

Current Status of ROS 2

Hands-on Feature Overview

Dr. Ralph Lange

Robert Bosch GmbH, Corporate Research

7 May 2019

ROS-I EU Spring '19 Workshop

Current Status of ROS 2 - Hands-on Feature Overview

Agenda

- ▶ Why, Roadmap and Governance
- ▶ Installation and Command Line Tools
- ▶ ROS 2 Architecture and Layer-by-Layer Walkthrough
- ▶ Launch
- ▶ System Modes
- ▶ Real-Time
- ▶ Bridge to ROS 1
- ▶ Security
- ▶ FMI Adapter
- ▶ Build Tooling
- ▶ Contributing to ROS 2

Why, Roadmap and Governance

Current Status of ROS 2 - Hands-on Feature Overview

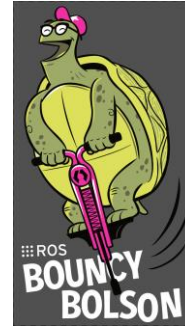
Why ROS ?

Use-case / basic requirements

- ▶ Teams of multiple robots
- ▶ Small embedded platforms
- ▶ Real-time systems
- ▶ Non-ideal networks
- ▶ Production environments
- ▶ Prescribed patterns for building and structuring systems



December '17



July '18



December '18

**Dashing
Diademata**

May '19

History

- ▶ Development started in 2014
- ▶ First stable release in December 2017
- ▶ First LTS release planned for May 2019



http://design.ros2.org/articles/why_ros2.html
<https://index.ros.org/doc/ros2/Roadmap/>

Current Status of ROS 2 - Hands-on Feature Overview

Key Features of ROS 2

- ▶ Data Distribution Service (DDS) as middleware
 - ▶ No central ROS master by DDS discovery
 - ▶ Built-in security: authentication, access control, encryption
- ▶ Run multiple nodes in one process
 - ▶ No differentiation between nodelets and nodes
 - ▶ Efficient zero-copy communication
- ▶ Node lifecycle and (more) deterministic launch
- ▶ ROS core functionality in C for easier support of different programming languages
- ▶ Real-time ready core algorithms
- ▶ New build systemament+colcon
- ▶ Windows-support
- ▶ Use of C++14 and Python 3 standards

```
4 <profiles>
5   <profile ns="/" node="talker">
6     <xi:include href="common/node.xml"
7       xpointer="xpointer(/profile/*)"/>
8     <topics publish="ALLOW" >
9       <topic>chatter</topic>
10    </topics>
11  </profile>
12  <profile ns="/" node="listener">
13    <xi:include href="common/node.xml"
14      xpointer="xpointer(/profile/*)"/>
15    <topics subscribe="ALLOW" >
16      <topic>chatter</topic>
17    </topics>
18  </profile>
19 </profiles>
```

Example from <https://github.com/ros2/sros2>



<http://design.ros2.org/>

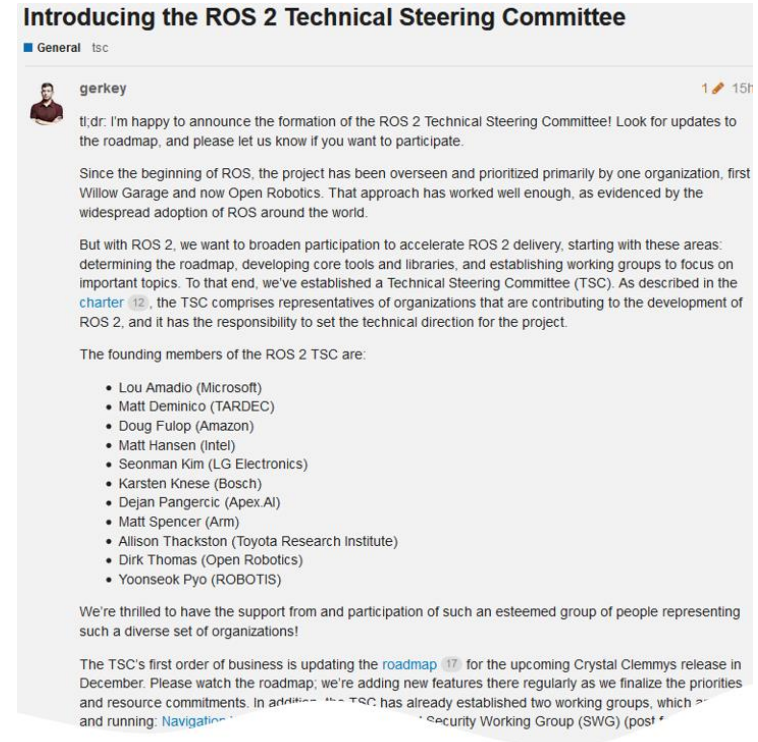
Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Governance

► Sep 2018: Formation of **ROS 2 Technical Steering Committee**

“broaden participation to accelerate ROS 2 delivery, starting with these areas: determining the **roadmap**, developing core tools and libraries, and establishing **working groups** to focus on important topics”

- Monthly meetings
 - Primarily status reports and roadmap development
 - Members provide feedback/priorities on OSRF tasks and contribute own work
- ### ► Current working groups
- Navigation, Security, Embedded Systems, Real-Time, Safety, Manipulation



Introducing the ROS 2 Technical Steering Committee

General tsc

gerkey 15h

tl;dr: I'm happy to announce the formation of the ROS 2 Technical Steering Committee! Look for updates to the roadmap, and please let us know if you want to participate.

Since the beginning of ROS, the project has been overseen and prioritized primarily by one organization, first Willow Garage and now Open Robotics. That approach has worked well enough, as evidenced by the widespread adoption of ROS around the world.

But with ROS 2, we want to broaden participation to accelerate ROS 2 delivery, starting with these areas: determining the roadmap, developing core tools and libraries, and establishing working groups to focus on important topics. To that end, we've established a Technical Steering Committee (TSC). As described in the [charter](#), the TSC comprises representatives of organizations that are contributing to the development of ROS 2, and it has the responsibility to set the technical direction for the project.

The founding members of the ROS 2 TSC are:

- Lou Amadio (Microsoft)
- Matt Deminico (TARDEC)
- Doug Fulop (Amazon)
- Matt Hansen (Intel)
- Seonman Kim (LG Electronics)
- Karsten Knese (Bosch)
- Dejan Pangercic (Apex.AI)
- Matt Spencer (Arm)
- Allison Thackston (Toyota Research Institute)
- Dirk Thomas (Open Robotics)
- Yoonseok Pyo (ROBOTIS)

We're thrilled to have the support from and participation of such an esteemed group of people representing such a diverse set of organizations!

The TSC's first order of business is updating the [roadmap](#) for the upcoming Crystal Clemmys release in December. Please watch the roadmap, we're adding new features there regularly as we finalize the priorities and resource commitments. In addition, the TSC has already established two working groups, which are currently running: [Navigation](#) and [Security Working Group \(SWG\)](#) (post #...)



<https://index.ros.org/doc/ros2/Governance/>

<https://discourse.ros.org/t/introducing-the-ros-2-technical-steering-committee/6132>

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Roadmap

- ▶ Long-term roadmap at <https://index.ros.org/doc/ros2/Roadmap/>
- ▶ Roadmap for next release Dashing Diademata:
 - ▶ Wiki: <https://index.ros.org/doc/ros2/Releases/Release-Dashing-Diademata/>
 - ▶ Meta-ticket: <https://github.com/ros2/ros2/issues/607>

Installation

Current Status of ROS 2 - Hands-on Feature Overview

Installation

- ▶ Binary packages for
 - ▶ Ubuntu 18.04 (Bionic)
 - ▶ OS X
 - ▶ Windows

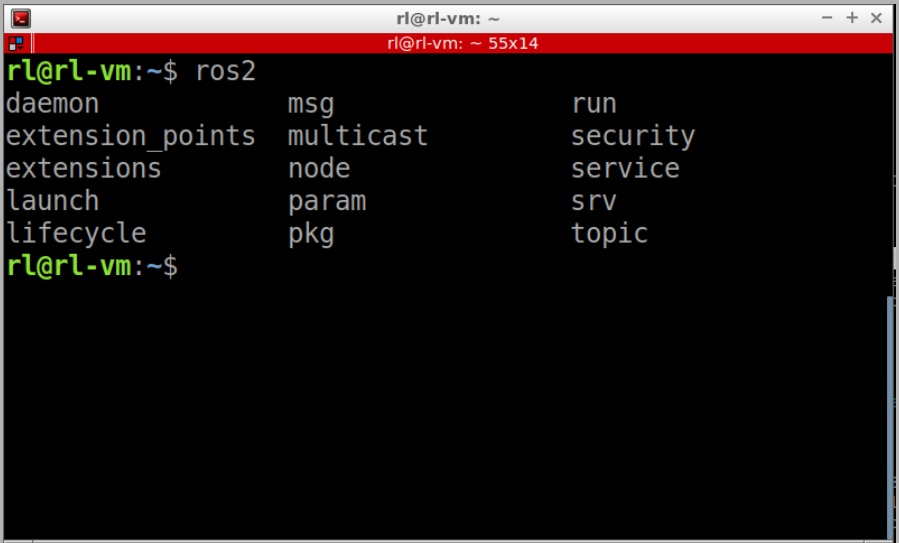
See <https://index.ros.org/doc/ros2/Installation/>

- ▶ ROS 2 Crystal can be also build directly on Ubuntu 16.04 Xenial – with exception of rqt packages
See <https://index.ros.org/doc/ros2/Installation/Linux-Development-Setup/>

Current Status of ROS 2 - Hands-on Feature Overview

Command Line Tools

- ▶ Don't forget to source!
 - ▶ `local_setup.bash` applies settings from current workspace only, i.e. without parent
- ▶ All run-time tools integrated in **ros2**
- ▶ <https://github.com/ros2/ros2cli>
 - ▶ Every verb is in a separate package
- ▶ Possible to extend, cf. for example <https://github.com/ros2/sros2/>
- ▶ Meta build tool: **colcon**
 - ▶ *Later more ...*



```
rl@rl-vm: ~  
rl@rl-vm: ~ 55x14  
rl@rl-vm:~$ ros2  
daemon          msg              run  
extension_points  multicast       security  
extensions       node            service  
launch          param           srv  
lifecycle       pkg             topic  
rl@rl-vm:~$
```

```
rl@rl-vm: ~  
rl@rl-vm: ~ 90x24  
[INFO] [talker]: Publishing: 'Hello World: 248'  
[INFO] [talker]: Publishing: 'Hello World: 249'  
[INFO] [talker]: Publishing: 'Hello World: 250'  
[INFO] [talker]: Publishing: 'Hello World: 251'  
[INFO] [talker]: Publishing: 'Hello World: 252'  
[INFO] [talker]: Publishing: 'Hello World: 253'  
[INFO] [talker]: Publishing: 'Hello World: 254'  
[INFO] [talker]: Publishing: 'Hello World: 255'  
[INFO] [talker]: Publishing: 'Hello World: 256'  
[INFO] [talker]: Publishing: 'Hello World: 257'  
[INFO] [talker]: Publishing: 'Hello World: 258'  
[INFO] [talker]: Publishing: 'Hello World: 259'  
[INFO] [talker]: Publishing: 'Hello World: 260'  
[INFO] [talker]: Publishing: 'Hello World: 261'  
[INFO] [talker]: Publishing: 'Hello World: 262'  
[INFO] [talker]: Publishing: 'Hello World: 263'  
[INFO] [talker]: Publishing: 'Hello World: 264'  
[INFO] [talker]: Publishing: 'Hello World: 265'  
[INFO] [talker]: Publishing: 'Hello World: 266'  
[INFO] [talker]: Publishing: 'Hello World: 267'  
[INFO] [talker]: Publishing: 'Hello World: 268'  
[INFO] [talker]: Publishing: 'Hello World: 269'  
[INFO] [talker]: Publishing: 'Hello World: 270'
```

