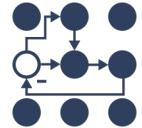
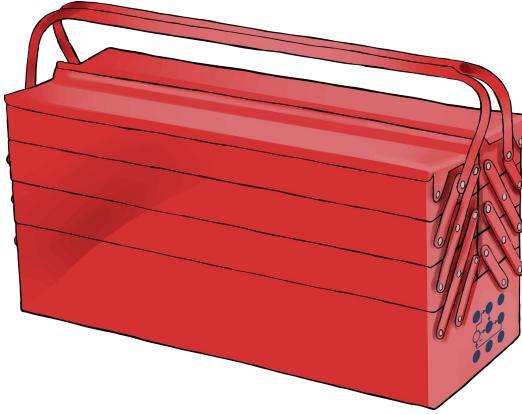
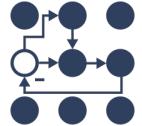


# A practitioner's guide to ros2\_control





bence.magyar@five.ai

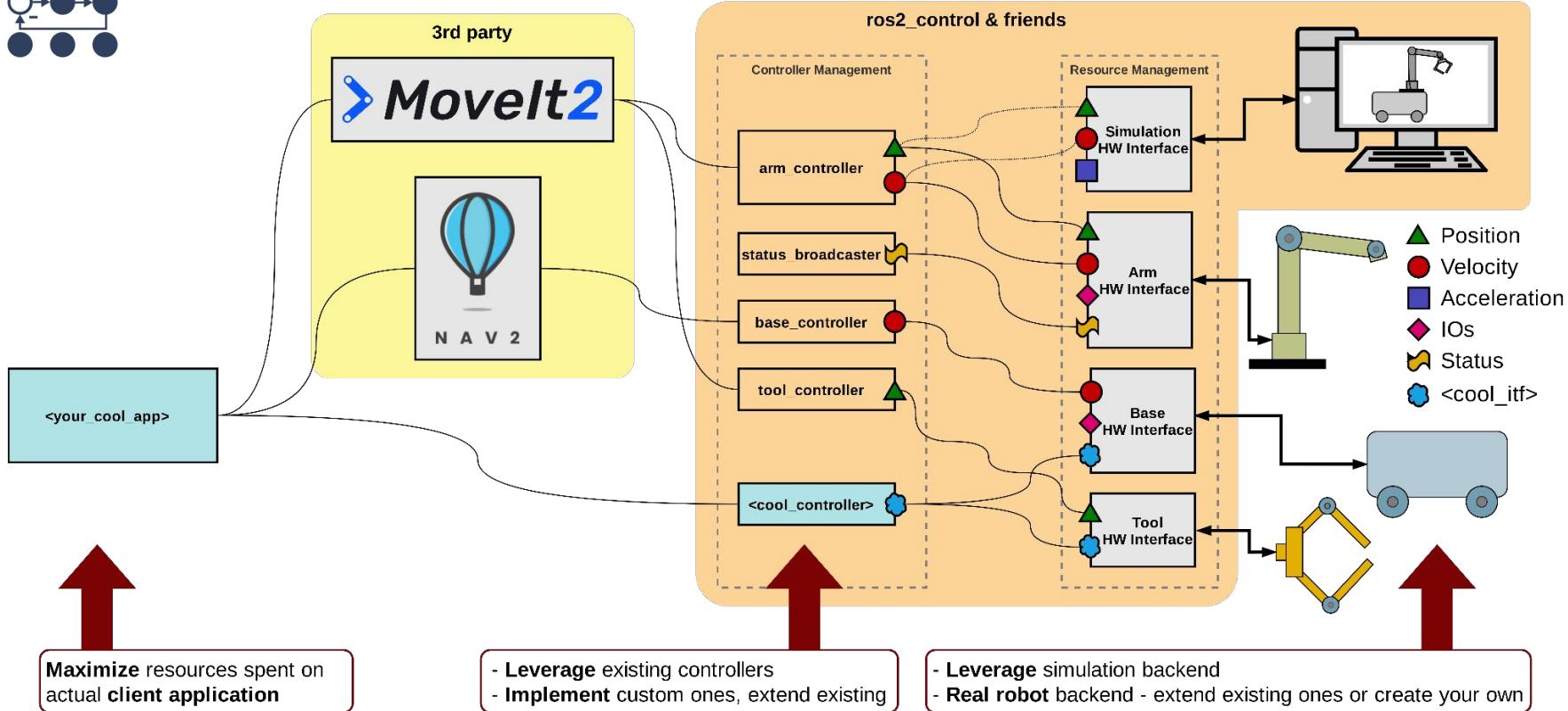
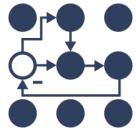
## Bence Magyar

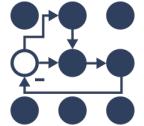
- Principal Software Engineer at FiveAI / Bosch UK



- ros\_control and ros2\_control maintainer
- PhD in Robotics from Heriot-Watt University, UK







# History

pr2\_controller\_manager  
(pr2\_mechanism)

