ROS
- NuttX was chosen by a partner active in the drone community
- At first I was indifferent
- Then I became annoyed with the configuration process
- …and more annoyed
- And started to doubt whether we would ever be able to make micro-ROS user-friendly using NuttX
- …but the alternatives were even worse
- Then I got more experience and started to like some aspects
- To the point where I started recommending it
- But I‘m still unsure whether we can put this into a product

Micro-ROS
NuttX Feedback: Why it was chosen

- Good track record in the drone community (PX4)
- Developer ecosystem
  - POSIX API
  - TCP/IP stack
  - Fairly complete C library
  - C++ support
  - Promises ability to work on multiple OS‘s
- Good support for STM32-based boards

- „Feels like tiny Linux“



Dronecode

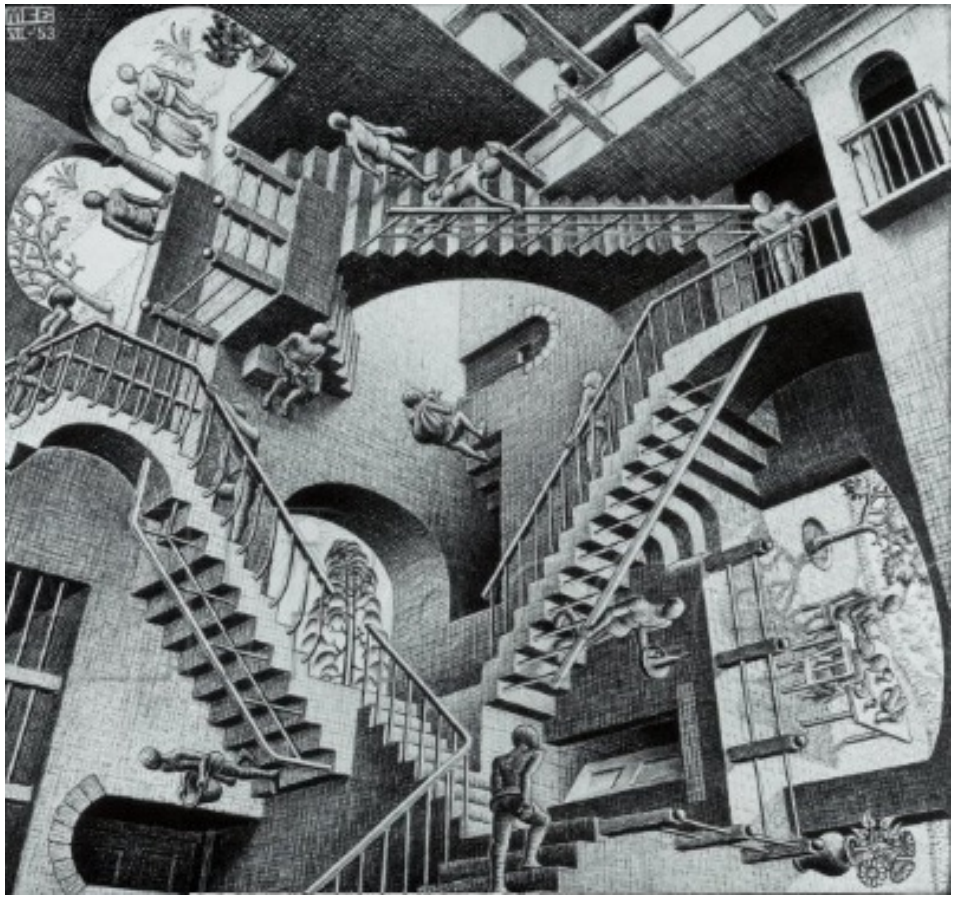
BOSCH

Micro-ROS
…but the configuration…

- NuttX could do a lot of things
  - If you would only find them!
    - Thank god for „/“!
  - And if you‘ve got the right board

- but this whole „upper half“/“lower half“ stuff…

→ and then it fails to compile…



Relativity, MC Escher, 1953

Micro-ROS
Hardware support

- Many boards are supported
  - But not fully!
  - Reduces trust
- This is a general problem with RTOS‘s
  - Maybe only Open Source ones?
- How about model-driven code generatic

