Hands On

Mikael Arguedas
mikael@openrobotics.org

Karsten Knese
karsten.knese@us.bosch.com

Getting Started

Install ROS 2: https://github.com/ros2/ros2/wiki/Installation

```
$ cd ~/overlay_workspace/src
$ git clone https://github.com/karsten1987/roscon2018.git

$ cd ..
$ source <path/to/ros2/setup.bash>
$ colcon build
```

Creating a package

$ ros2 pkg create mypkg --cpp-node-name my_node

$ vim mypkg/CMakeLists.txt

$ colcon build --packages-select mypkg
$ source install/setup.bash
$ ros2 run mypkg my_node
$ colcon test --packages-select mypkg --event-handlers console_direct+

Getting Started

$ source install/setup.bash

$ vim src/roscon2018/confbot_bringup/launch/confbot_bringup.launch.py

$ ros2 launch confbot_bringup confbot_bringup.launch.py

```
15 import os
16
17 from ament_index_python.packages import get_package_share_directory
18 from launch import LaunchDescription
19 from launch_ros.actions import Node
20
21
22 def generate_launch_description():
23     urdf = os.path.join(
24         get_package_share_directory('confbot_description'),
25         'urdf', 'confbot.urdf')
26
27     return LaunchDescription([
28         Node(package='robot_state_publisher', node_executable='robot_state_publisher',
29              output='screen', arguments=[urdf]),
30         Node(package='confbot_driver', node_executable='confbot_driver', output='screen'),
31         Node(package='confbot_driver', node_executable='twist_publisher', output='screen'),
32         Node(package='confbot_sensors', node_executable='confbot_laser', output='screen'),
33         Node(package='confbot_tools', node_executable='safe_zone_publisher', output='screen')
34     ])
```

Introspect a running system

$ ros2 node list

$ ros2 topic list -t

$ ros2 topic echo /cmd_vel

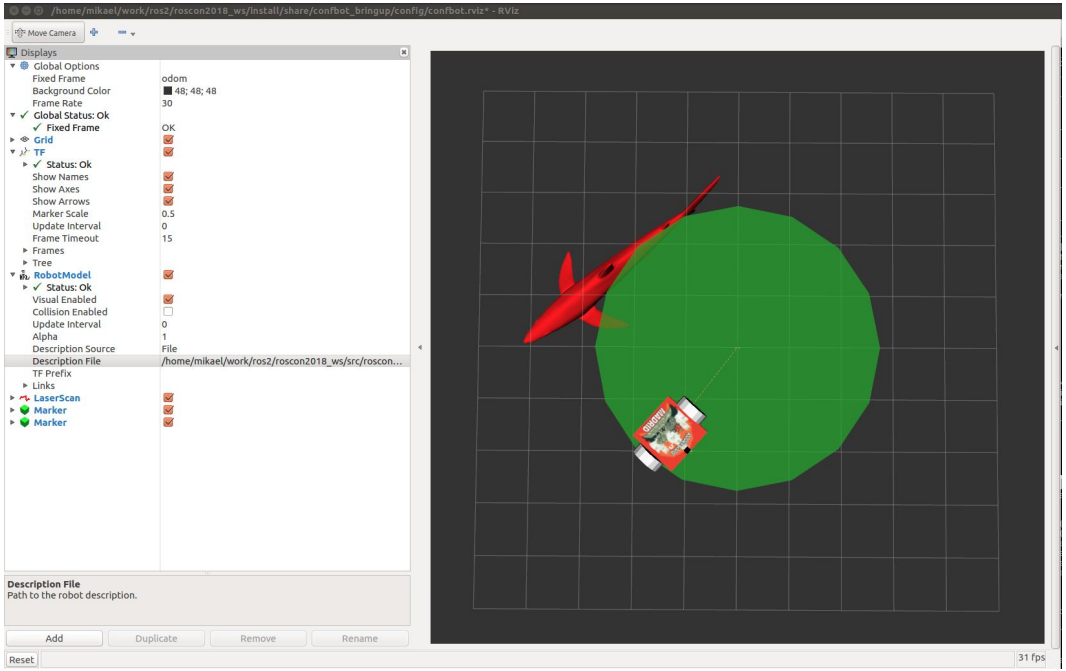
```
$ ros2 node list
RTI Data Distribution Service Evaluation License issued to OSRF
Expires on 5-Nov-2018 See www.rti.com for more information.
/launch_ros
/robot_state_publisher
/confbot_driver
/twist_publisher
/confbot_laser
/safe_zone_publisher
```

```
$ ros2 topic list -t
/clock [rosgraph_msgs/Clock]
/cmd_vel [geometry_msgs/Twist]
/confbot_laser/transition_event [lifecycle_msgs/TransitionEvent]
/danger_zone [visualization_msgs/Marker]
/joint_states [sensor_msgs/JointState]
/parameter_events [rcl_interfaces/ParameterEvent]
/robot_description [std_msgs/String]
/safe_zone [visualization_msgs/Marker]
/tf [tf2_msgs/TFMessage]
/tf_static [tf2_msgs/TFMessage]
```

```
$ ros2 topic echo /cmd_vel
RTI Data Distribution Service E
Expires on 5-Nov-2018 See www.r
linear:
  x: 0.1
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.1
```