**source /opt/ros/crystal/setup.bash
ros2 run demo_nodes_cpp talker**

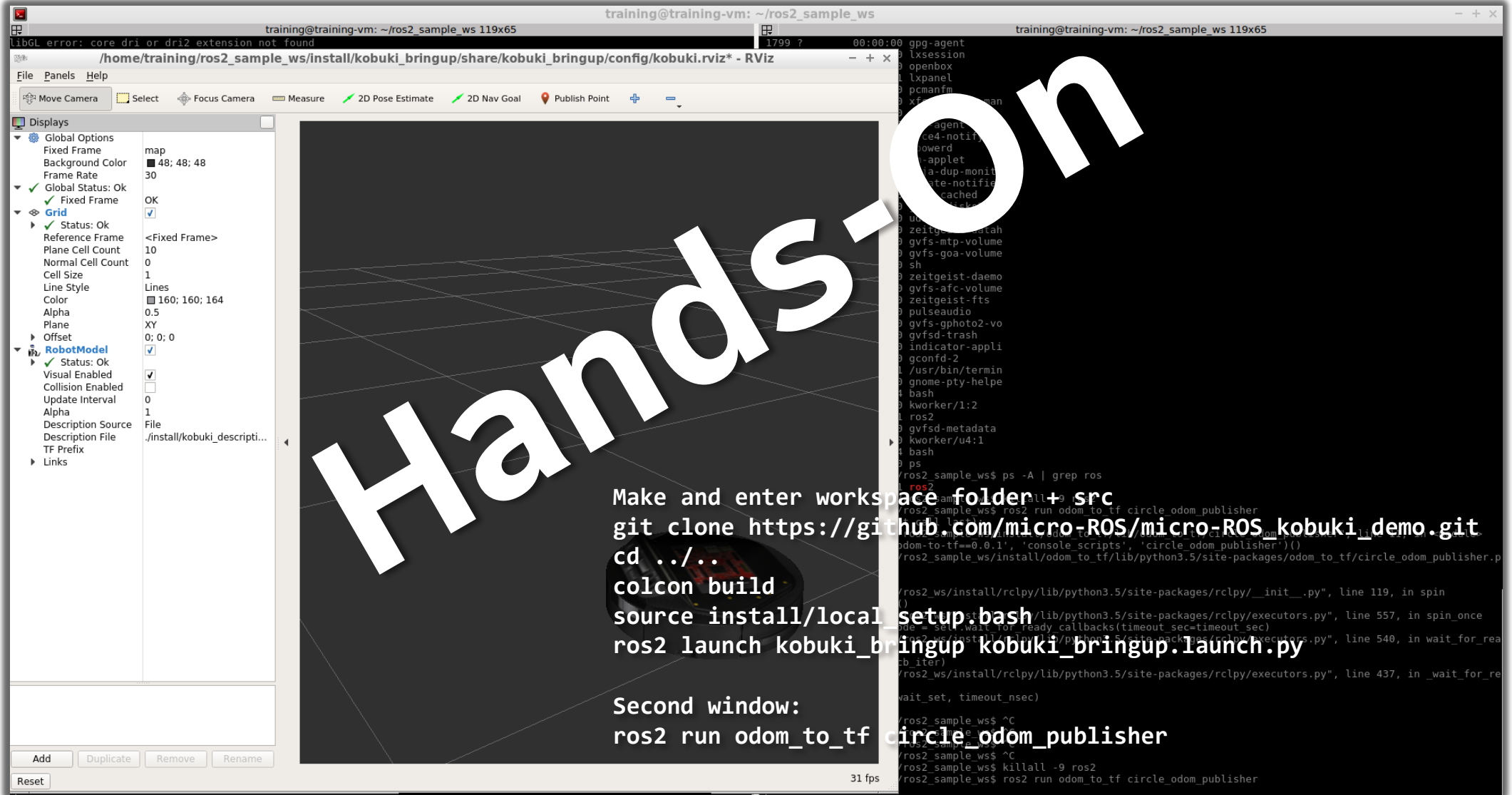
```
rl@rl-vm: ~  
rl@rl-vm: ~ 94x24  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 260]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 261]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 262]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 263]  
[INFO] [listener]: I heard: [Hello World: 264]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 265]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 266]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 267]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 268]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 269]  
[INFO] [listener]: I heard: [test]  
[INFO] [listener]: I heard: [Hello World: 270]
```

**source /opt/ros/crystal/setup.bash
ros2 run demo_nodes_py listener**

```
rl@rl-vm: ~  
rl@rl-vm: ~ 94x24  
publishing #96: std_msgs.msg.String(data='test')  
publishing #97: std_msgs.msg.String(data='test')  
publishing #98: std_msgs.msg.String(data='test')  
publishing #99: std_msgs.msg.String(data='test')  
publishing #100: std_msgs.msg.String(data='test')  
publishing #101: std_msgs.msg.String(data='test')  
publishing #102: std_msgs.msg.String(data='test')  
publishing #103: std_msgs.msg.String(data='test')  
publishing #104: std_msgs.msg.String(data='test')  
publishing #105: std_msgs.msg.String(data='test')  
publishing #106: std_msgs.msg.String(data='test')
```

**source /opt/ros/crystal/setup.bash
ros2 topic list
ros2 topic hz /chatter
ros2 topic pub /chatter std_msgs/String
" data: 'test' "**

Hands-On



Architecture and Layer-by-Layer Walkthrough

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Architecture

rcl* – language-specific ROS client libraries

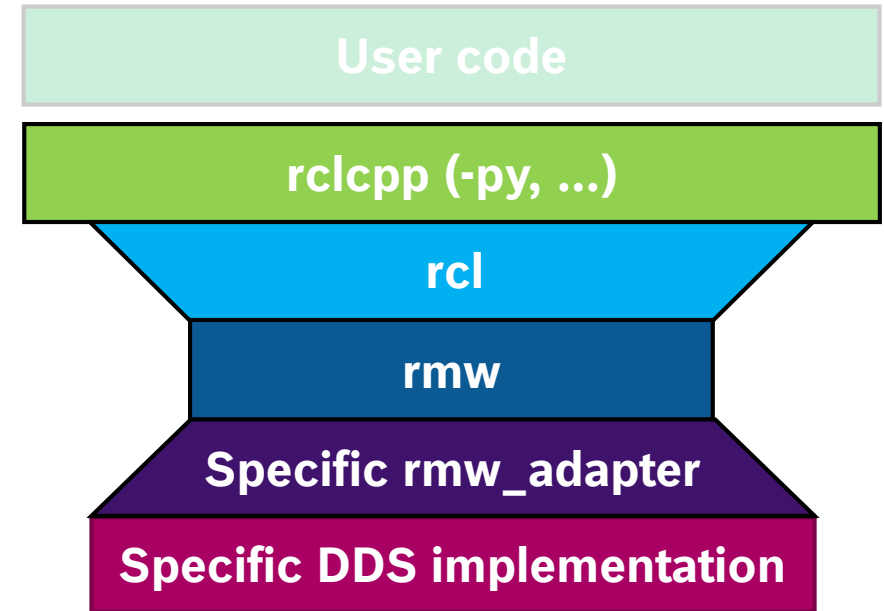
rcl – C library

- ▶ Ensures same core algorithms in all language-specific client libraries

rmw – ROS middleware interface

- ▶ Hide specifics of DDS implementations
- ▶ Streamline QoS configuration

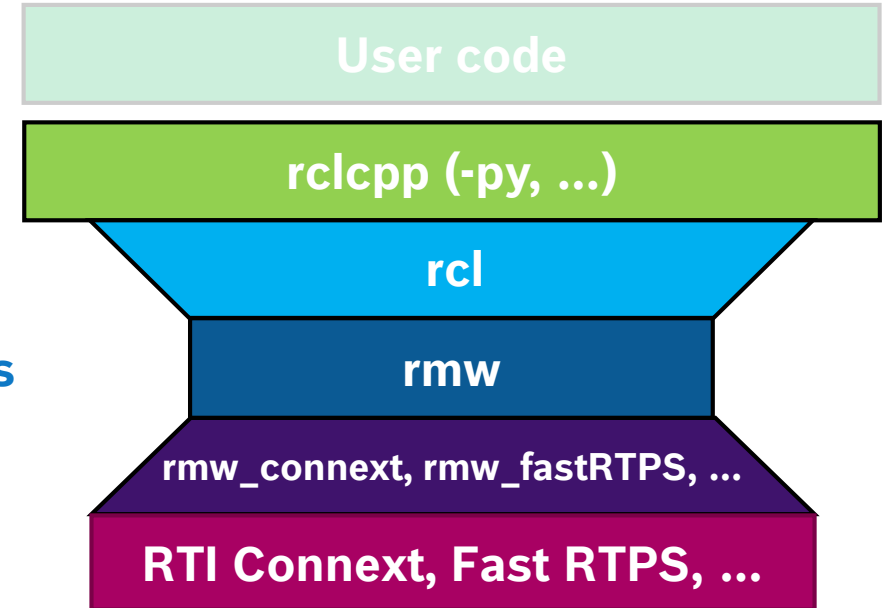
rmw_* – DDS adapters



Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: DDS and rmw

- ▶ You should be the experts after yesterday's talks (-:
- ▶ FastRTPS is the default distributed with ROS 2
- ▶ Instructions for OpenSplice and RTI Connex: <https://index.ros.org/doc/ros2/Installation/Linux-Install-Debian/#install-additional-rmw-implementations>
 - ▶ Adapters for those both are provided as binary packages
- ▶ rmw defines interface to be implemented by adapters
 - ▶ E.g., search `rmw_get_publisher_names_and_types_by_node` in <https://github.com/ros2/rmw/>, https://github.com/ros2/rmw_fastrtps/ and https://github.com/ros2/rmw_connext/



Current Status of ROS 2 - Hands-on Feature Overview

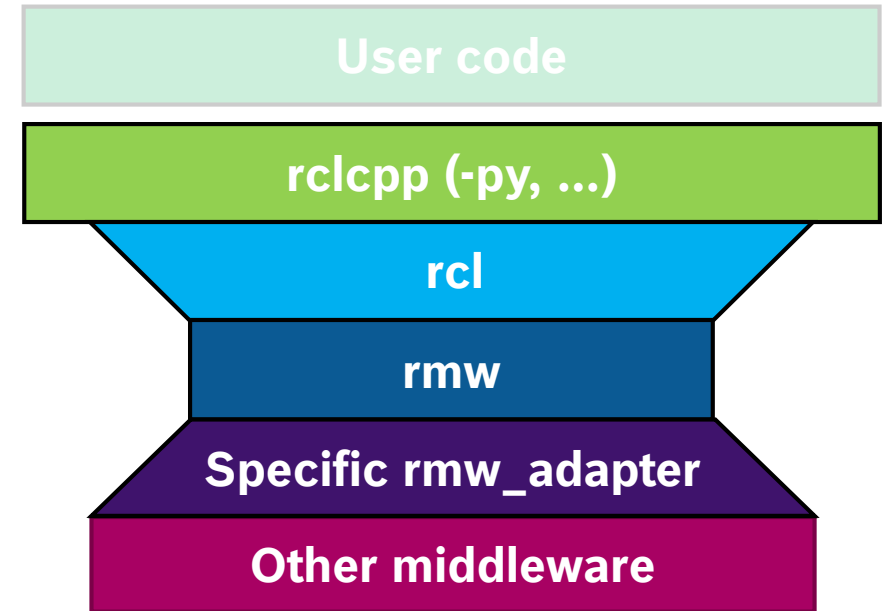
ROS 2 Layer by Layer: Other Middlewares

rmw defines few basic concepts only:

- ▶ some endpoint naming
- ▶ publish-subscribe
- ▶ request-response

Several non-DDS implementations have been developed, for example:

- ▶ OPC UA – an M2M communication protocol
<https://www.elektrotechnik.vogel.de/mit-opc-ua-gelingt-die-nahtlose-integration-mobiler-robotersysteme-a-537411/> (in Germany only)
- ▶ Intel's Distributed Publish & Subscribe for IoT
https://github.com/ros2/rmw_dps

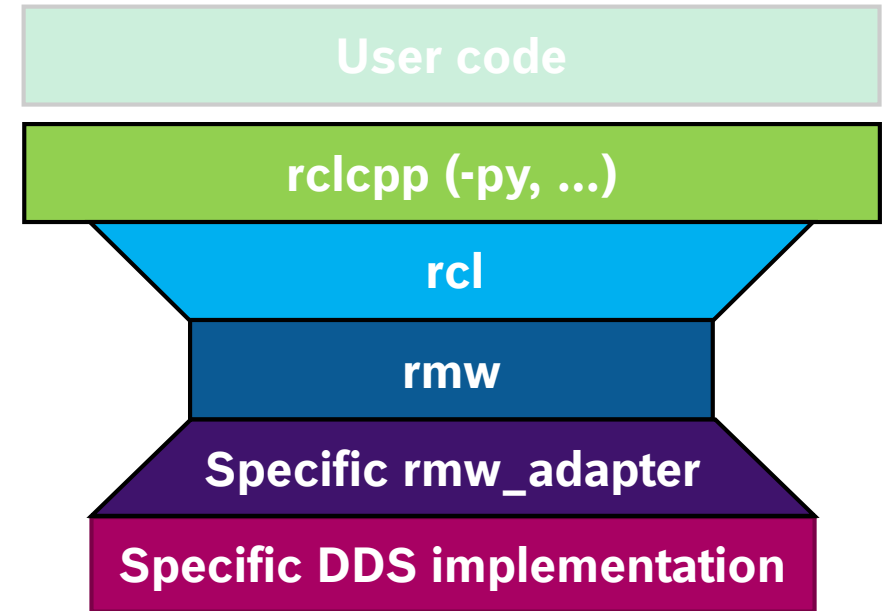


Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: rcl – ROS Client Support Library

rcl implements major ROS concepts in C

- ▶ node, naming, publisher, subscription, service, action, lifecycle, time, logging, ...
- ▶ Very well documented API: <https://github.com/ros2/rcl/>
- ▶ Allows use of custom allocator (e.g., TLSF)
<https://github.com/ros2/rcutils> -> include/rcutils/allocator.h
- ▶ Lacks of execution management – no threads on rcl layer



<https://index.ros.org/doc/ros2/Tutorials/Allocator-Template-Tutorial/>
<http://www.gii.upv.es/tlsf/>

