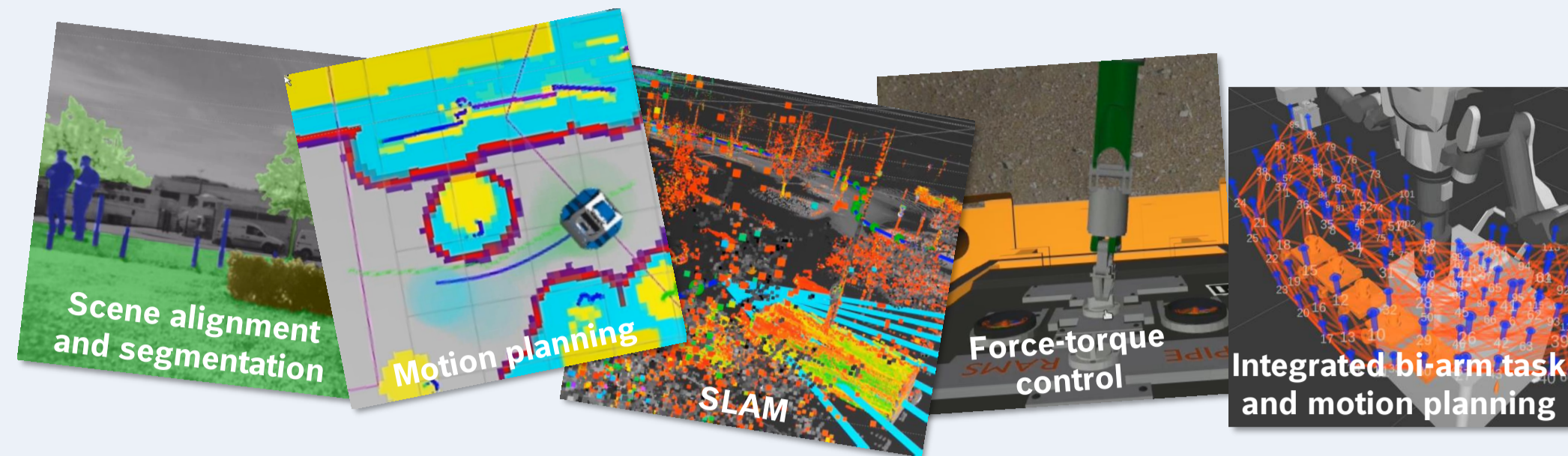


Practical and Easy-to-Use Real-Time Execution Mechanisms for ROS

Situation



Diverse mix of real-time and non-real-time algorithms in typical robotic applications

- ▶ Rates from sub-hertz to kilohertz
- ▶ Computing times from few μ s on MCUs ...
... to several 100 ms on dedicated GPU hardware