2009

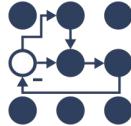


ros\_control  
2012/2017



ros2\_control  
2017/2022



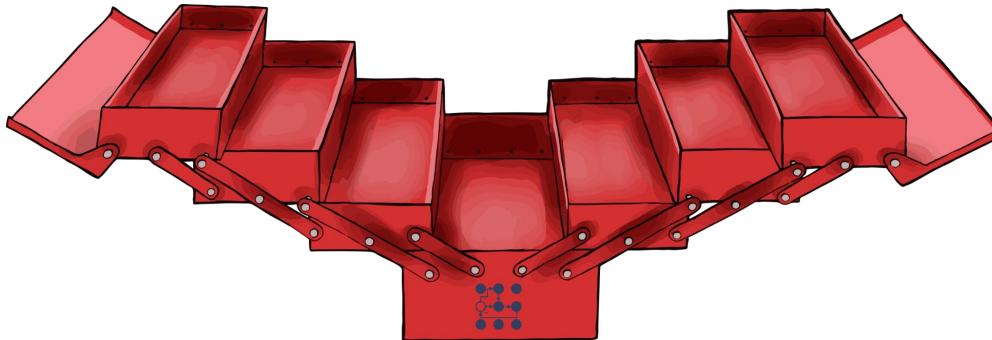


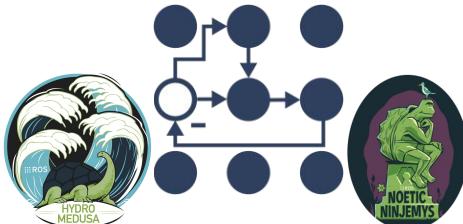
ros2\_control

no ros2\_control

- Reuse hardware drivers
- Free controllers!
- Free simulator integration
- MoveIt2, rviz, Nav2
- Manage your hardware access like a pro
- Cheat: things you've never thought about

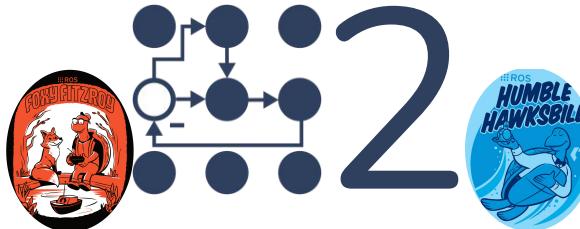
- I have controllers on an embedded board already
- “I know control better than X” and have controllers already written
- “I have just this one robot, why bother with this complexity?”
- “Hah! I already have ros\_control”





- General, robot-agnostic framework
- Collection of official controllers, defining de-facto standard ROS interfaces to 3rd party
- Off-the-shelf Gazebo integration
- Stability
- Supported joint interfaces: position, velocity, effort
- Code complexity high, templating and inheritance
- Controller lifecycle inspired by Orocос, custom
- Unclear semantics: everything is the RobotHW or controller
- Opt-in Hardware Composition
- RobotHW and boilerplate code

- 
- 
- 
- New features!
- Supported joint interfaces: no limitations
- Code leaner, more modern C++
- Controller lifecycle via ROS2 LifecycleNode
- [System|Actuator|Sensor]Component, Controller and Broadcaster
- Hardware Composition is first class citizen
- Default `ros2_control_node`
- Hardware lifecycle
- Synchronous but variable rate for controllers
- Chaining controllers
- Asynchronous controllers\* & hardware\*
- Hardware failure handling
- Emergency stop handling\*

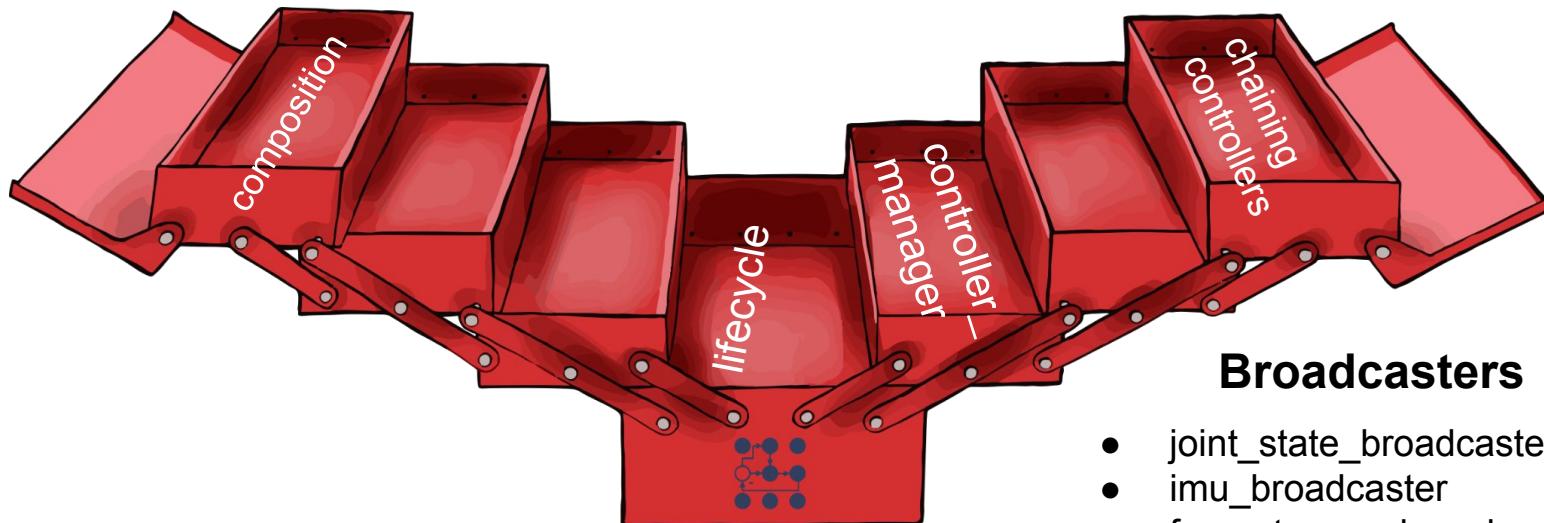


## Hardware components

- SystemComponent
- SensorComponent
- ActuatorComponent

## Controllers

- joint\_trajectory\_controller
- diff\_drive\_controller
- forwarding controllers
- gripper\_controllers