Current Status of ROS 2

Simple Subscription with rcl

```
int main(int argc, char* argv[])
{
    rcl_init_options_t initOptions = rcl_get_zero_initialized_init_options();
    rcl_init_options_init(&initOptions, rcutils_get_default_allocator());
    rcl_context_t context = rcl_get_zero_initialized_context();
    rcl_init(argc, argv, &initOptions, &context);
    rcl_init_options_fini(&initOptions);

    rcl_node_t node = rcl_get_zero_initialized_node();
    rcl_node_options_t nodeOptions = rcl_node_get_default_options();
    rcl_node_init(&node, "rcl_int32_subscriber", "", &context, &nodeOptions);

    const rosidl_message_type_support_t * typeSupport = ROSIDL_GET_MSG_TYPE_SUPPORT(std_msgs, msg, Int32);
    rcl_subscription_t subscription = rcl_get_zero_initialized_subscription();
    rcl_subscription_options_t subscriptionOptions = rcl_subscription_get_default_options();
    rcl_subscription_init(&subscription, &node, typeSupport, "std_msgs_msg_Int32", &subscriptionOptions);

    rcl_wait_set_t waitSet = rcl_get_zero_initialized_wait_set();
    rcl_wait_set_init(&waitSet, 1, 0, 0, 0, 0, rcl_get_default_allocator());
    size_t index;
    rcl_wait_set_add_subscription(&waitSet, &subscription, &index);
    rcl_wait(&waitSet, RCL_MS_TO_NS(1000));

    // Check waitSet for available messages

    std_msgs__msg__Int32 msg;
    rmw_message_info_t messageInfo;
    rcl_take(&subscription, &msg, &messageInfo);
    printf("Message data is %ld\n", msg.data);

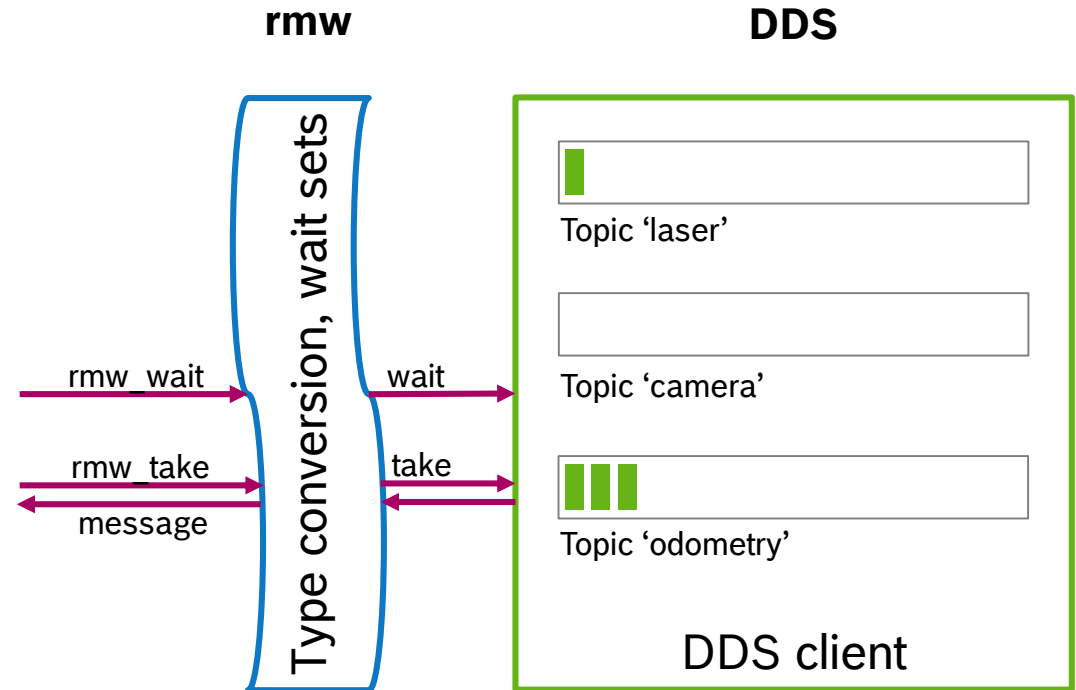
    rcl_subscription_fini(&subscription, &node);
    rcl_node_fini(&node);
    return 0;
}
```

https://github.com/micro-ROS/micro-ROS_experiments
→ branch `measure_allocations`

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: Waitsets

- ▶ SELECT-like mechanism to query for messages on subscribed topics
- ▶ Does not provide information on the message count, i.e. current queue size

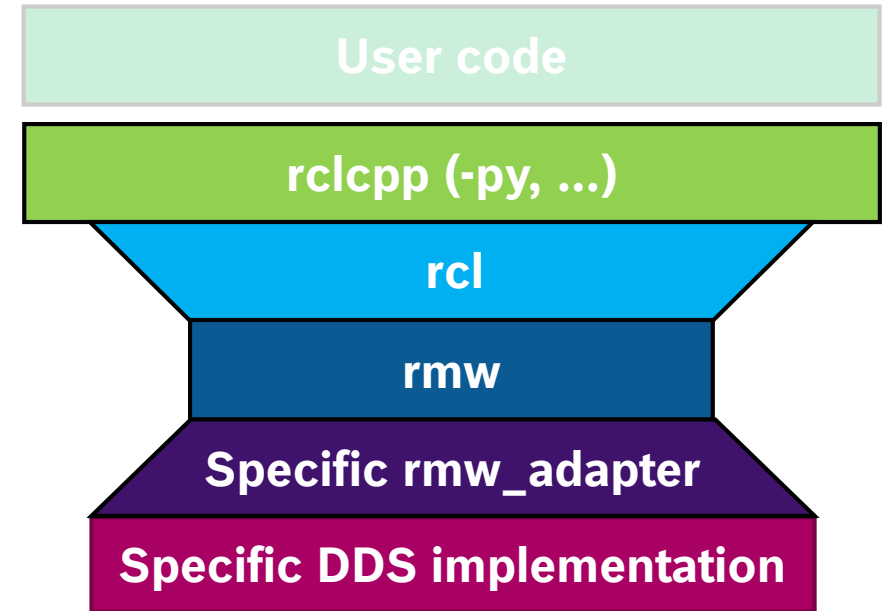


Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: rclcpp Architecture

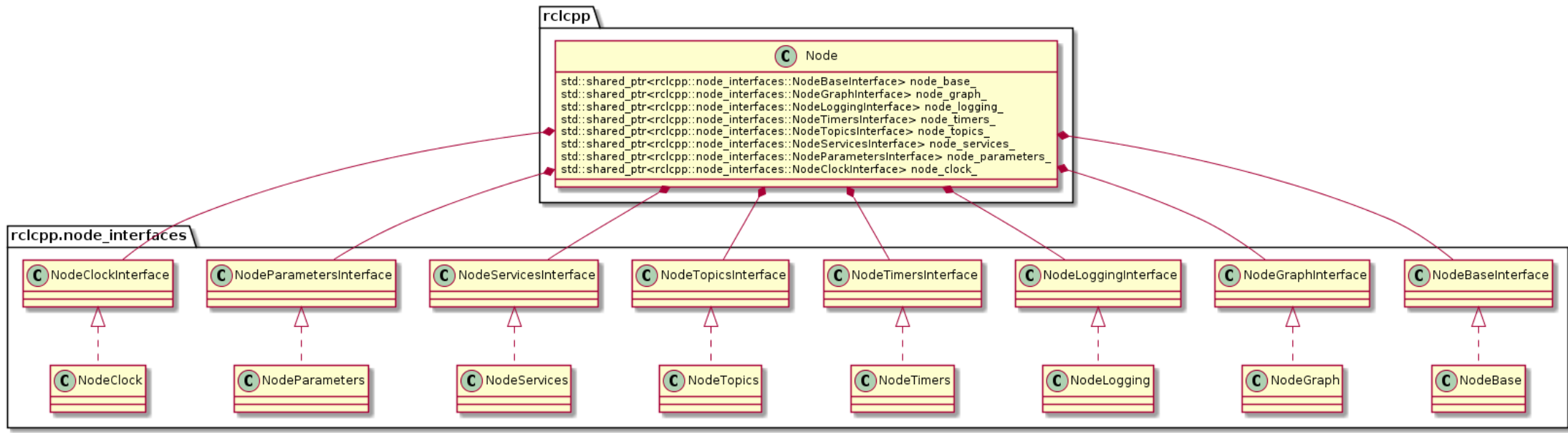
rclcpp wraps rcl into C++ data types and brings

- ▶ Execution management
 - ▶ Intraprocess communication
 - ▶ Parameters
 - ▶ Callback_groups
 - ▶ Listeners for graph, parameters
-
- ▶ Allows dynamic creation and deletion of almost everything
 - ▶ Elaborate architecture



Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: rclcpp Architecture



► <https://github.com/ros2/rclcpp/> → `include/rclcpp/node_interfaces`

Current Status of ROS 2

Simple Subscription with rclcpp

```
class MyNode : public rclcpp::Node {
public:
    MyNode() : Node("my_node") {
        auto callback = [this](const std_msgs::msg::String::SharedPtr msg) -> void {
            RCLCPP_INFO(this->get_logger(), "I heard: [%s]", msg->data.c_str());
        };
        sub_ = create_subscription<std_msgs::msg::String>("/chatter", callback);
    }

private:
    rclcpp::Subscription<std_msgs::msg::String>::SharedPtr sub_;
};

int main(int argc, char * argv[]) {
    rclcpp::init(argc, argv);

    auto node = std::make_shared<MyNode>(topic);
    rclcpp::spin(node);

    rclcpp::shutdown();
    return 0;
}
```

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: rclcpp Parameters

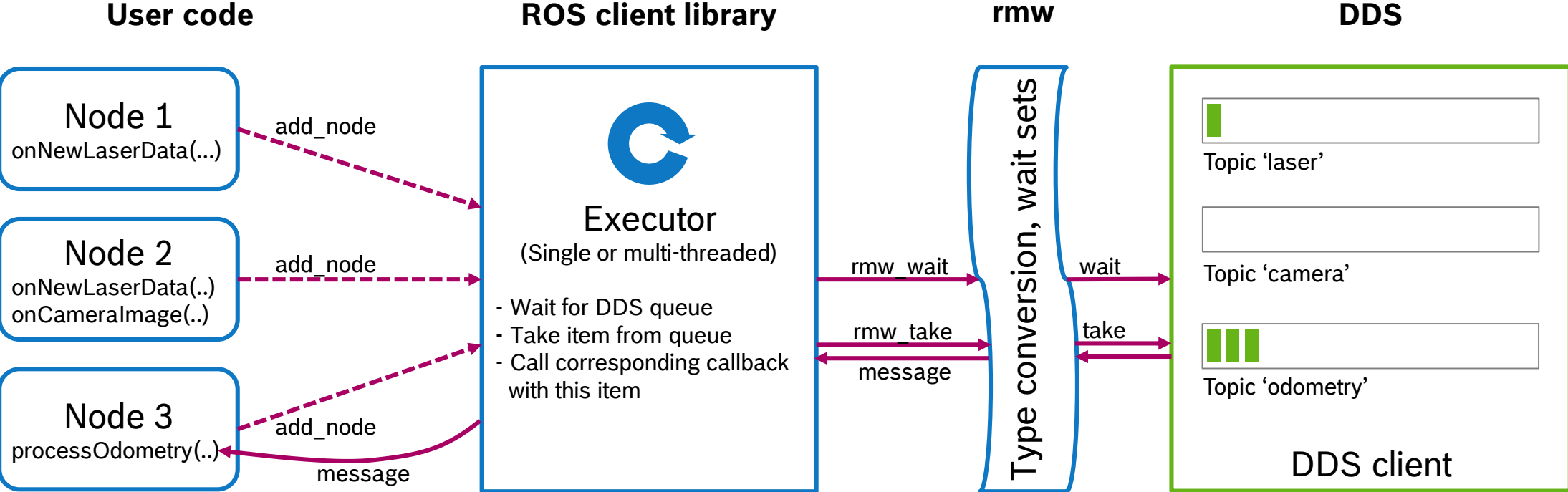
- ▶ Managed with node – no parameter server
- ▶ Handy tooling with ros2 param
- ▶ Typed (and to be declared from Dashing on)
 - ▶ ... but nice template in rclcpp/node.hpp
- ▶ Old semantics – no declaration – still available via NodeOptions

```
template<typename ParameterT>
auto
declare_parameter(
    const std::string & name,
    const ParameterT & default_value,
    const rcl_interfaces::msg::ParameterDescriptor & parameter_descriptor =
        rcl_interfaces::msg::ParameterDescriptor());

rclcpp::Parameter parameter;
if (nodeInterface->get_parameter(name, parameter)) {
    value = parameter.as_double();
}
```

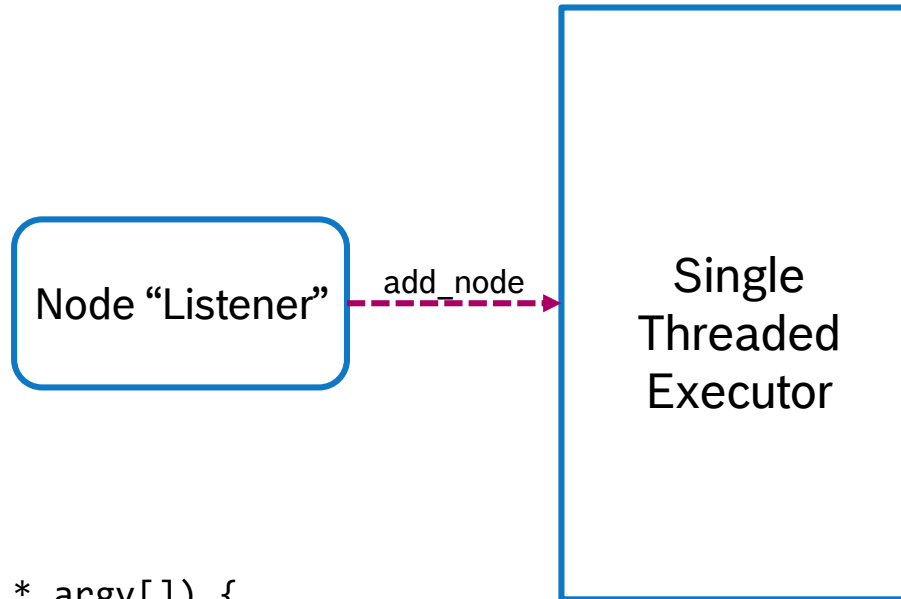
Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: rclcpp – Node Composition and Executor