Small market volumes in service robotics

Small teams

- ▶ No dedicated real-time experts

Frequent use of third-party software with complex dependencies

- ▶ Unknown models of computation

Integration of sensors and actuators with own computing hardware

- ▶ Distributed computing platforms, even in small consumer products

- ▶ Diverse communication protocols and heterogeneous time/clock mechanisms



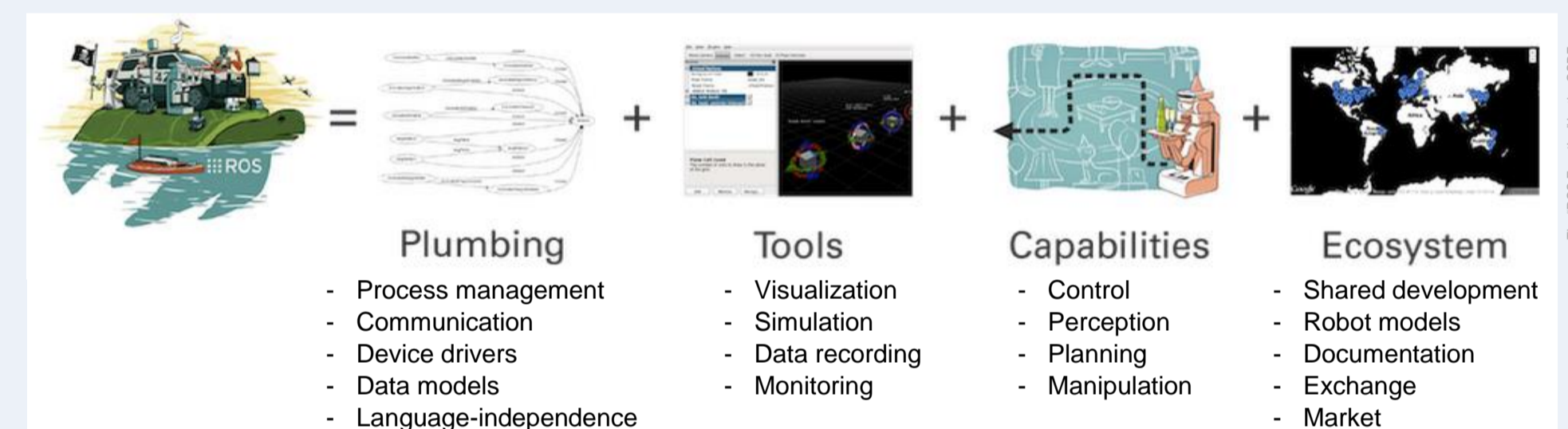
Challenge

How to support roboticists with practical and easy-to-use real-time execution mechanisms?

- ▶ Timing analysis of existing software
- ▶ Modeling of requirements
 - ▶ Latencies, determinism in subsystems, ...
 - ▶ Two-stages: component-level and system-level
- ▶ Mapping to ROS framework and OS scheduler
 - ▶ Semi-automated, optimized (e.g. latencies) mapping desired
 - ▶ Generic, well-understood framework mechanisms required
- ▶ Synchronization, possibly with external events
- ▶ Monitoring and contingency handling