- joint\_state\_broadcaster
- imu\_broadcaster
- force\_torque\_broadcaster

# ros2\_control CLI - Integrated with ROS2 CLI

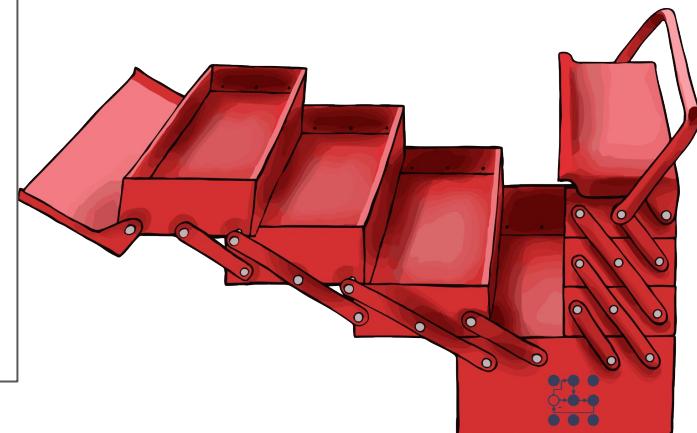
```
$ ros2 control list_hardware_interfaces
```

command interfaces

```
flange_gpios/digital_out_1 [available] [unclaimed]
flange_gpios/digital_out_2 [available] [unclaimed]
joint1/position [available] [claimed]
joint1/velocity [available] [unclaimed]
joint2/position [available] [claimed]
joint2/velocity [available] [unclaimed]
```

state interfaces

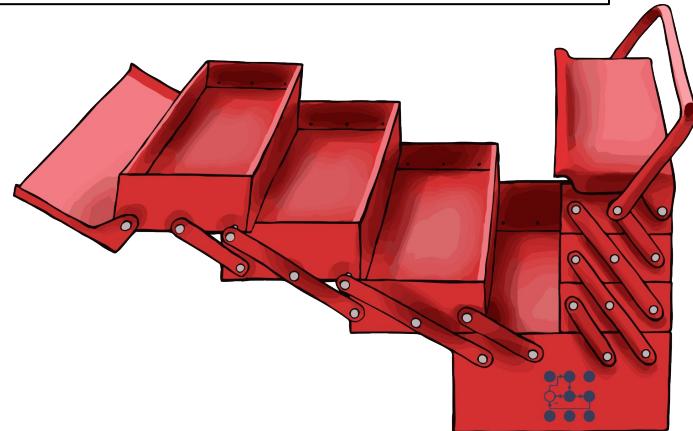
```
flange_gpios/digital_in_1
flange_gpios/digital_in_2
flange_gpios/digital_out_1
flange_gpios/digital_out_2
joint1/effort
joint1/position
joint1/velocity
joint2/effort
joint2/position
joint2/velocity
```



# ros2\_control CLI - Integrated with ROS2 CLI

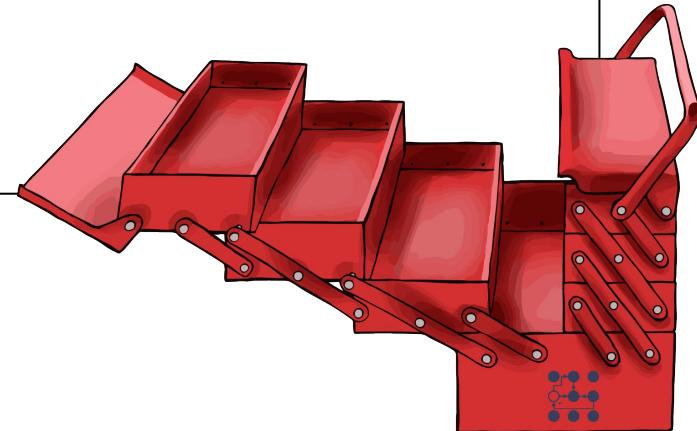
```
$ ros2 control list_controllers
```

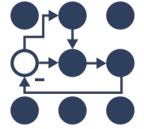
```
joint_state_broadcaster[joint_state_broadcaster/JointStateBroadcaster] active
forward_position_controller[forward_command_controller/ForwardCommandController] active
joint_trajectory_controller[joint_trajectory_controller/JointTrajectoryController] inactive
```



# ros2\_control CLI - Integrated with ROS2 CLI

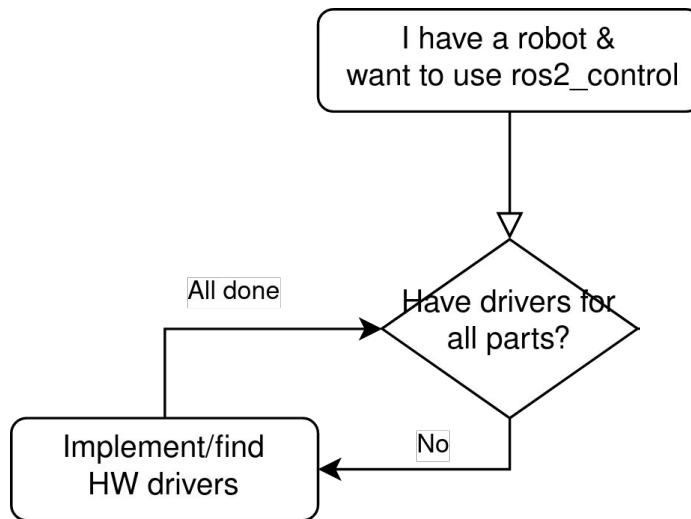
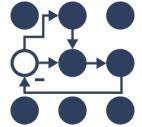
```
$ ros2 control list_controllers -v
...
forward_position_controller[forward_command_controller/ForwardCommandController] active
  claimed interfaces:
    joint1/position
    joint2/position
  required command interfaces:
    joint1/position
    joint2/position
  required state interfaces:
  chained to interfaces:
  exported reference interfaces:
...
...
```