Visualize data

$ ros2 run rviz2 rviz2 -d `ros2 pkg prefix confbot_bringup --share`/config/confbot.rviz



Modifying a running system

$ ros2 param list

```
➪ ros2 param list
/confbot_driver:
/confbot_laser:
/launch_ros:
/robot_state_publisher:
/safe_zone_publisher:
/twist_publisher:
  speed
```

$ ros2 param set /twist_publisher speed 0.1

```
Adding fixed segment from base_link to wheel_r_link
[INFO] [twist_publisher]: set new parameter "speed" to "0.200000"
[INFO] [twist_publisher]: changed parameter "speed" to "0.100000"
```

Lifecycle nodes

$ ros2 lifecycle list

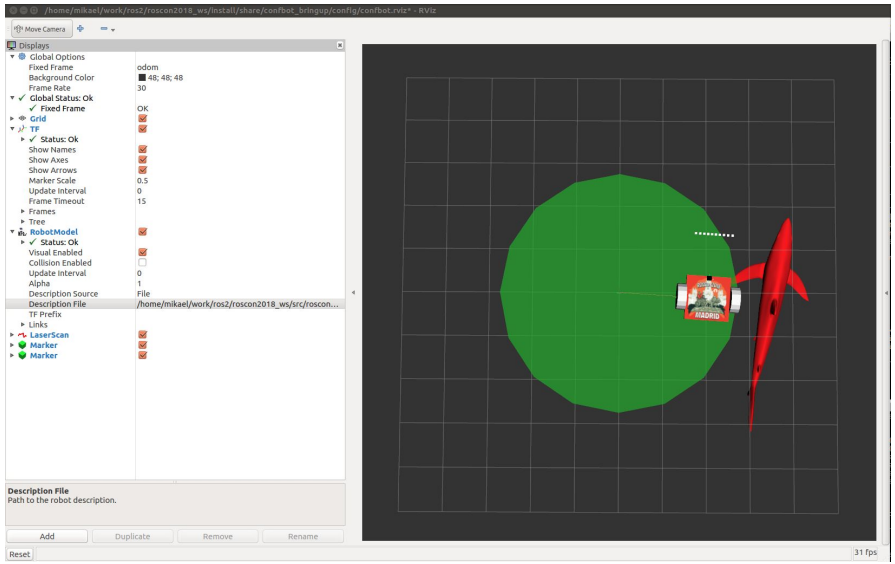
$ ros2 lifecycle set /confbot_laser configure

$ ros2 lifecycle set /confbot_laser activate

```
$ ros2 lifecycle list
/confbot_laser
```

```
[INFO] [confbot_laser]: laser is configured
```

```
[INFO] [confbot_laser]: laser is activated
```