Current Status of ROS 2 - Hands-on Feature Overview

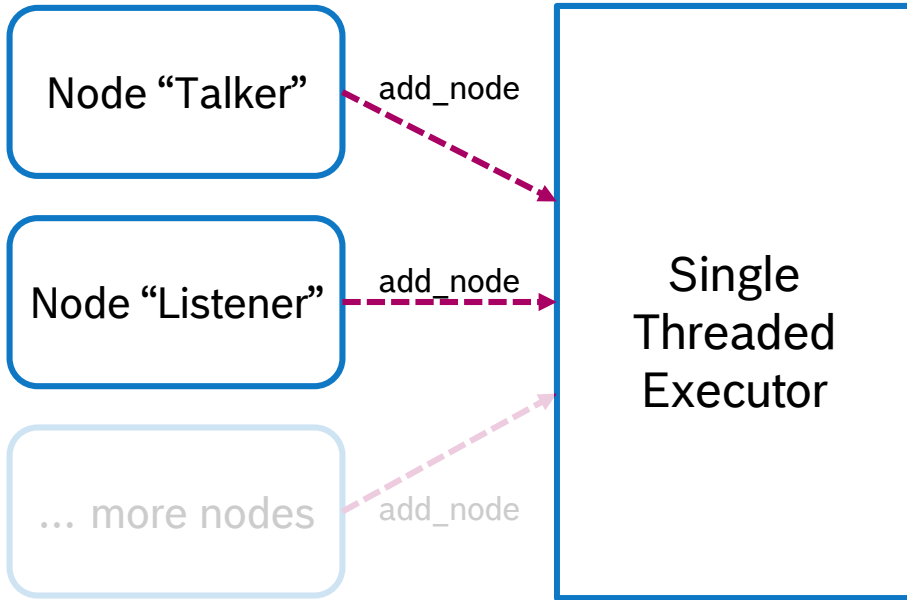
ROS 2 Layer by Layer: rclcpp – Node Composition and Executor



```
int main(int argc, char * argv[]) {  
    rclcpp::init(argc, argv);  
    auto node = rclcpp::Node::make_shared("listener");  
    auto sub = node->create_subscription<std_msgs::msg::String>("/chatter", callback);  
    rclcpp::spin(node);  
    rclcpp::shutdown();  
}
```

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: rclcpp – Node Composition and Executor



```
int main(int argc, char * argv[]) {
    rclcpp::init(argc, argv);

    rclcpp::executors::SingleThreadedExecutor executor;

    auto talker_node = std::make_shared<Talker>();
    executor.add_node(talker_node);

    auto listener_node = std::make_shared<Listener>();
    executor.add_node(listener_node);

    executor.spin();

    rclcpp::shutdown();
}
```

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: Executor Semantics

- ▶ Analyzed and described current semantics of Executor implementation with models from real-time community
 - ▶ *Daniel Casini, Tobias Blaß, Ingo Lütkebohle and Björn Brandenburg: “Response-Time Analysis of ROS 2 Processing Chains under Reservation-Based Scheduling”, ECRTS 2019, to appear.*
 - ▶ https://github.com/boschresearch/ros2_response_time_analysis
- ▶ Complicated semantics consisting of
 - Priority scheduling on first level and
 - Round-robin for subscriptions on second level
- ▶ See <https://github.com/ros2/rclcpp/blob/master/rclcpp/src/rclcpp/executor.cpp#L540>
- ▶ On-going discussion in several GitHub issues and real-time working group

Current Status of ROS 2 - Hands-on Feature Overview

ROS 2 Layer by Layer: Managed Nodes (aka Lifecycle)

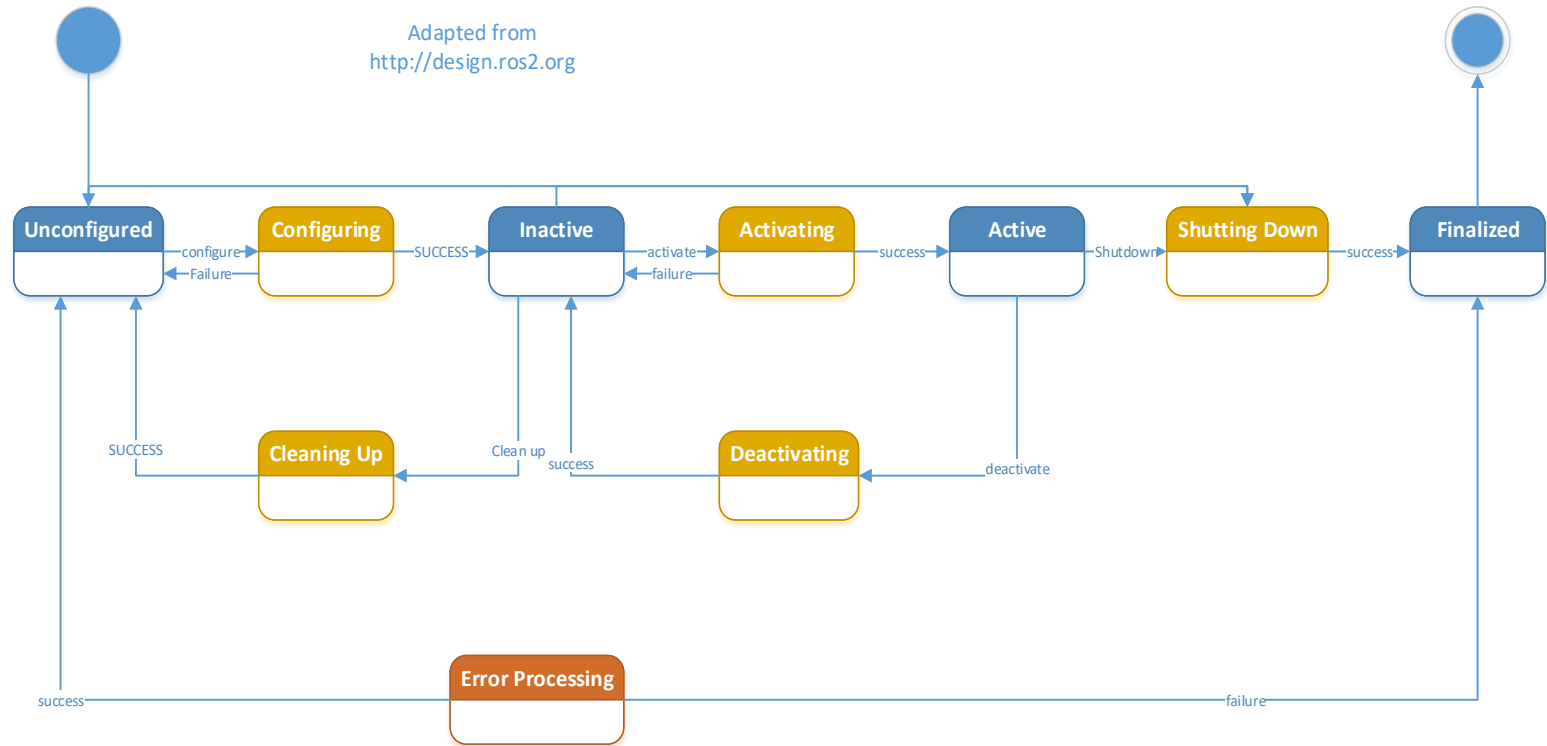
Lifecycle design
by Biggs & Foote, 2015

- ▶ Inspired by OMG RTC Managed Lifecycle

Concepts:

- ▶ Primary states
- ▶ Transition states
- ▶ Transitions

Tool: `ros2 lifecycle`



https://design.ros2.org/articles/node_lifecycle.html

https://github.com/bosch-robotics-cr/ros1_lifecycle

Current Status of ROS 2

ROS 2 Layer by Layer: rclpy

- ▶ <https://github.com/ros2/rclpy/>
- ▶ Well-developed and maintained
... and used for most command line tools

```
class OdomToTf(Node):

    def __init__(self):
        super().__init__('odom_to_tf')

        self.last_position = Vector3()
        self.last_orientation = Quaternion()
        self.last_orientation.x = 0.0
        self.last_orientation.y = 0.0
        self.last_orientation.z = 0.0
        self.last_orientation.w = 1.0

        self.sub_robot_pose = self.create_subscription(Vector3, "/robot_pose", self.
        self.pub_tf = self.create_publisher(TFMessage, "/tf")

        self.tf_timer = self.create_timer(0.05, self.tf_timer_callback)

    def robot_pose_callback(self, msg):
        self.last_position.x = msg.x;
        self.last_position.y = msg.y;
        self.last_orientation.z = math.sin(msg.z / 2.0);
        self.last_orientation.w = math.cos(msg.z / 2.0);
```

Example from https://github.com/micro-ROS/micro-ROS_kobuki_demo

Current Status of ROS 2

ROS 2 Layer by Layer: rclpy

- ▶ <https://github.com/ros2/rclpy/>
- ▶ Well-developed and maintained
... and used for most command line tools
- ▶ No need of CMakeLists.txt with colcon
but real Python package

```
from setuptools import find_packages
from setuptools import setup

package_name = 'odom_to_tf'

setup(
    name=package_name,
    version='0.0.1',
    packages=find_packages(exclude=['test']),
    data_files=[
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    author='Ralph Lange',
    author_email='ralph.lange@de.bosch.com',
    maintainer='Ralph Lange',
    maintainer_email='ralph.lange@de.bosch.com',
    keywords=['odom', 'tf'],
    classifiers=[
        'Intended Audience :: Developers',
        'License :: OSI Approved :: Apache Software License',
        'Programming Language :: Python',
        'Topic :: Software Development',
    ],
    license='Apache 2.0',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'odom_to_tf = odom_to_tf.odom_to_tf:main',
            'circle_odom_publisher = odom_to_tf.circle_odom_publisher:main',
        ],
    },
)
```

Example from https://github.com/micro-ROS/micro-ROS_kobuki_demo

Launch

Current Status of ROS 2 - Hands-on Feature Overview

Launch – Overview

- ▶ Launch files are Python scripts, i.e. very little prescribed structure
- ▶ **ros2 launch**
- ▶ Little documentation available
- ▶ On-going discussion and developments ... also regarding more prescribed mechanisms and structures
- ▶ Major know concepts from ROS 1 implemented
 - ▶ Nested launch files
 - ▶ Arguments
 - ▶ Parameter files

```
def generate_launch_description():  
    fmu_path = (ament_index_python.packages.get_package_share_directory('fmi_adapter_examples') +  
               '/share/DampedPendulum.fmu')  
  
    fmi_adapter_description = launch.actions.IncludeLaunchDescription(  
        launch.launch_description_sources.PythonLaunchDescriptionSource(  
            ament_index_python.packages.get_package_share_directory(  
                'fmi_adapter') + '/launch/fmi_adapter_node.launch.py'),  
        launch_arguments={'fmu_path': fmu_path}.items())  
  
    description = launch.LaunchDescription()  
    description.add_action(fmi_adapter_description)  
  
    return description
```

Example from https://github.com/boschresearch/fmi_adapter_ros2

Current Status of ROS 2 - Hands-on Feature Overview

Launch – New Concepts

Support for Managed Nodes (Lifecycle)

- ▶ Launch allows to register event handlers – e.g., on lifecycle state changes
- ▶ Nodes can be started in synchronized stages or as a pipeline or ...

```
activate_delay_node = launch.actions.EmitEvent(event=launch_ros.events.lifecycle.ChangeState(
    lifecycle_node_matcher=launch.events.process.matches_action(delay_node),
    transition_id=lifecycle_msgs.msg.Transition.TRANSITION_ACTIVATE))

pendulum_fmu_path = (
    ament_index_python.packages.get_package_share_directory('fmi_adapter_examples') +
    '/share/DampedPendulum.fmu')

pendulum_node = launch_ros.actions.LifecycleNode(
    package='fmi_adapter',
    node_executable='fmi_adapter_node',
    node_name='damped_pendulum',
    node_namespace='example',
    parameters=[{
        'fmu_path': pendulum_fmu_path,
        'l': 25.0, # Set pendulum length to 25m.
        'd': 0.01 # Reduce damping ratio (default is 0.1).
    }],
    output='screen')

configure_pendulum_node = launch.actions.EmitEvent(
    event=launch_ros.events.lifecycle.ChangeState(
        lifecycle_node_matcher=launch.events.process.matches_action(pendulum_node),
        transition_id=lifecycle_msgs.msg.Transition.TRANSITION_CONFIGURE))
```

Example from https://github.com/boschresearch/fmi_adapter_ros2

Current Status of ROS 2 - Hands-on Feature Overview

Launch – New Concepts

Support for Managed Nodes (Lifecycle)

- ▶ Launch allows to register event handlers – e.g., on lifecycle state changes
- ▶ Nodes can be started in synchronized stages or as a pipeline or ...