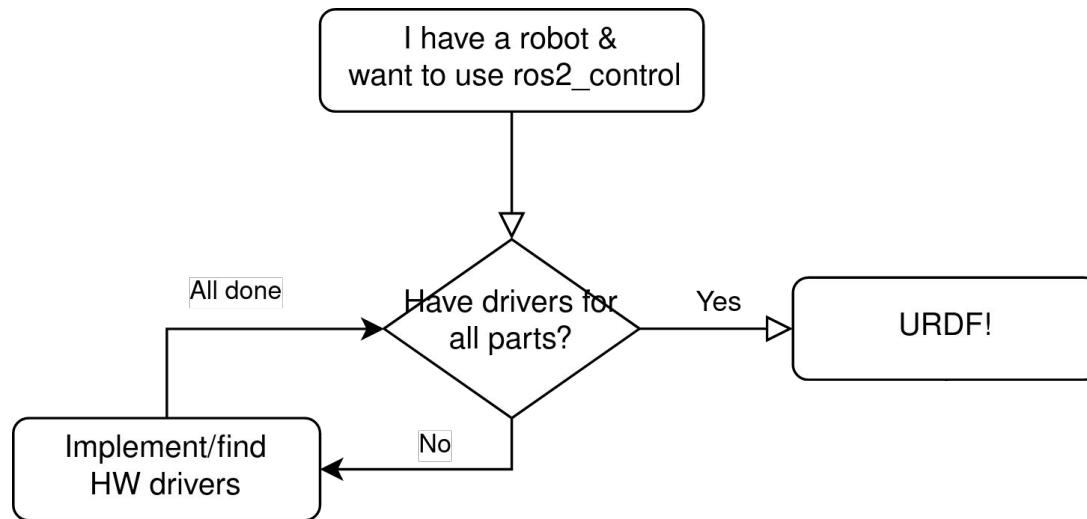
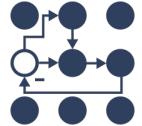




I have a robot &  
want to use ros2\_control

Have drivers for  
all parts?





# URDF extension with <ros2\_control>-tag

```
<ros2_control name="robot" type="system">

    <hardware>
        <plugin>robot_package/Robot</plugin>
        <param name="hardware_parameter">some_value</param>
    </hardware>

    <joint name="joint_first">
        <command_interface name="position"/>
        <state_interface name="acceleration"/>
    </joint>
    .
    .
    <joint name="joint_last">
        <command_interface name="velocity">
            <param name="min">-1</param>
            <param name="max">1</param>
        </command_interface>
        <state_interface name="temperature"/>
    </joint>

    <sensor name="tcp_sensor">
        <state_interface name="sensing_interface"/>
        <param name="sensor_parameter">another_value</param>
    </sensor>

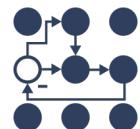
    <gpio name="flange_IOs">
        <command_interface name="digital_output" data_type="bool" size="8" />
        <state_interface name="digital_output" data_type="bool" size="8" />
        <command_interface name="analog_output" data_type="double" size="2" />
        <state_interface name="analog_output" data_type="double" size="2" />

        <state_interface name="digital_input" data_type="bool" size="4" />
        <state_interface name="analog_input" data_type="double" size="4" />
    </gpio>

    <gpio name="rrbot_status">
        <state_interface name="mode" data_type="int"/>
        <state_interface name="bit" data_type="bool" size="4"/>
    </gpio>

    <joint name="tool">
        <command_interface name="command"/>
    </joint>

</ros2_control>
```