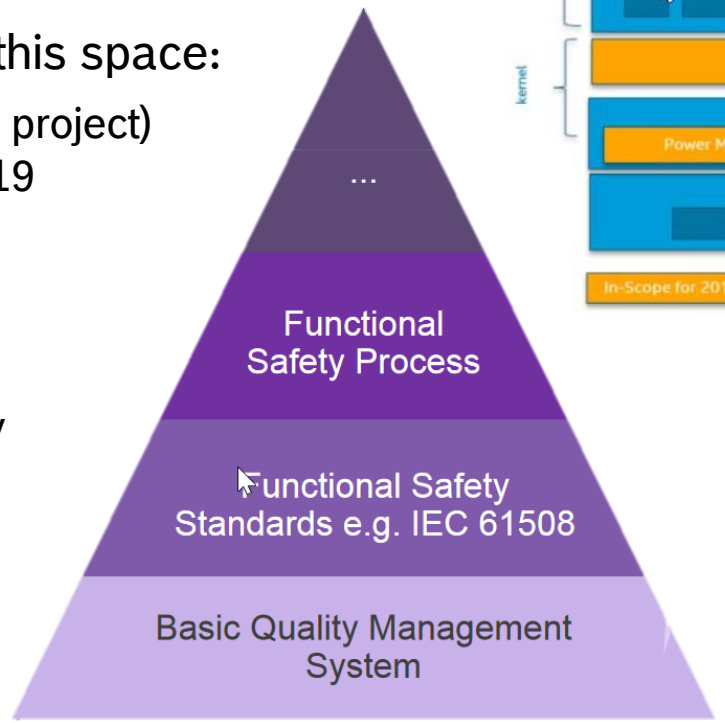
Micro-ROS
POSIX or not POSIX? Abstraction or not?

- Sometimes, the POSIX API is inferior
- Example: timers
  - clock_gettime only supports real-time clock
  - Pre-scalers are not configurable

  → low resolution
  - Dedicated timer API much more capable – but not POSIX
- File-system mapping of devices feels awkward at times
  - Direct reading and writing of registers is one of the attractions of microcontrollers (for me)
  - C++ template mechanisms could make this safe

Micro-ROS
The stumbling block: Safety

- We need safety for many applications
  - Currently using proprietary RTOS's
- Only one Open Source RTOS in this space:
  - Zephyr RTOS (a Linux Foundation project) attempts Safety Certification in 2019
  - Subset of whole OS
    - Orange boxes: In scope for 2019
    - Notably no drivers!
  - Based on existing work on security