Node composition from shared libraries

- ▶ <https://index.ros.org/doc/ros2/Tutorials/Composition/>
- ▶ Example: https://github.com/boschresearch/fmi_adapter_ros2
Branch Dashing → examples → launch

```
node = launch_ros.actions.ComposableNodeContainer(  
    node_name='fmi_adapter_nodes',  
    node_namespace='',  
    package='rclcpp_components',  
    node_executable='component_container',  
    composable_node_descriptions=[  
        launch_ros.descriptions.ComposableNode(  
            package='fmi_adapter',  
            node_plugin='fmi_adapter::FMIAdapterNode',  
            node_namespace='/damped_pendulum',  
            node_name='damped_pendulum',  
            parameters=[  
                'fmu_path': pendulum_fmu_path,  
                'l': 25.0, # Set pendulum length to 25m.  
                'd': 0.01 # Reduce damping ratio (default is 0.1).  
            ]),  
        launch_ros.descriptions.ComposableNode(  
            package='fmi_adapter',  
            node_plugin='fmi_adapter::FMIAdapterNode',  
            node_namespace='/transport_delay',  
            node_name='transport_delay',  
            parameters=[  
                'fmu_path': delay_fmu_path,  
                'd': 2.33 # Set transport delay to 2.33s.  
            ]),  
    ],
```

System Modes

On-going activity, in the EU project
OFERA for micro-ROS

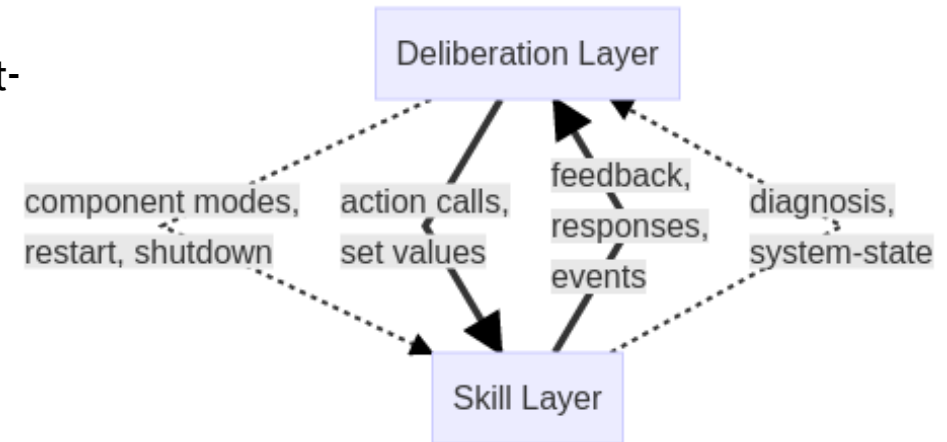


THE OFERA PROJECT IS FUNDED BY THE
EUROPEAN UNION'S HORIZON 2020 RESEARCH
AND INNOVATION PROGRAMME UNDER GRANT
AGREEMENT NO 780785

Current Status of ROS 2 - Hands-on Feature Overview

Extended Lifecycle and System Modes

- ▶ We observed three different but closely interwoven aspects to be handled on the deliberation layer:
 - 1. Task Handling:** Orchestration of the actual task, the straightforward, error-free flow
 - 2. Contingency Handling:** Handling of task-specific contingencies, e.g., expectable retries and failure attempts, obstacles, low battery.
 - 3. System Error Handling:** Handling of exceptions, e.g., sensor/actuator failures.
- ▶ The mechanisms being used to orchestrate the skills are service and action calls, re-parameterizations, set values, activating/deactivating of components, etc.
- ▶ Function-oriented calls vs. system-oriented calls **intermixed and implicit in code**



Current Status of ROS 2 - Hands-on Feature Overview

Extended Lifecycle and System Modes

► Assumption: System is built from nodes with lifecycle

► Introducing:

1. Modes

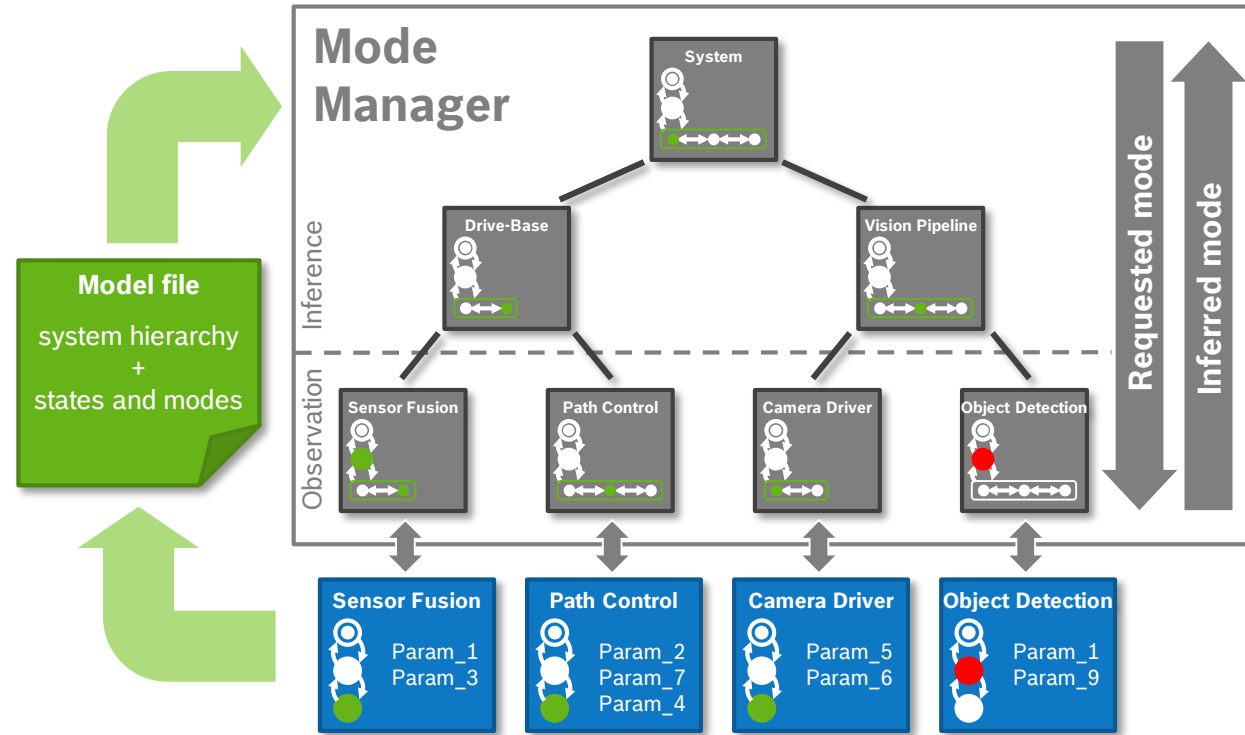
- Specialize *ACTIVE* state
- Set of values or ranges for node parameters

2. Notion of (sub)system

- Collection of nodes and (sub)systems (*system of system*)

3. Modes for (sub)systems

- (Sub)system get same lifecycle as nodes
- Modes of (sub)systems specify modes of their parts



https://micro-ros.github.io/system_modes/
https://github.com/microROS/system_modes

```

ri@ri-vm: ~/system_modes_ws 116x12
^C [INFO] [rclcpp]: signal_handler(signal_value=2)
ri@ri-vm: ~/system_modes_ws ^C
ri@ri-vm: ~/system_modes_ws ^C
ri@ri-vm: ~/system_modes_ws ^C
ri@ri-vm: ~/system_modes_ws ./install/system_modes_examples/bin/drive_base
[INFO] [drive_base]: Constructed lifecycle node 'drive_base'
^C [INFO] [rclcpp]: signal_handler(signal_value=2)
ri@ri-vm: ~/system_modes_ws ^C
ri@ri-vm: ~/system_modes_ws drive_base
[INFO] [drive_base]: Constructed lifecycle node 'drive_base'

ri@ri-vm: ~/system_modes_ws 116x5
ri@ri-vm: ~/system_modes_ws man
man      mandb      manipulator_manpath
ri@ri-vm: ~/system_modes_ws manipulator
[INFO] [manipulator]: Constructed lifecycle node 'manipulator'

ri@ri-vm: ~/system_modes_ws 116x13
System Modes Monitor - Mon May 6 09:34:37 2019
Model: system_modes_examples/share/example_modes.yaml

+-----+-----+-----+-----+-----+
| part | mode | target | actual |
+-----+-----+-----+-----+
| systems | actuation | | |
+-----+-----+-----+-----+
| nodes | | | |
+-----+-----+-----+-----+
| manipulator | | | |
+-----+-----+-----+-----+

ri@ri-vm: ~/system_modes_ws/install 116x10
[INFO] [mode_manager]: - actuation/get_available_modes
[INFO] [mode_manager]: - node 'drive_base'
[INFO] [mode_manager]: - drive_base/change_mode
[INFO] [mode_manager]: - drive_base/get_mode
[INFO] [mode_manager]: - drive_base/get_available_modes
[INFO] [mode_manager]: - node 'manipulator'
[INFO] [mode_manager]: - manipulator/change_mode
[INFO] [mode_manager]: - manipulator/get_mode
[INFO] [mode_manager]: - manipulator/get_available_modes

```

Hands-On

```

# Clone system_modes
colcon build & source install/local_setup.bash
ros2 run system_modes_examples drive_base

```

```

# New bash
ros2 run system_modes_examples manipulator

```

```

# New bash
ros2 run system_modes mode-manager --modelfile
src/system_modes/system_modes_examples/example_modes.yaml

```

```

# New bash
ros2 run system_modes mode-monitor --modelfile
src/system_modes/system_modes_examples/example_modes.yaml

```

```

# New bash
ros2 service call /actuation/change_state lifecycle_msgs/ChangeState
"{transition: {id: 1, label: configure}}"
ros2 service call /actuation/change_state lifecycle_msgs/ChangeState
"{transition: {id: 3, label: activate}}"
ros2 service call /actuation/change_mode system_modes/ChangeMode
"{node_name: 'actuation', mode_name: 'PERFORMANCE'}"
ros2 service call /drive_base/change_mode system_modes/ChangeMode
"{node_name: 'drive_base', mode_name: 'SLOW'}"
ros2 service call /manipulator/change_mode system_modes/ChangeMode
"{node_name: 'manipulator', mode_name: 'WEAK'}"

```



https://github.com/micro-ROS/system_modes/tree/master/system_modes_examples

Real-Time

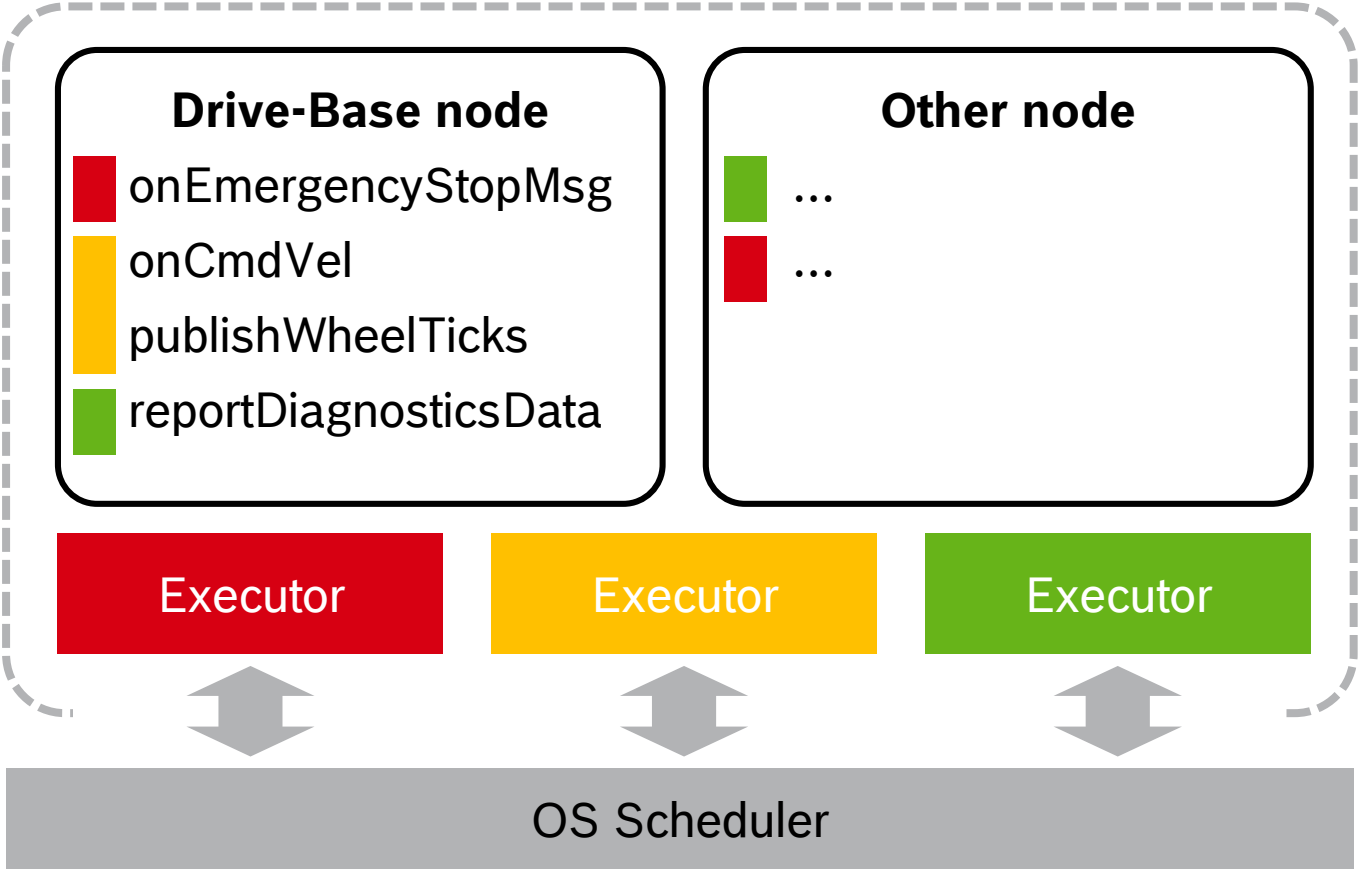
On-going activity, partially in the
EU project OFERA for micro-ROS



THE OFERA PROJECT IS FUNDED BY THE
EUROPEAN UNION'S HORIZON 2020 RESEARCH
AND INNOVATION PROGRAMME UNDER GRANT
AGREEMENT NO 780785