# URDF extension with `<ros2_control>`-tag

```
<ros2_control name="robot" type="system">

  <hardware>
    <plugin>robot_package/Robot</plugin>
    <param name="hardware_parameter">some_value</param>
  </hardware>

  <joint name="joint_first">
    <command_interface name="position"/>
    <state_interface name="acceleration"/>
  </joint>
  .
  .
  <joint name="joint_last">
    <command_interface name="velocity">
      <param name="min">-1</param>
      <param name="max">1</param>
    </command_interface>
    <state_interface name="temperature"/>
  </joint>

  <sensor name="tcp_sensor">
    <state_interface name="sensing_interface"/>
    <param name="sensor_parameter">another_value</param>
  </sensor>

  <gpio name="rrbot_status">
    <state_interface name="mode" data_type="int"/>
    <state_interface name="bit" data_type="bool" size="4"/>
  </gpio>

</ros2_control>

<ros2_control name="tool" type="actuator">

  <hardware>
    <plugin>tool_package/Tool</plugin>
    <param name="hardware_parameter">some_value</param>
  </hardware>

  <joint name="tool">
    <command_interface name="command"/>
  </joint>
</ros2_control>
```

```
<ros2_control name="robot" type="system">

  <hardware>
    <plugin>robot_package/Robot</plugin>
    <param name="hardware_parameter">some_value</param>
  </hardware>

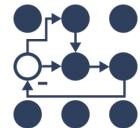
  <joint name="joint_first">
    <command_interface name="position"/>
    <state_interface name="acceleration"/>