Composing nodes

```cpp
class ConfbotDriver : public rclcpp::Node
{
public:
  explicit ConfbotDriver()
  : Node("confbot_driver")
  {}

  void init()
  {
    rclcpp::Clock::SharedPtr clock = std::make_shared<rclcpp::Clock>(RCL_ROS_TIME);
    msg_.header.stamp = clock->now();
    msg_.header.frame_id = "odom";
    msg_.child_frame_id = "base_link";

    tf_broadcaster_ = std::make_shared<tf2_ros::StaticTransformBroadcaster>(shared_from_this());
    timer_ = this->create_wall_timer(100ms, std::bind(&ConfbotDriver::update_odometry, this));

    cmd_vel_subscriber_ = this->create_subscription<geometry_msgs::msg::Twist>(
      "cmd_vel", std::bind(&ConfbotDriver::update_position, this, std::placeholders::_1));
  }

  void update_odometry() {
    robot_position_.heading += vel_ang_;
    robot_position_.x += 2.0 * cos(robot_position_.heading) * vel_lin_;
    robot_position_.y += 2.0 * sin(robot_position_.heading) * vel_lin_;
    robot_position_.to_transform(msg_.transform);

    tf_broadcaster_->sendTransform(msg_);
  }

  void update_position(std::shared_ptr<geometry_msgs::msg::Twist> twist_msg)
  {
    vel_lin_ = twist_msg->linear.x;
    vel_ang_ = twist_msg->angular.z;
  }
```

```cpp
#include "confbot_driver/confbot_driver.hpp"

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

  auto node = std::make_shared<confbot_driver::ConfbotDriver>();
  node->init();

  rclcpp::spin(node);

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

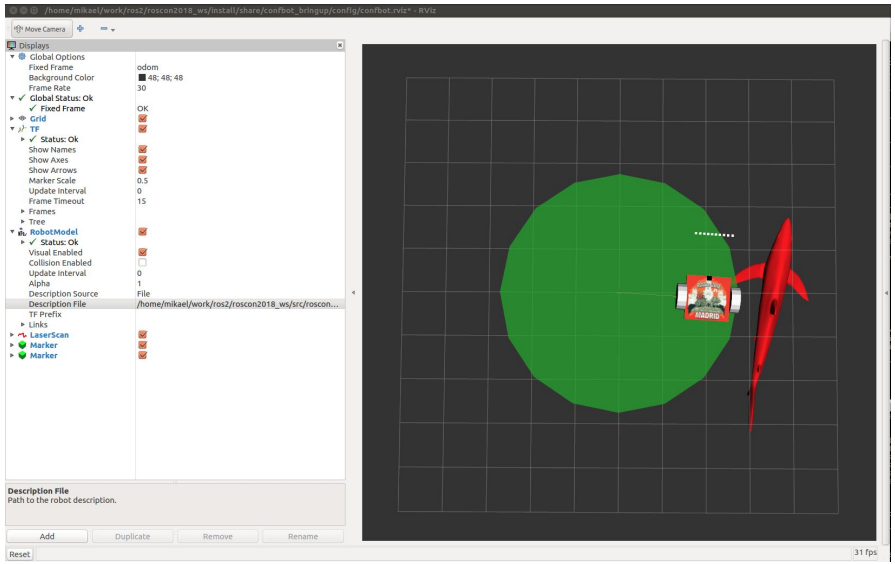
Composing nodes

```
#include "confbot_driver/confbot_driver.hpp"
#include "confbot_driver/twist_publisher.hpp"

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

  auto confbot_driver = std::make_shared<confbot_driver::ConfbotDriver>();
  auto twist_publisher = std::make_shared<confbot_driver::TwistPublisher>();
  confbot_driver->init();

  rclcpp::executors::SingleThreadedExecutor exe;
  exe.add_node(confbot_driver);
  exe.add_node(twist_publisher);
  exe.spin();

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

Tampering with the system

$ ros2 topic pub /cmd_vel geometry_msgs/Twist "linear: {x: 0}" -r 100

A malicious node publishes on /cmd_vel, sending our turtle in the shark's mouth!



Hardening the system

Setting up secure shells for our nodes:

$ export ROS_SECURITY_ENABLE=true

$ export ROS_SECURITY_STRATEGY=Enforce

$ export ROS_SECURITY_ROOT_DIRECTORY=`pwd`/src/roscon2018/confbot_security/confbot_keystore

$ ros2 launch confbot_bringup confbot_bringup.launch.py

$ ros2 run rviz2 rviz2 -d `ros2 pkg prefix confbot_bringup --share`/config/confbot.rviz

Hardening the system

Denying access to unidentified nodes:

$ ros2 topic pub /cmd_vel geometry_msgs/Twist "linear: {x: 0}" -r 100

```
$ ros2 topic pub /cmd_vel geometry_msgs/Twist "linear: {x: 0}" -r 100
Unknown error creating node: SECURITY ERROR: unable to find a folder matching the node name in the "ROS_SECURITY_
ROOT_DIRECTORY" directory while the requested security strategy requires it, at /home/mikael/work/ros2/bouncy_ws/
src/ros2/rcl/rcl/src/rcl/node.c:326
```