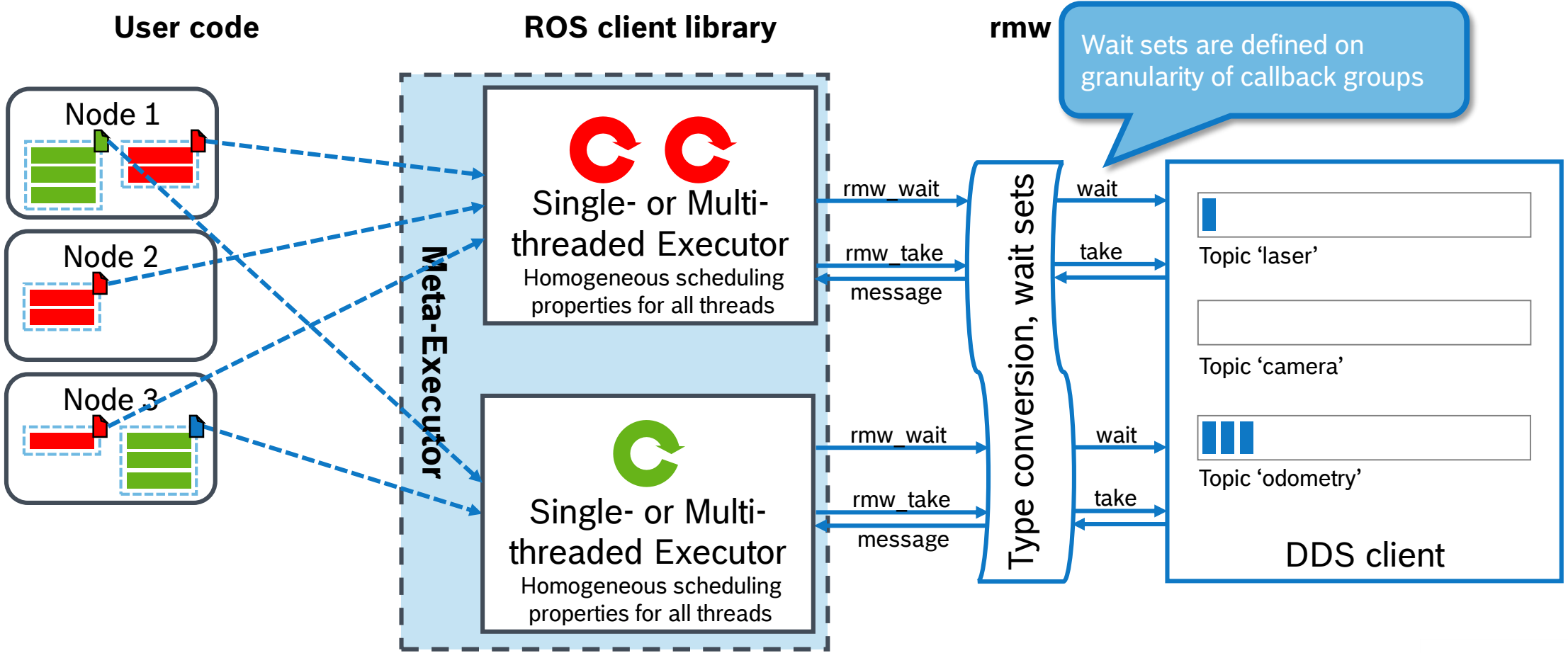
Current Status of ROS 2 - Hands-on Feature Overview

Callback-group-level Executor



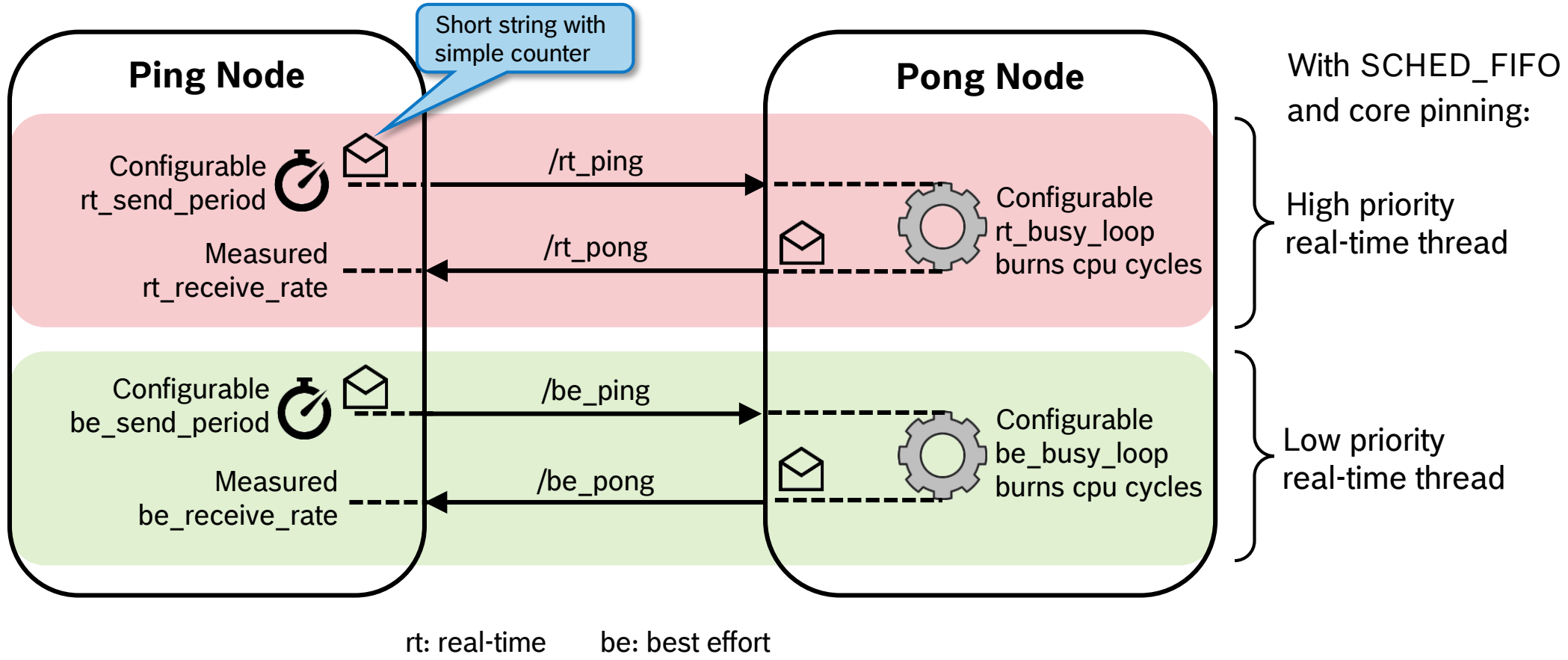
Current Status of ROS 2 - Hands-on Feature Overview

Callback-group-level Executor



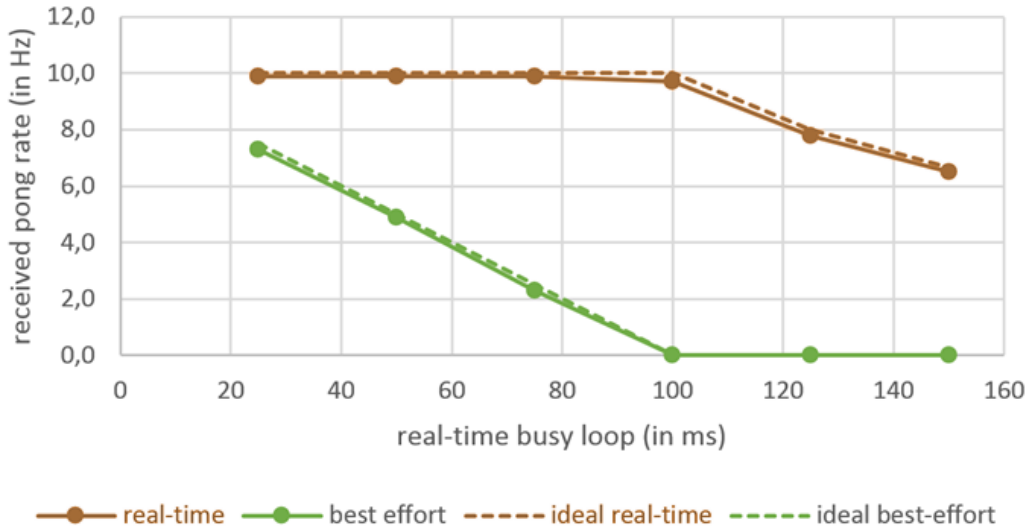
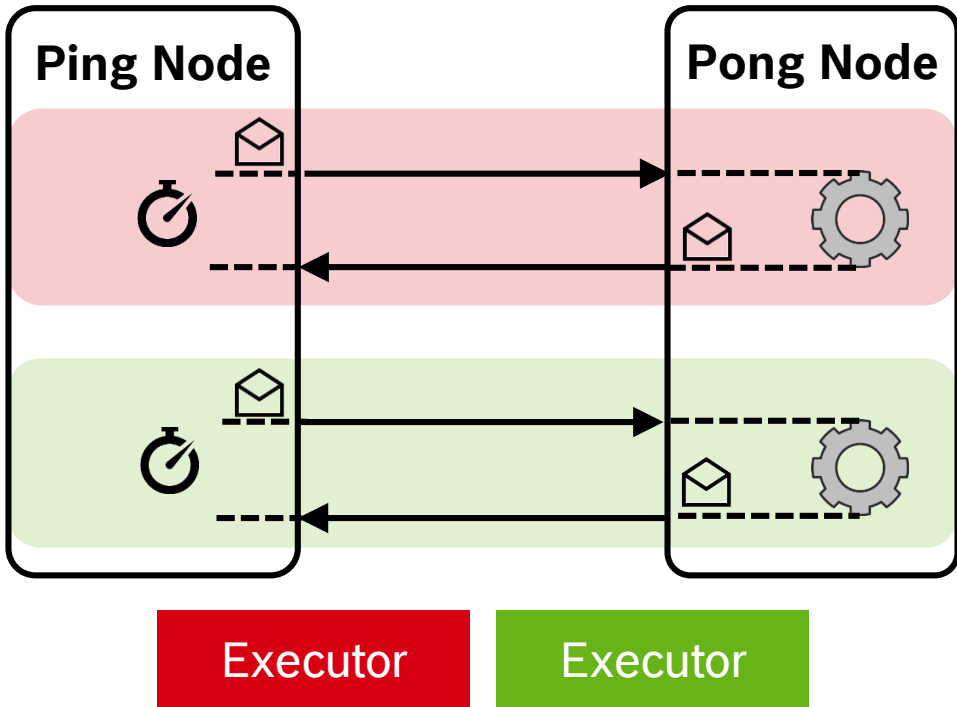
Current Status of ROS 2 - Hands-on Feature Overview