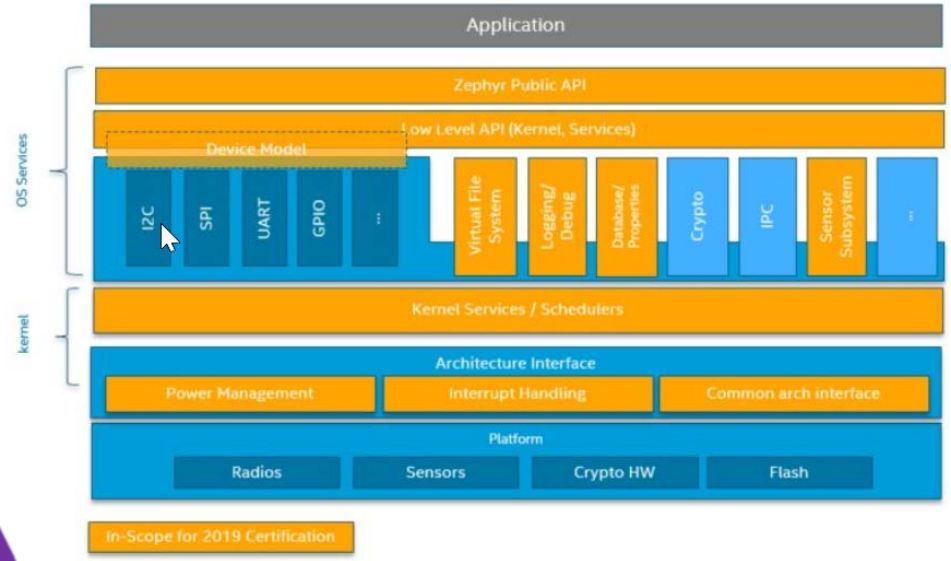
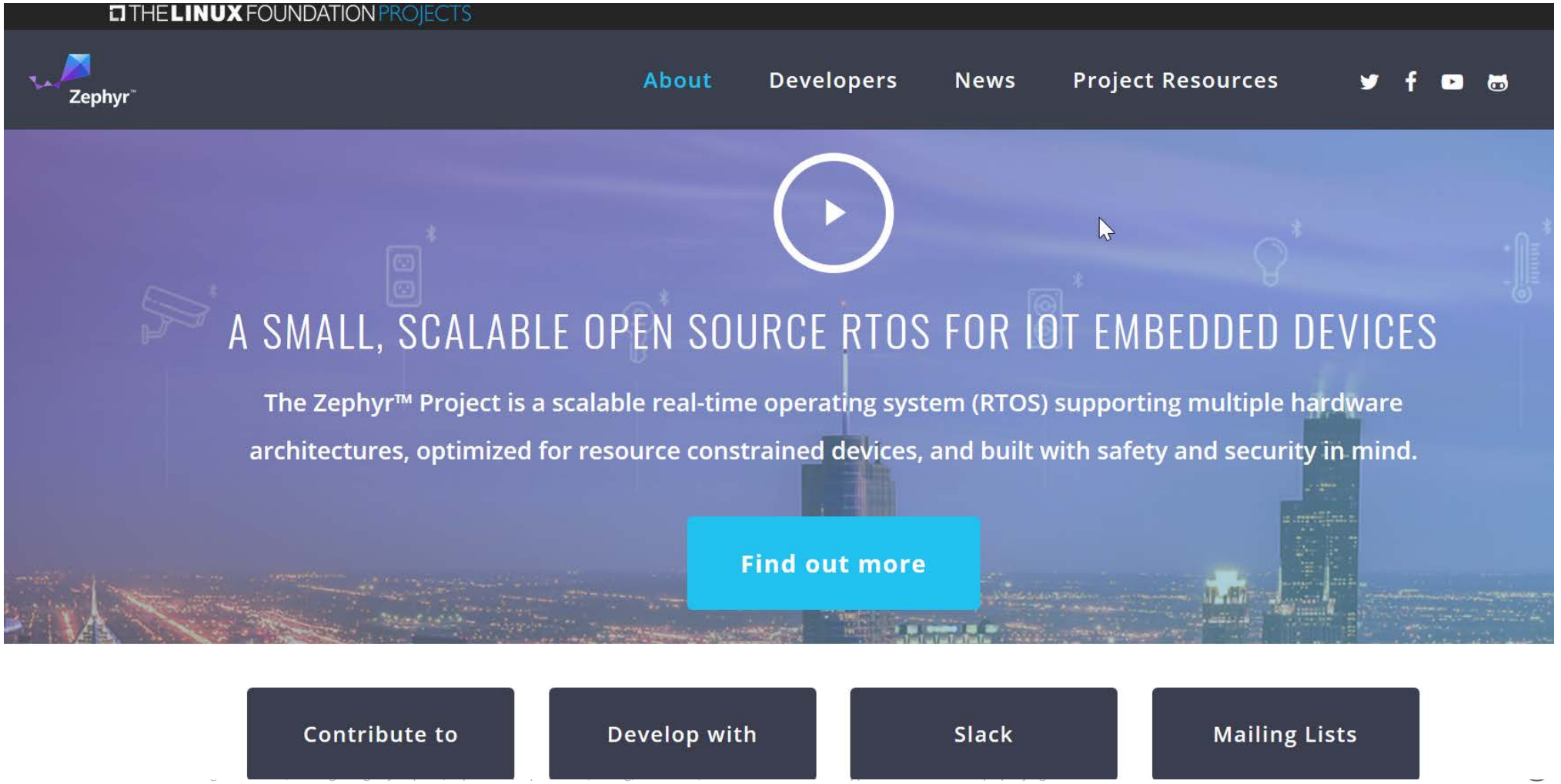


Image source: https://events.linuxfoundation.org/wp-content/uploads/2018/07/OSLS-2019_-Zephyr-Project-.pdf