  </joint>
  .
  .
  <joint name="joint_last">
    <command_interface name="velocity">
      <param name="min">-1</param>
      <param name="max">1</param>
    </command_interface>
    <state_interface name="temperature"/>
  </joint>

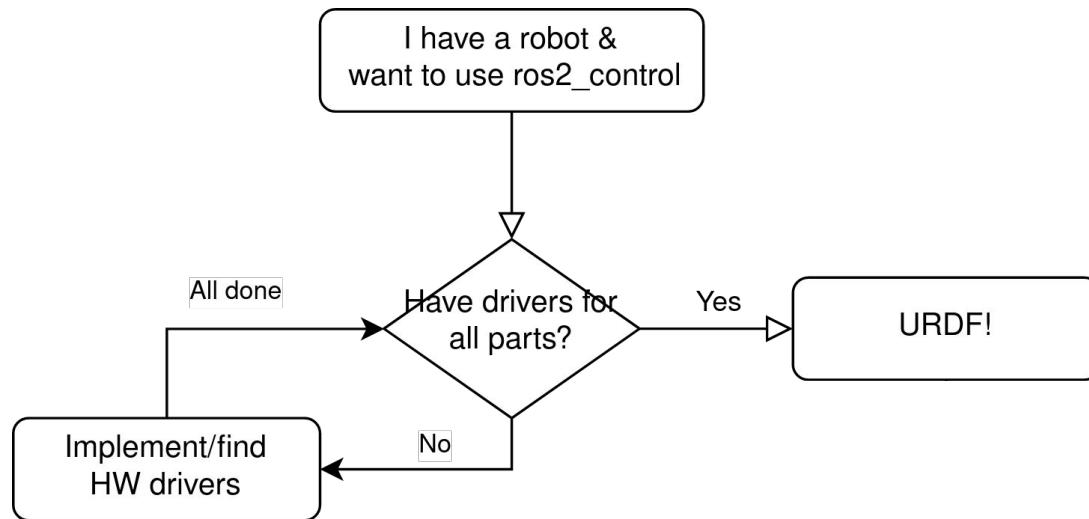
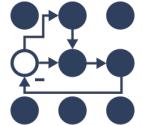
  <sensor name="tcp_sensor">
    <state_interface name="sensing_interface"/>
    <param name="sensor_parameter">another_value</param>
  </sensor>

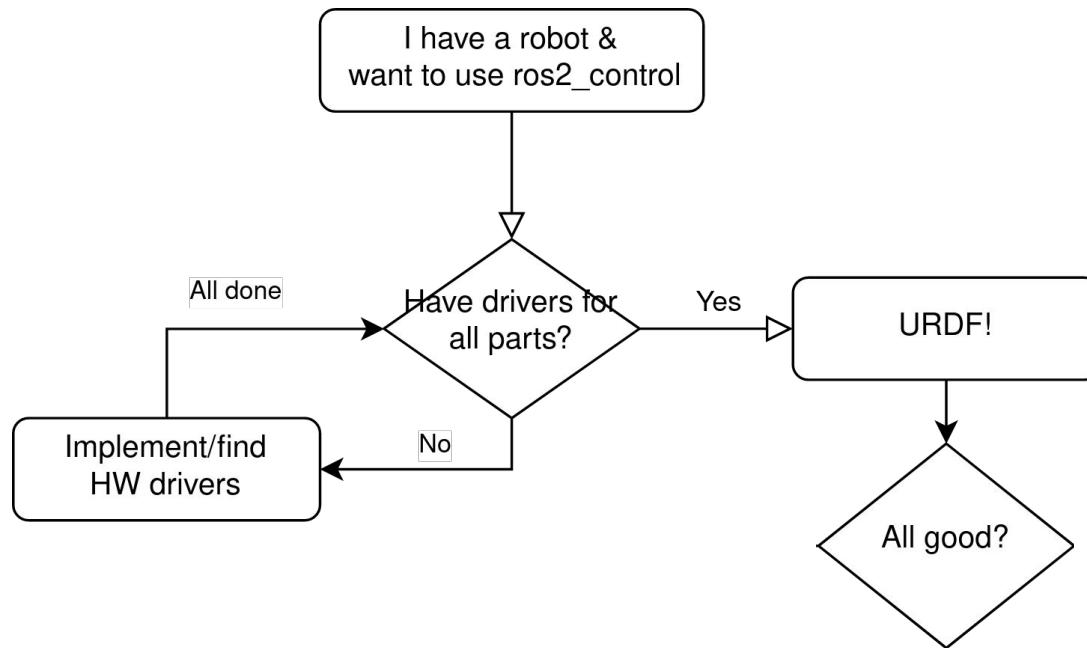
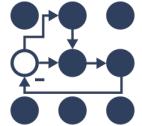
  <gpio name="flange_IOs">
    <command_interface name="digital_output" data_type="bool" size="8" />
    <state_interface name="digital_output" data_type="bool" size="8" />
    <command_interface name="analog_output" data_type="double" size="2" />
    <state_interface name="analog_output" data_type="double" size="2" />
    <state_interface name="digital_input" data_type="bool" size="4" />
    <state_interface name="analog_input" data_type="double" size="4" />
  </gpio>

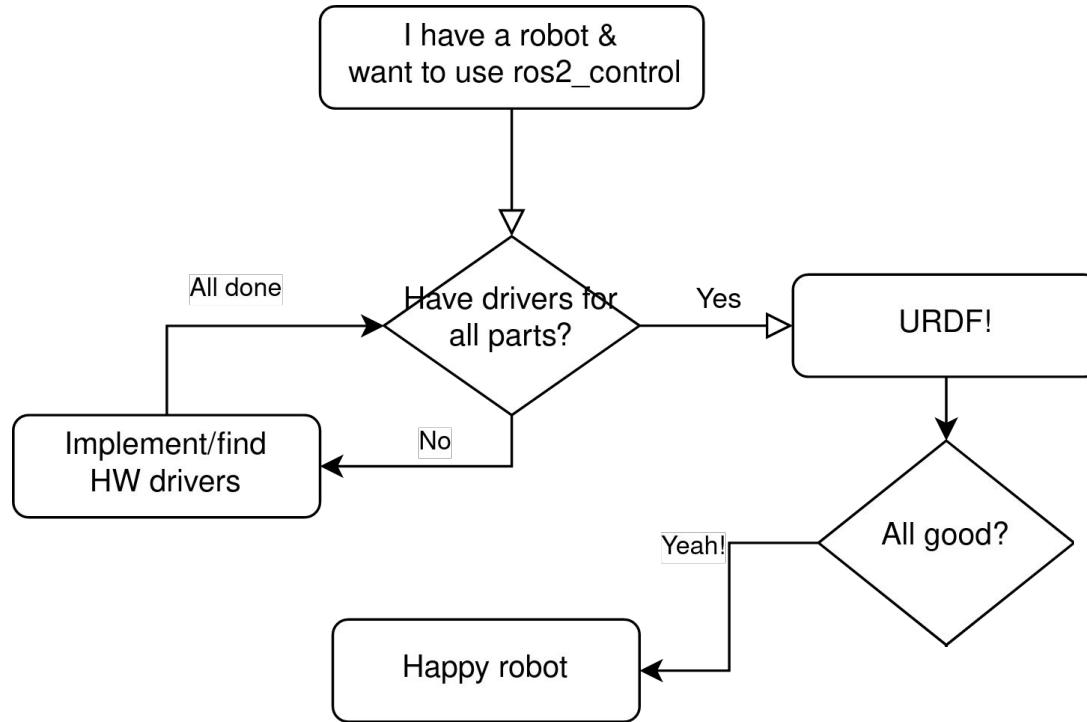
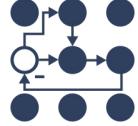
  <gpio name="rrbot_status">
    <state_interface name="mode" data_type="int"/>
    <state_interface name="bit" data_type="bool" size="4"/>
  </gpio>

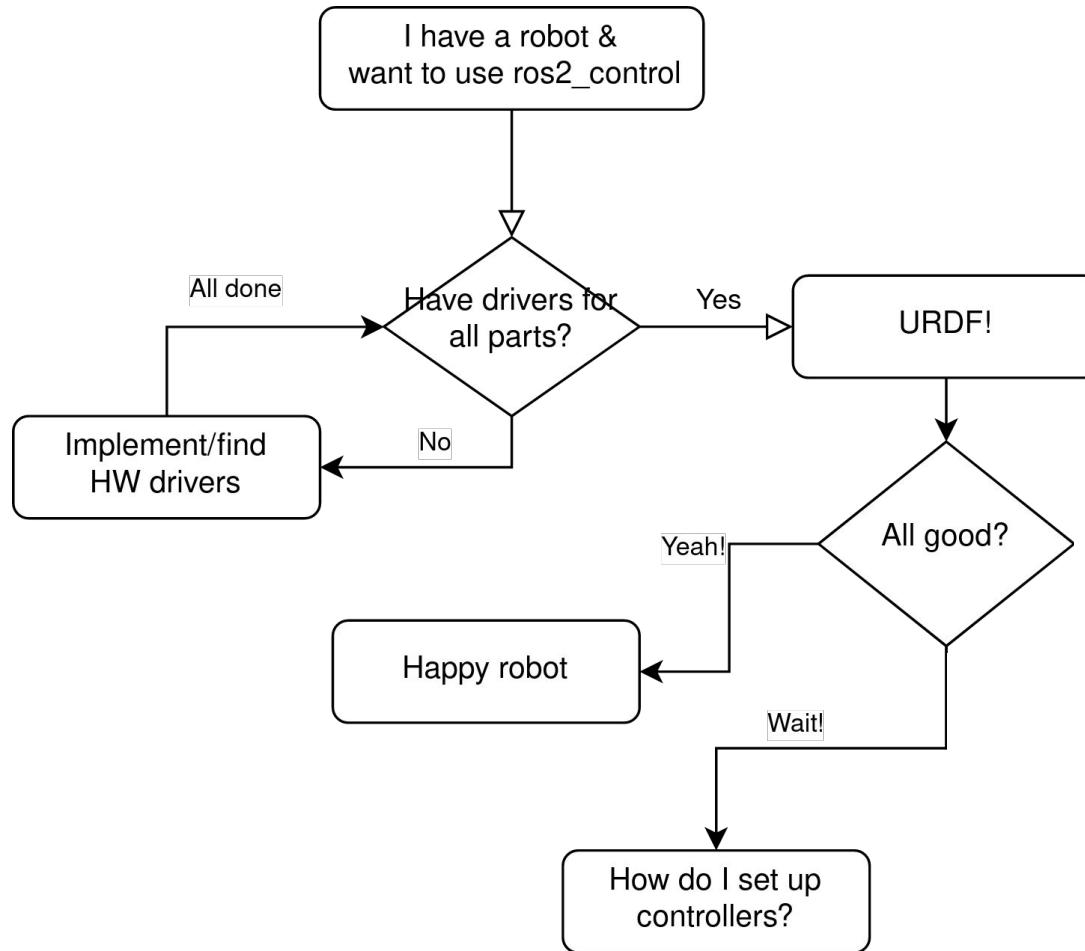
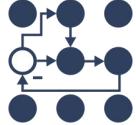
  <joint name="tool">
    <command_interface name="command"/>
  </joint>
</ros2_control>
```