Callback-group-level Executor



Current Status of ROS 2 - Hands-on Feature Overview

Callback-group-level Executor

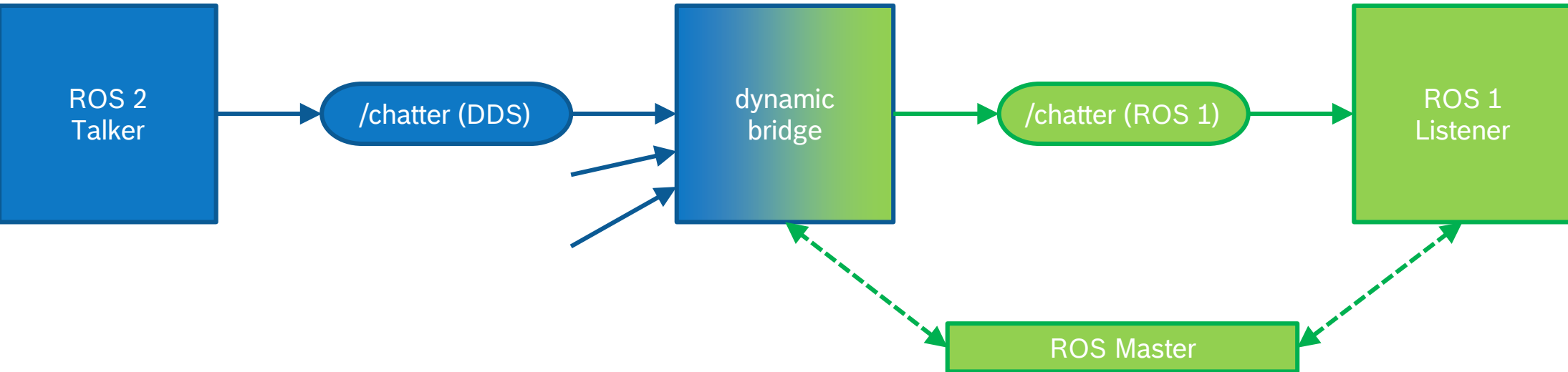


 github.com/boschresearch/ros2_examples → meta-executor

Bridge to ROS 1

Current Status of ROS 2 - Hands-on Feature Overview

Bridge to ROS 1



```
training@training-vm: ~
training@training-vm: ~ 80x24
[INFO] [talker]: Publishing: 'Hello World: 19'
[INFO] [talker]: Publishing: 'Hello World: 20'
[INFO] [talker]: Publishing: 'Hello World: 21'
[INFO] [talker]: Publishing: 'Hello World: 22'
[INFO] [talker]: Publishing: 'Hello World: 23'
[INFO] [talker]: Publishing: 'Hello World: 24'
[INFO] [talker]: Publishing: 'Hello World: 25'
[INFO] [talker]: Publishing: 'Hello World: 26'
[INFO] [talker]: Publishing: 'Hello World: 27'
[INFO] [talker]: Publishing: 'Hello World: 28'
[INFO] [talker]: Publishing: 'Hello World: 29'
[INFO] [talker]: Publishing: 'Hello World: 30'
[INFO] [talker]: Publishing: 'Hello World: 31'
[INFO] [talker]: Publishing: 'Hello World: 32'
[INFO] [talker]: Publishing: 'Hello World: 33'
[INFO] [talker]: Publishing: 'Hello World: 34'
[INFO] [talker]: Publishing: 'Hello World: 35'
[INFO] [talker]: Publishing: 'Hello World: 36'
[INFO] [talker]: Publishing: 'Hello World: 37'
[INFO] [talker]: Publishing: 'Hello World: 38'
[INFO] [talker]: Publishing: 'Hello World: 39'
[INFO] [talker]: Publishing: 'Hello World: 40'
[INFO] [talker]: Publishing: 'Hello World: 41'
```

```
training@training-vm: ~
training@training-vm: ~ 80x24
simple bridge
sudo apt install ros-crystal-ros1-bridge*
source /opt/ros/melodic/setup.bash
source /opt/ros/crystal/local_setup.bash
ros2 run ros1_bridge dynamic_bridge
[INFO] [1536436782.280424195]: Connected to master at [localhost:11311]
[ERROR] [1536436782.280424195]: [rmw_systemState] Failed to contact master at [localhost:11311]. Retrying...
^Csignal_handler(2)
failed to get systemState from ROS 1 master
training@training-vm:~$ ros2 run ros1_bridge dynamic_bridge
[ERROR] [1536436783.73888913]: [rmw_SystemStatePublisher] Failed to contact master at [localhost:11311]. Retrying...
[INFO] [1536436783.73888913]: Connected to master at [localhost:11311]
created 2to1 bridge for topic '/chatter' with ROS 2 type 'std_msgs/String' and ROS 1 type 'std_msgs/String'
[ros1_bridge]: Passing message from ROS 2 std_msgs/String to ROS 1 std_msgs/String (showing msg only once per type)
```

```
roscore http://training-vm:11311/
roscore http://training-vm:11311/ 80x24
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://training-vm:4567/
ros_comm version 1.12.13

SUMMARY
=====
PARAMETERS
 * /roscpp: ros_c++
 * /rosdistro: kinetic
 * /rosversion: 1.12.13

NODES
auto-starting new master
process[master]: started with pid [6842]
ROSLauncher started with pid [6841]
setting /run_id to e16fb132-b3a0-11e8-9a12-080027e8d0ab
process[roscppout1]: started with pid [6867]
started core service [/roscout]
```

```
training@training-vm: ~
training@training-vm: ~ 80x24
[INFO] [1536436434.343627120]: I heard: [Hello World: 19]
[INFO] [1536436435.344156988]: I heard: [Hello World: 20]
[INFO] [1536436436.362277654]: I heard: [Hello World: 21]
[INFO] [1536436437.363714925]: I heard: [Hello World: 22]
[INFO] [1536436438.364618945]: I heard: [Hello World: 23]
[INFO] [1536436439.365073437]: I heard: [Hello World: 24]
[INFO] [1536436440.373557883]: I heard: [Hello World: 25]
[INFO] [1536436441.373529285]: I heard: [Hello World: 26]
[INFO] [1536436442.375625960]: I heard: [Hello World: 27]
[INFO] [1536436443.386129289]: I heard: [Hello World: 28]
[INFO] [1536436444.385019343]: I heard: [Hello World: 29]
[INFO] [1536436445.389552383]: I heard: [Hello World: 30]
[INFO] [1536436446.387802024]: I heard: [Hello World: 31]
[INFO] [1536436447.388582848]: I heard: [Hello World: 32]
[INFO] [1536436448.391535436]: I heard: [Hello World: 33]
[INFO] [1536436449.390184456]: I heard: [Hello World: 34]
[INFO] [1536436450.392357955]: I heard: [Hello World: 35]
[INFO] [1536436451.393027315]: I heard: [Hello World: 36]
[INFO] [1536436452.394499241]: I heard: [Hello World: 37]
[INFO] [1536436453.396442871]: I heard: [Hello World: 38]
[INFO] [1536436454.398241923]: I heard: [Hello World: 39]
[INFO] [1536436455.399472771]: I heard: [Hello World: 40]
[INFO] [1536436456.399933546]: I heard: [Hello World: 41]
```

Hands-On

source /opt/ros/melodic/setup.bash
roscpp

source /opt/ros/melodic/setup.bash
nosrun roscpp_tutorials listener

Security

Current Status of ROS 2 - Hands-on Feature Overview

Security

- ▶ Use of DDS Security mechanisms
- ▶ <https://www.omg.org/spec/DDS-SECURITY/>
 - ▶ Version 1.1 is from July 2018
- ▶ SROS2 put convenience tooling on top:
ros2 security
- ▶ <https://github.com/ros2/sros2>
- ▶ Requires rebuild with
colcon build --cmake-args -DSECURITY=ON
- ▶ Nice demo by Karsten and Mikael from ROSCon '18 (and recently updated) at
https://github.com/Karsten1987/roscon2018/tree/master/confbot_security

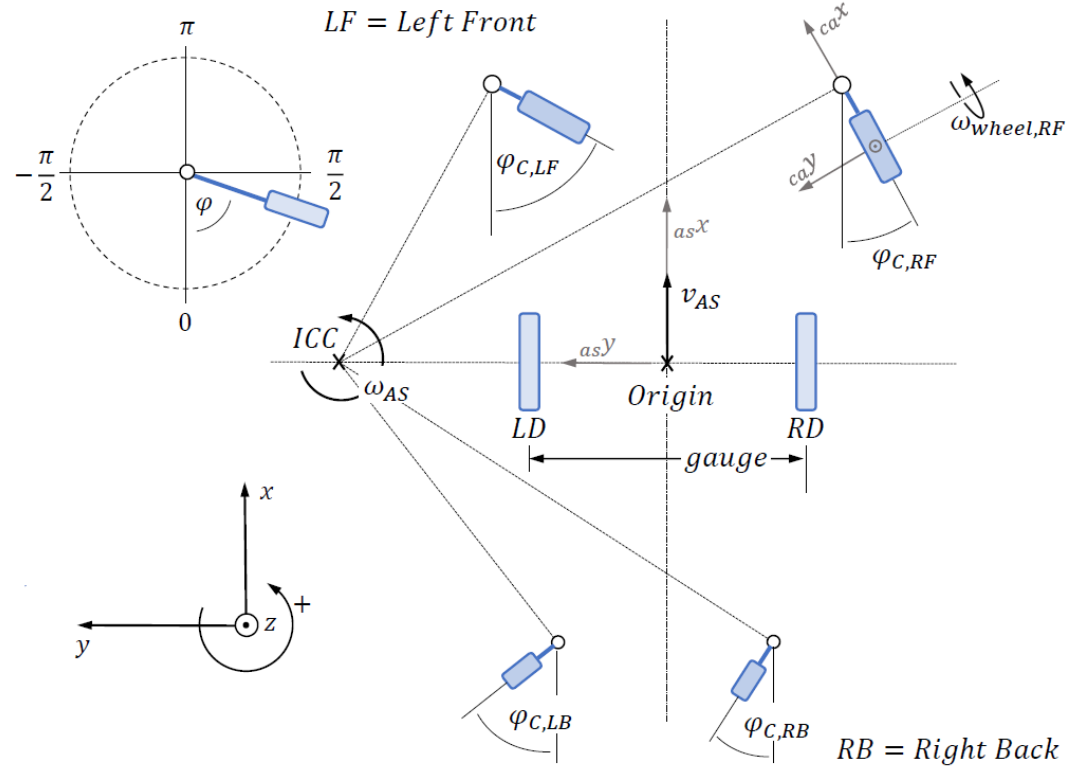
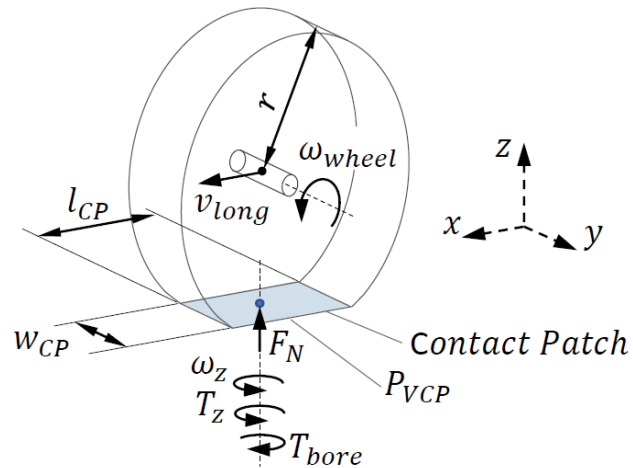
```
<?xml version="1.0" encoding="UTF-8"?>
<policy version="0.1.0"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <profiles>
    <profile ns="/" node="talker">
      <xi:include href="common/node.xml"
        xpointer="xpointer(/profile/*)"/>
      <topics publish="ALLOW" >
        <topic>chatter</topic>
      </topics>
    </profile>
    <profile ns="/" node="listener">
      <xi:include href="common/node.xml"
        xpointer="xpointer(/profile/*)"/>
      <topics subscribe="ALLOW" >
        <topic>chatter</topic>
      </topics>
    </profile>
  </profiles>
</policy>
```


FMI_Adapter

Current Status of ROS 2 - Hands-on Feature Overview


FMI_Adapter

- ▶ Example: ActiveShuttle DevKit at Bosch
- ▶ Problem with lever to caster wheels
 - ▶ Slipping drive wheels
 - ▶ Blocking caster wheels
- ▶ Root cause: Bore friction



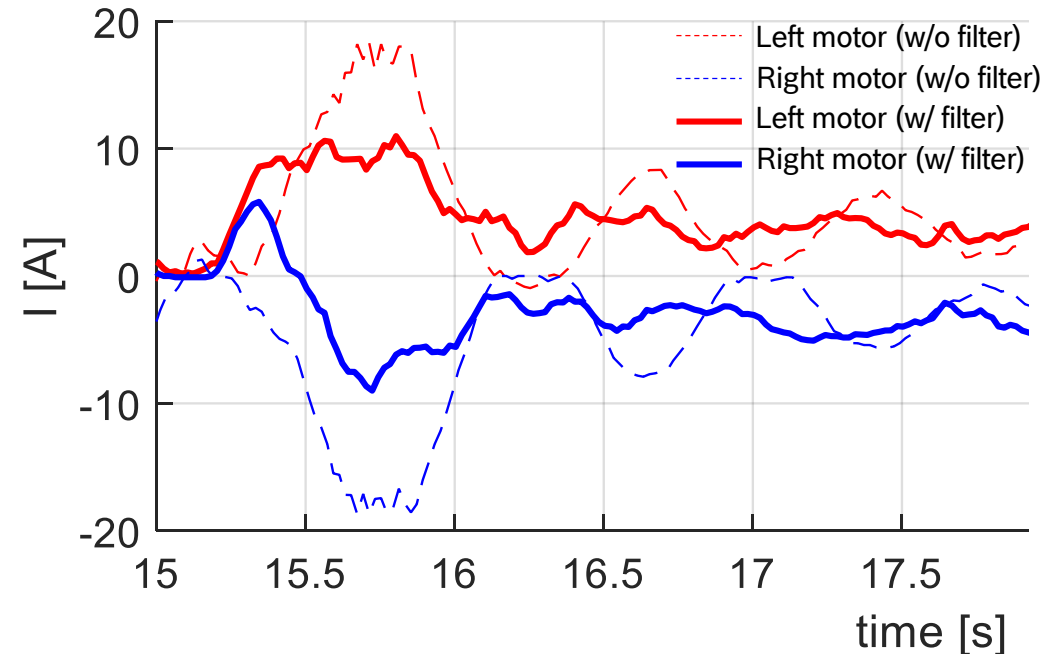
Current Status of ROS 2 - Hands-on Feature Overview

FMI_Adapter

- ▶ Model-based approach: **Path Filter** reduces bore friction
- ▶ Realized with generic  wrapper



- ▶ Reduction of peak currents, jerk, oscillations



 Schröder et al.: “Enhanced Motion Control of a Self-Driving Vehicle Using Modelica, FMI and ROS”, Modelica Conference 2019