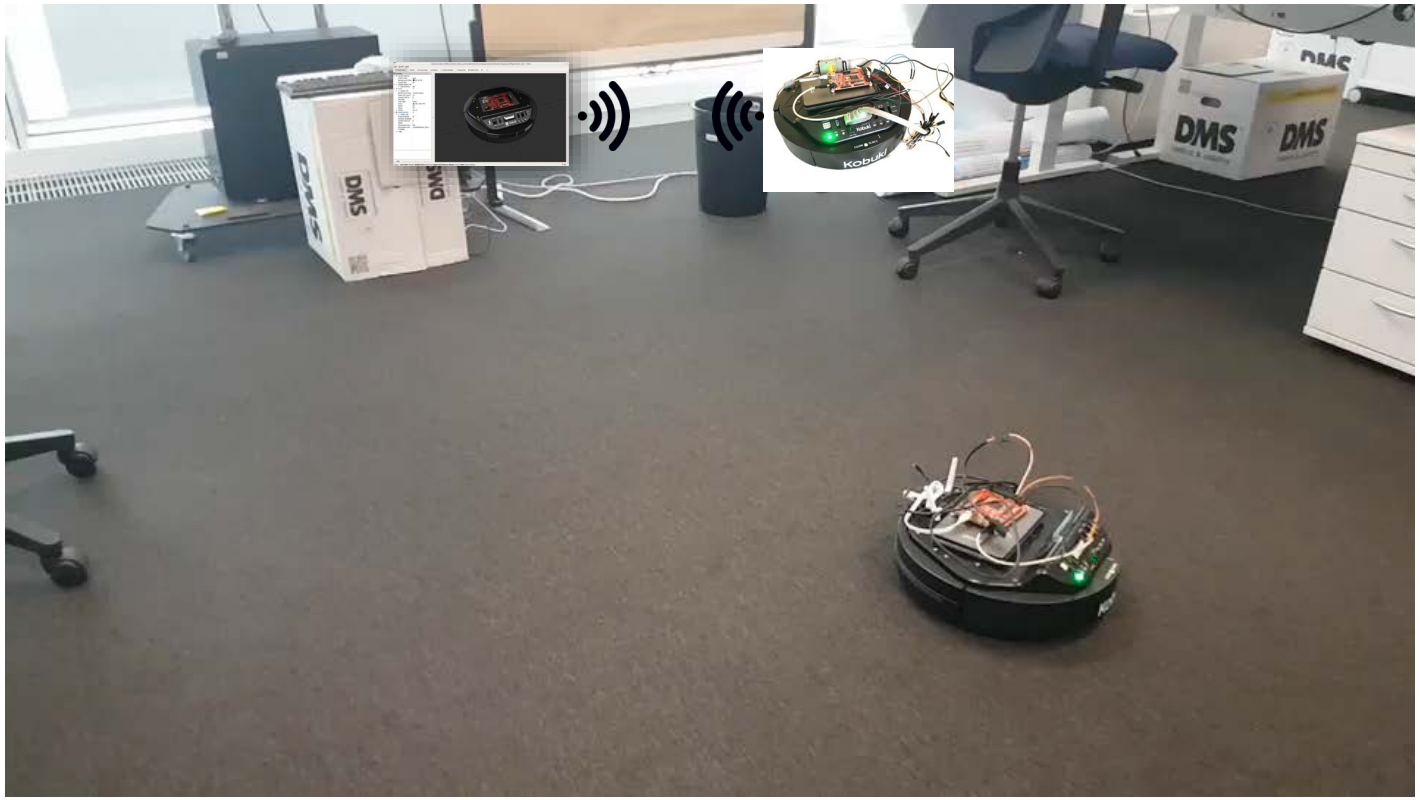
Micro-ROS
On foundations…



BOSCH

Micro-ROS
Turtlebot 2 Demo

- Based on „thin kobuki“ driver
- Converted to use rcl API
  - rclcpp wasn‘t ready at the time
- Porting issues?
  - A few issues with C++ initialization



THANK YOU!

QUESTIONS?

https://micro-ros.github.io/

Micro-ROS
Building an ecosystem

- Does this mean that every ROS developer can now start using MCUs?
- Well...

ROS 2 Embedded
Further information

- microROS organization at GitHub
  - https://micro-ros.github.io/
  - https://github.com/micro-ROS/
- OFERA website: https://ofera.eu/
- ROS 2 Embedded Design Page
  - Currently at https://github.com/ros2/design/pull/197
  - After merge: http://design.ros2.org/articles/embedded.html

THANK YOU

BOSCH