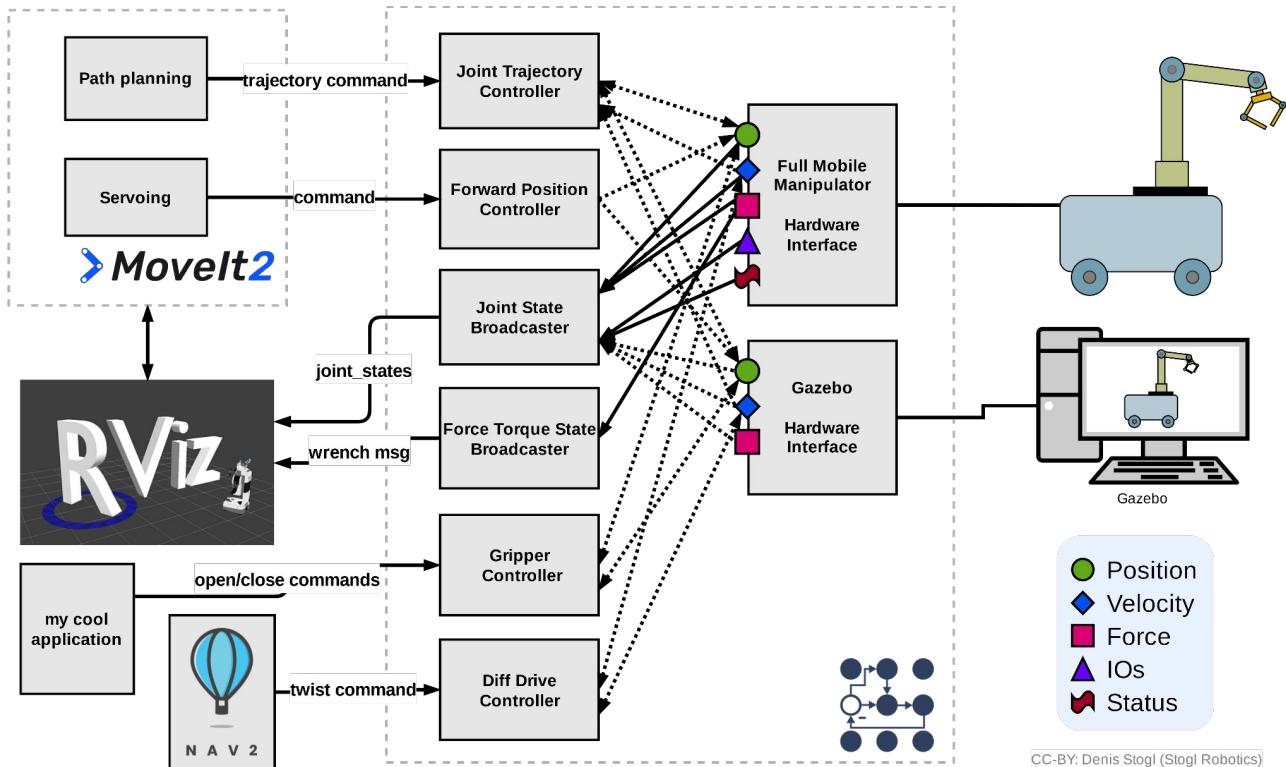




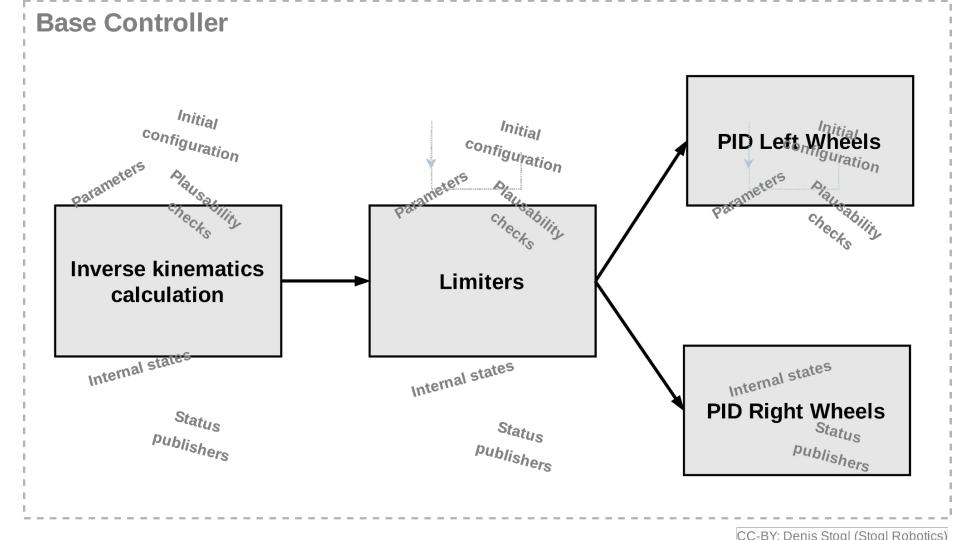
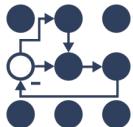
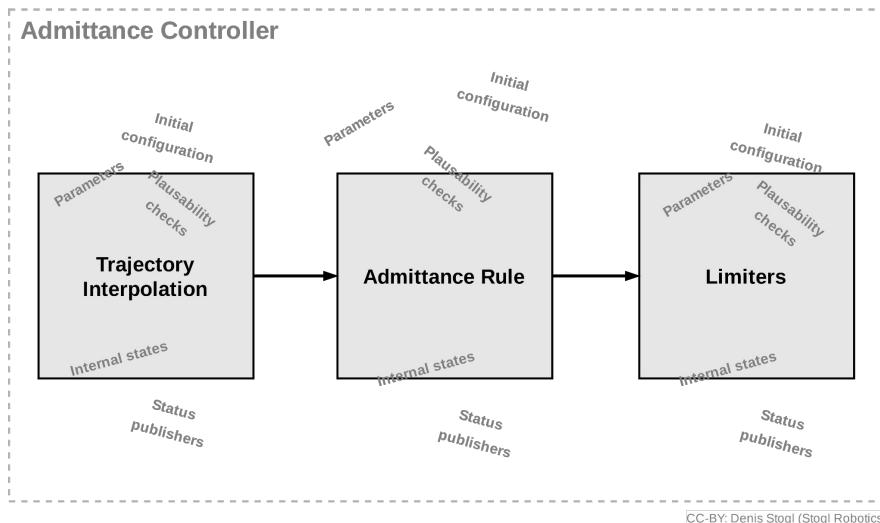