Current Status of ROS 2 - Hands-on Feature Overview

FMI_Adapter

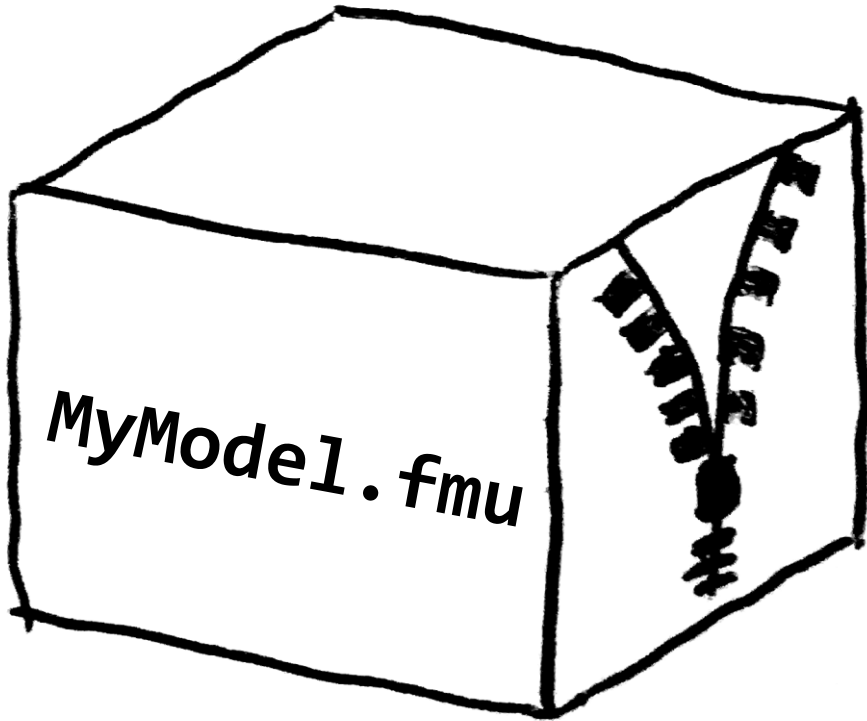


<http://fmi-standard.org/>

D-sim 4C · @Source · Adams · AGX Dynamics · AMESim · ANSYS DesignXplorer · ANSYS SCADE Display
SCADE Suite · ANSYS Simplorer · AUTOSAR Builder · AVL CRUISE · AVL CRUISE M · AVL Model.CONNECT
B&R Automation Studio · BEAST · Building Controls Virtual Test Bed · Cameo Simulation Toolkit (MagicDraw)
ANoe · CarMaker · CarSim · CATIA · ControlBuild · Coral · CosiMate · CPPFMU · Cybernetica CENIT · Cy
DACCOSIM · DAE Tools · DAFUL · DS - FMU Export from Simulink · DS - FMU Import into Simulink · DS
SCALEXIO · dSPACE SYNECT · dSPACE TargetLink · dSPACE VEOS · Dymola · DYNA4 · Easy5 · Ecosim
· EnergyPlus · ETAS - ASCMO · ETAS - FMI-based Integration and Simulation Platform · ETAS - FMU Ge
· ETAS - FMU Generator for Simulink® · ETAS - INCA-FLOW (MiL/SiL Connector) · ETAS - ISOLAR-EVE
U) · ETAS - LABCAR-OPERATOR · Flowmaster · FMI Add-in for Excel · FMI add-on for NI VeriStand · FMI
set for Simulink · FMI Composer · FMI Library · FMI Target for Simulink Coder · FMI Toolbox for MATLAB/S
· FMI4j · FMPy · FMU-proxy · FMUSDK · General Energy Systems (GES) · GT-SUITE · Hopsan · IBM Rati
· ICOS Independent Co-Simulation · IGNITE · INTO-CPS Co-simulation Orchestration Engine (COE) · Jav
odelica.org · LMS Virtual.Lab Motion · MapleSim · MESSINA · MoBA Lab · Morphee · MpCCI CouplingEnvi
· NI LabVIEW · OpenModelica · OPTIMICA Compiler Toolkit · optiSLang · Overture · PROOSIS · Ptolemy
ator · RecurDyn · Scilab/Xcos FMU wrapper · Silver · SIMPACK · Simulation Workbench (SimWB) · Simulat
ing Activate · Squish GUI Tester · SystemModeler · TLK FMI Suite · TLK TISC Suite · TRNSYS FMU Expo
Simulation Framework · TWT FMU Trust Centre · VALDYN · Virtual Engine · WAVE-RT · XFlow · xMOD

Current Status of ROS 2 - Hands-on Feature Overview

FMI_Adapter



Shared library

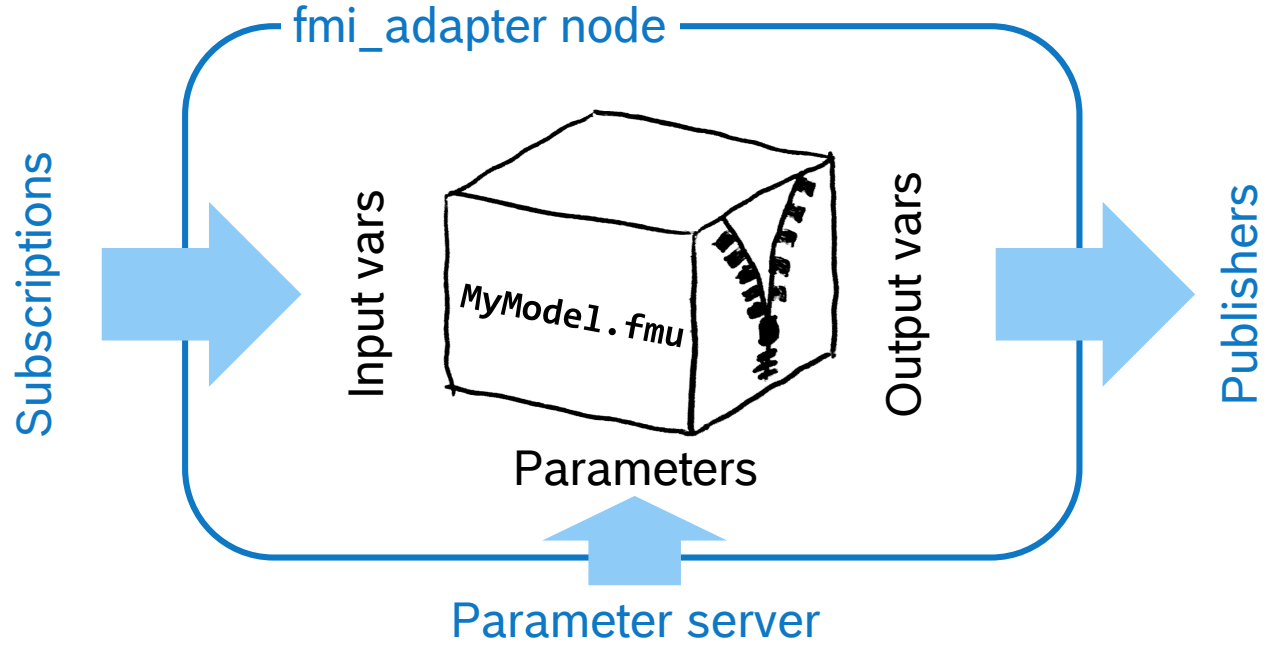
- Equations
- Solver

modelDescription.xml

(C sources)

Current Status of ROS 2 - Hands-on Feature Overview

FMI_Adapter



https://github.com/boschresearch/fmi_adapter_ros2

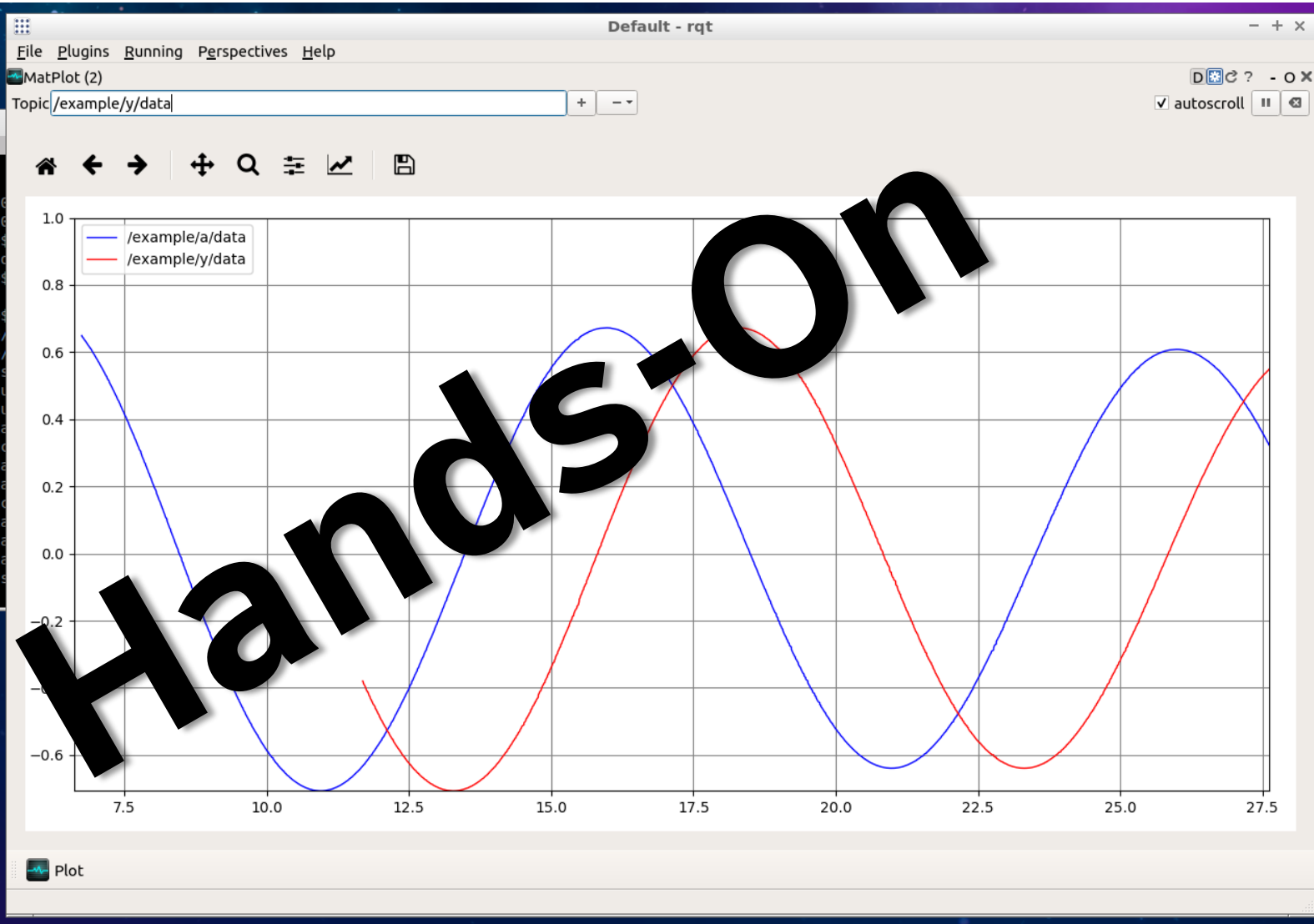
Current Status of ROS 2 - Hands-on Feature Overview

FMI_Adapter

```
FMIAdapter adapter("MyModel.fmu");  
  
vector<string> names =  
    adapter.getParameterNames();  
  
adapter.setInputValue("phi", 4.5);  
adapter.doStepsUntil(ros::Time::now());  
double y = adapter.getOutputValue("y");
```



https://github.com/boschresearch/fmi_adapter_ros2



Build Tooling

Current Status of ROS 2 - Hands-on Feature Overview

Build Tooling (*very brief overview*)

Colcon + Ament + Cmake

- ▶ Colcon – meta-build system
 - ▶ Can be extended for other, non-ROS environments

<https://colcon.readthedocs.io/>

- ▶ Ament – “iteration on catkin”
 - ▶ Many handy CMake macros for linting, package declaration, testing, ...
- ▶ colcon build, test, test-results
- ▶ *but:* ros2 pkg create

```
44 target_link_libraries(${PROJECT_NAME} ${PROJECT_NAME}_fmilib dl)
45 install(DIRECTORY include/ DESTINATION include)
46 install(TARGETS ${PROJECT_NAME} DESTINATION lib)
47
48 add_executable(${PROJECT_NAME}_node src/${PROJECT_NAME}_node.cpp)
49 target_link_libraries(${PROJECT_NAME}_node ${PROJECT_NAME})
50 install(TARGETS ${PROJECT_NAME}_node DESTINATION lib/${PROJECT_NAME})
51
52 install(DIRECTORY launch DESTINATION share/${PROJECT_NAME}/)
53
54
55 if(BUILD_TESTING)
56   find_package(ament_cmake_gtest REQUIRED)
57   ament_add_gtest(${PROJECT_NAME}_test
58     test/${PROJECT_NAME}/FMIAdapterTest.cpp
59     ENV test_fmus_path=${CMAKE_CURRENT_SOURCE_DIR}/test/fmu/
60   )
61   target_include_directories(${PROJECT_NAME}_test PRIVATE include)
62   target_link_libraries(${PROJECT_NAME}_test ${PROJECT_NAME})
63   ament_target_dependencies(${PROJECT_NAME}_test
64     ${PROJECT_NAME}
65     "rclcpp"
66     "rclcpp_lifecycle"
67     "rcutils"
68     "std_msgs"
69   )
70
71   find_package(ament_lint_auto REQUIRED)
72   ament_lint_auto_find_test_dependencies()
73 endif()
74
75 ament_export_include_directories(include)
76 ament_export_libraries(${PROJECT_NAME})
77 ament_export_dependencies(rclcpp)
78 ament_export_dependencies(rclcpp_lifecycle)
```

Contributing

Current Status of ROS 2 - Hands-on Feature Overview

Contributing

ROS 2 core packages and features

- ▶ Long-term roadmap at <https://index.ros.org/doc/ros2/Roadmap/>
- ▶ Meta-ticket for next release: <https://github.com/ros2/ros2/issues/607>

New packages

- ▶ Bloom as with ROS 1:
 - ▶ <https://index.ros.org/doc/ros2/Tutorials/Releasing-a-ROS-2-package-with-bloom/>
 - ▶ Separate build farm instance at <http://build.ros2.org/>

Porting packages

- ▶ Contact ROS 1 maintainer early
- ▶ Decide about refactoring, coding guidelines, ... – maybe start with reformatting PR first

THANK YOU

Dr. Ralph Lange

Robert Bosch GmbH, Corporate Research

Robert-Bosch-Campus 1

71272 Renningen, Germany

ralph.lange@de.bosch.com | github.com/ralph-lange