$ ros2 run confbot_driver twist_publisher __node:=my_hacky_node

```
$ ros2 run confbot_driver twist_publisher __node:=my_hacky_node
terminate called after throwing an instance of 'rclcpp::exceptions::RCLError'
  what():  failed to initialize rcl node: SECURITY ERROR: unable to find a folder matching the node name in the "
ROS_SECURITY_ROOT_DIRECTORY" directory while the requested security strategy requires it, at /home/mikael/work/ro
s2/bouncy_ws/src/ros2/rcl/rcl/src/rcl/node.c:326
```

Restricting individual node permissions

Specifying what the nodes is allowed to do (topics it's to publish/subscribe to):

"safe_zone_publisher" is only allowed to publish on "danger_zone" and "safe_zone"

```
# safe_zone_publisher_policies.yaml
nodes:
  safe_zone_publisher:
    topics:
      danger_zone:
        allow: p # can publish on danger_zone
      safe_zone:
        allow: p # can publish on safe_zone
```

$ ros2 security create_permission confbot_keystore safe_zone_publisher safe_zone_publisher_policies.yaml

Restricting individual node permissions

Generated permission file (extract):

```
39          <publish>^M
40            <partitions>^M
41              <partition></partition>^M
42            </partitions>^M
43            <topics>^M
44              <topic>rt/danger_zone</topic>^M
45            </topics>^M
46          </publish>^M
47          <publish>^M
48            <partitions>^M
49              <partition></partition>^M
50            </partitions>^M
51            <topics>^M
52              <topic>rt/safe_zone</topic>^M
53            </topics>^M
54          </publish>^M
```

Injecting malicious code in node

```
diff --git a/confbot_tools/confbot_tools/safe_zone_publisher.py b/confbot_
tools/confbot_tools/safe_zone_publisher.py
index 3eaadab..2c0ad08 100644
--- a/confbot_tools/confbot_tools/safe_zone_publisher.py
+++ b/confbot_tools/confbot_tools/safe_zone_publisher.py
@@ -4,6 +4,7 @@ import rclpy
 from rclpy.clock import Clock
 from rclpy.node import Node

+from geometry_msgs.msg import Twist
 from visualization_msgs.msg import Marker


@@ -13,6 +14,7 @@ class SafeZonePublisher(Node):
         super().__init__('safe_zone_publisher')

         self.pub = self.create_publisher(Marker, 'safe_zone')
+        self.hacked_pub = self.create_publisher(Twist, 'cmd_vel')
         self.shark_pub = self.create_publisher(Marker, 'danger_zone')
         timer_period = 1.0
         self.tmr = self.create_timer(timer_period, self.timer_callback)
```

Run compromised node

```
RTI_Security_AccessControl_check_create_topic:topic not allowed: cannot be published or subscribed
DDS_DomainParticipant_checkCreateTopic:!security function check_create_topic
DDS_DomainParticipant_create_topic_disabled:SECURITY ERROR: denied permissions
DDSTopic_impl::create:!create topic
Traceback (most recent call last):
  File "./confbot_tools/confbot_tools/safe_zone_publisher_hacked.py", line 93, in <module>
    main()
  File "./confbot_tools/confbot_tools/safe_zone_publisher_hacked.py", line 84, in main
    node = SafeZonePublisher()
  File "./confbot_tools/confbot_tools/safe_zone_publisher_hacked.py", line 17, in __init__
    self.pub = self.create_publisher(Twist, 'cmd_vel')
  File "/home/mikael/work/ros2/bouncy_ws/install_debug/rclpy/lib/python3.5/site-packages/rclpy/node.py", line 222
, in create_publisher
    self.handle, msg_type, topic, qos_profile.get_c_qos_profile())
RuntimeError: Failed to create publisher: failed to create topic, at /home/mikael/work/ros2/bouncy_ws/src/ros2/rm
w_connext/rmw_connext_cpp/src/rmw_publisher.cpp:164, at /home/mikael/work/ros2/bouncy_ws/src/ros2/rcl/rcl/src/rcl
/publisher.c:173
```

(some) ROS 2 features not covered in this talk

- In Bouncy
  - Bridging communication between ROS 1 and ROS 2
  - Intra process communication
  - Node Parameters via yaml files
  - Various command line tools (ros2 security, ros2 service...)

- On master (Crystal)
  - message filters / image_transport
  - gazebo_ros2_pkgs
  - rosbag
  - ROS Time

What’s next?

- ROS 2 Crystal Release (Dec 2018)

https://github.com/ros2/ros2/wiki/Roadmap

Questions?