Technical Basis

ROS – Robot Operating System



Huge open-source undertaking with a vibrant global community

openrobotics.org

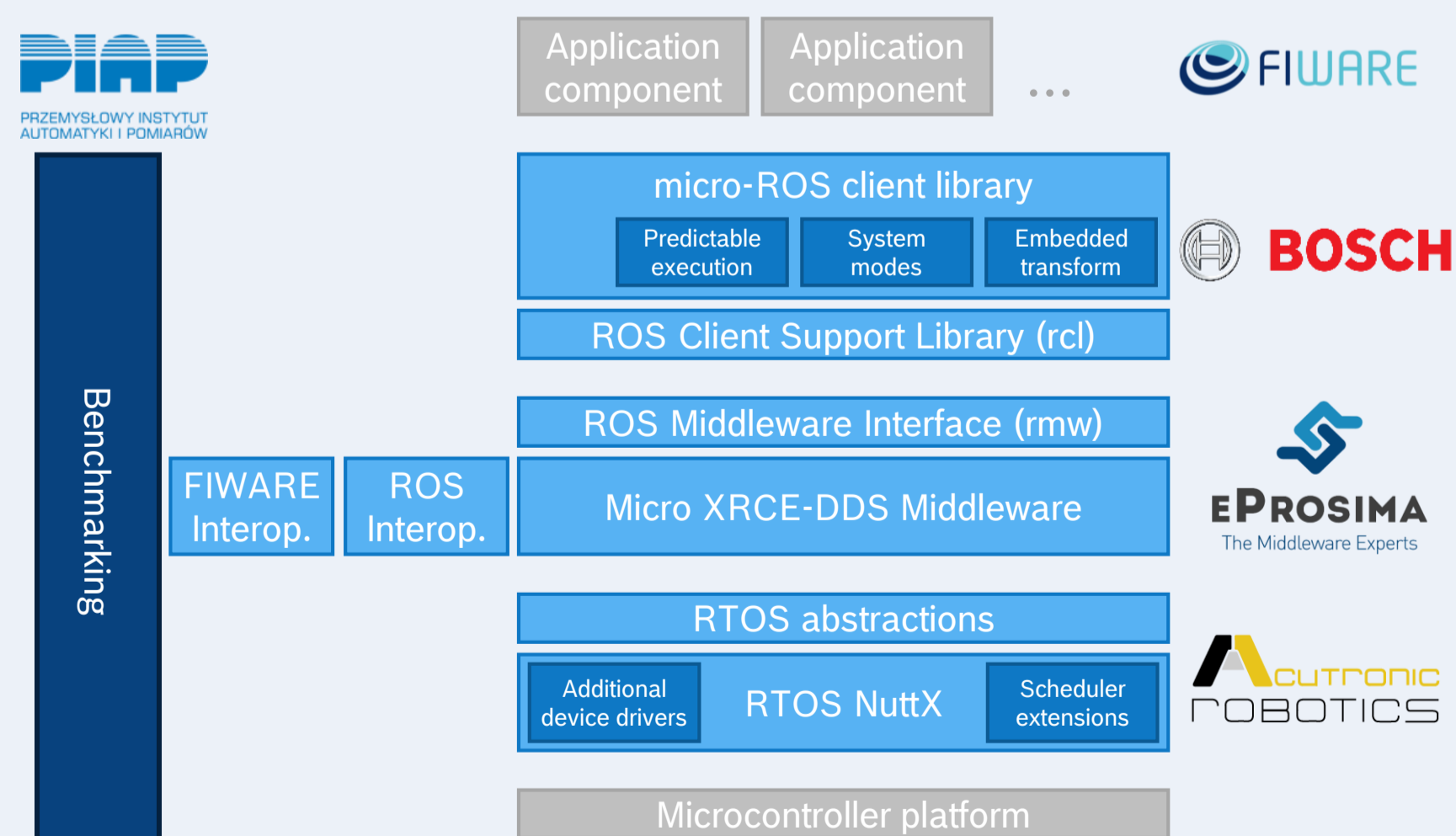
ros.org

micro-ROS

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



... put ROS 2 on microcontrollers

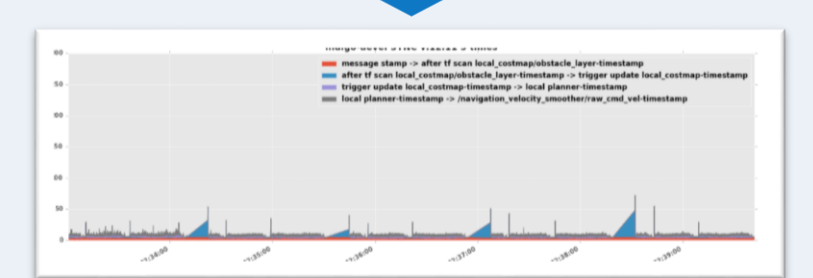
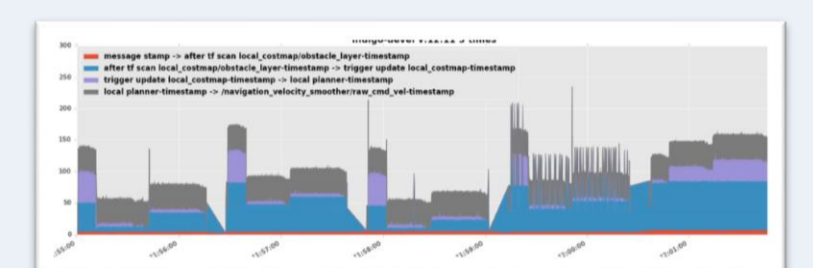


- ▶ ROS event model \leftrightarrow typical RTOS scheduling
 - ▶ rcl Executor with logical execution time (LET) semantics
- ▶ Model-based, deterministic runtime reconfiguration using *system modes*

micro-ros.github.io

Runtime Tracing for ROS

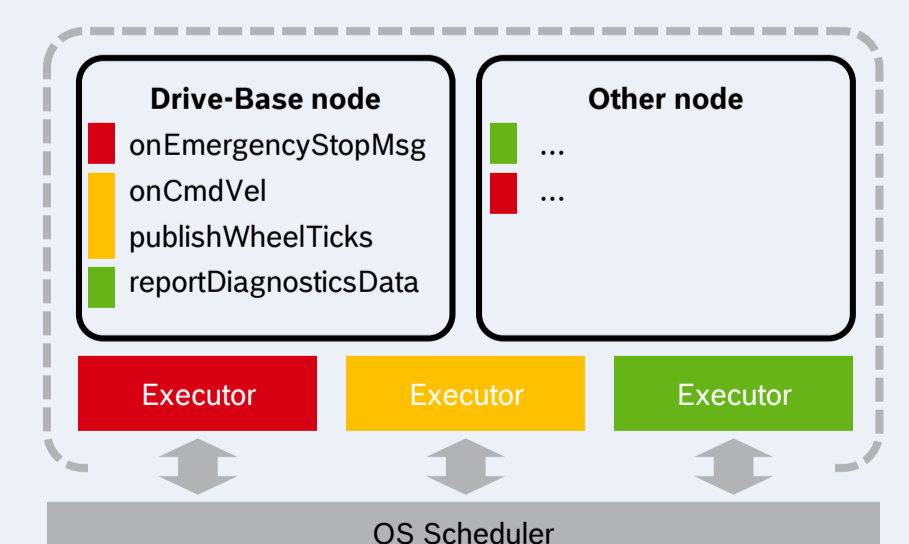
- ▶ Collect timing information with minimal overhead
- ▶ Instrumented ROS core packages using LTTng
- ▶ Hooks for application code



gitlab.com/micro-ROS/ros_tracing

ROS 2 Executor Concepts

- ▶ ROS 2 allows for own execution semantics
- ▶ Analyzed current default implementation
 - ... cf. today's talk on response-time analysis of ROS 2
- ▶ Developed fine-grained executor assignment API
- ▶ Design discussion in ROS 2 *Real-time Working Group* on real-time executors



Contributions welcome!

github.com/boschresearch/ros2_rclcpp/