# Configuring standard controllers

```
controller_manager:  
  update_rate: 500 # Hz  
  
joint_trajectory_controller:  
  type: joint_trajectory_controller/JointTrajectoryController  
  
forward_position_controller:  
  type: position_controllers/JointGroupPositionController  
  
joint_state_broadcaster:  
  type: joint_state_broadcaster/JointStateBroadcaster  
  
force_torque_sensor_broadcaster:  
  type: force_torque_sensor_broadcaster/ForceTorqueStateBroadcaster  
  
gripper_controller:  
  type: position_controllers/GripperActionController  
  
diff_drive_controller:  
  type: diff_drive_controller/DiffDriveController  
  
  
joint_trajectory_controller:  
  joints:  
    - joint1  
    - ...  
  command_interfaces:  
    - position  
  state_interfaces:  
    - position  
    - velocity  
  
forward_position_controller:  
  joints:  
    - joint1  
    - ...  
  
force_torque_sensor_broadcaster:  
  sensor_name: tcp_fts_sensor  
  frame_id: tool0  
  topic_name: ft_data  
  
gripper_controller:  
  joints:  
    - gripper_joint  
  command_interface: position  
  
diff_drive_controller:  
  left_wheel_names:  
    - left_wheel_1  
  ...
```

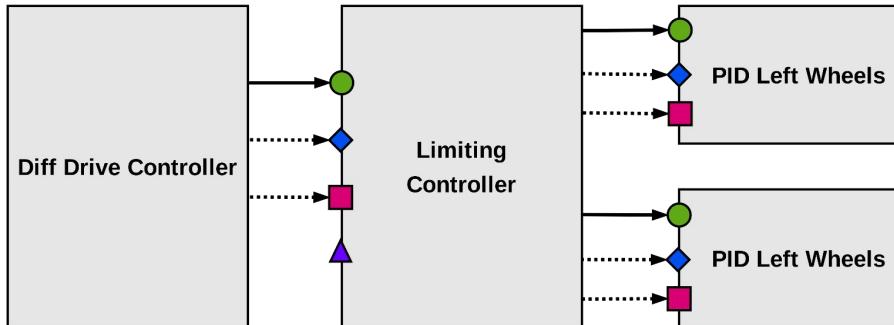


# This can end-up in convoluted and complex controllers...



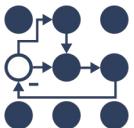
# Using controller-chaining...

Controller Manager



CC-BY: Denis Stogl (Stogl Robotics)

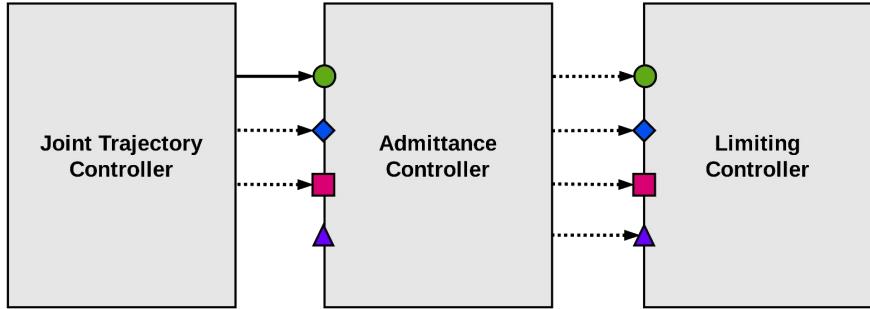
- Position
- ◆ Velocity
- Acceleration
- ▲ Force



```
controller_manager:  
  update_rate: 500 # Hz  
  
diff_drive_controller:  
  type: diff_drive_controller/DiffDriveController  
  
limiting_controller:  
  type: limiting_controllers/JointLimitingController  
  
pid_left_wheels:  
  type: pid_controllers/PIDController  
  
pid_right_wheels:  
  type: pid_controllers/PIDController  
  
  
diff_drive_controller:  
  left_wheel_names:  
    - left_wheel_1  
    ...  
  
# export reference interfaces: "<controller_name>/<joint_name>/<interface_name>"  
limiting_controller:  
  joints:  
    - left_wheel_1  
    - ...  
  command_joints:  
    - pid_left_wheels/joint1/velocity  
    - ...  
    - pid_right_wheels/joint1/velocity  
    - ...  
  interfaces:  
    - velocity  
  
# export reference interfaces: "<controller_name>/<joint_name>/<interface_name>"  
pid_left_wheels:  
  joints:  
    - left_wheel_1  
    ...  
  
# export reference interfaces: "<controller_name>/<joint_name>/<interface_name>"  
pid_right_wheels:  
  joints:  
    - right_wheel_1  
    ...
```

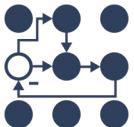
# Using controller-chaining...

Controller Manager



CC-BY: Denis Stogl (Stogl Robotics)

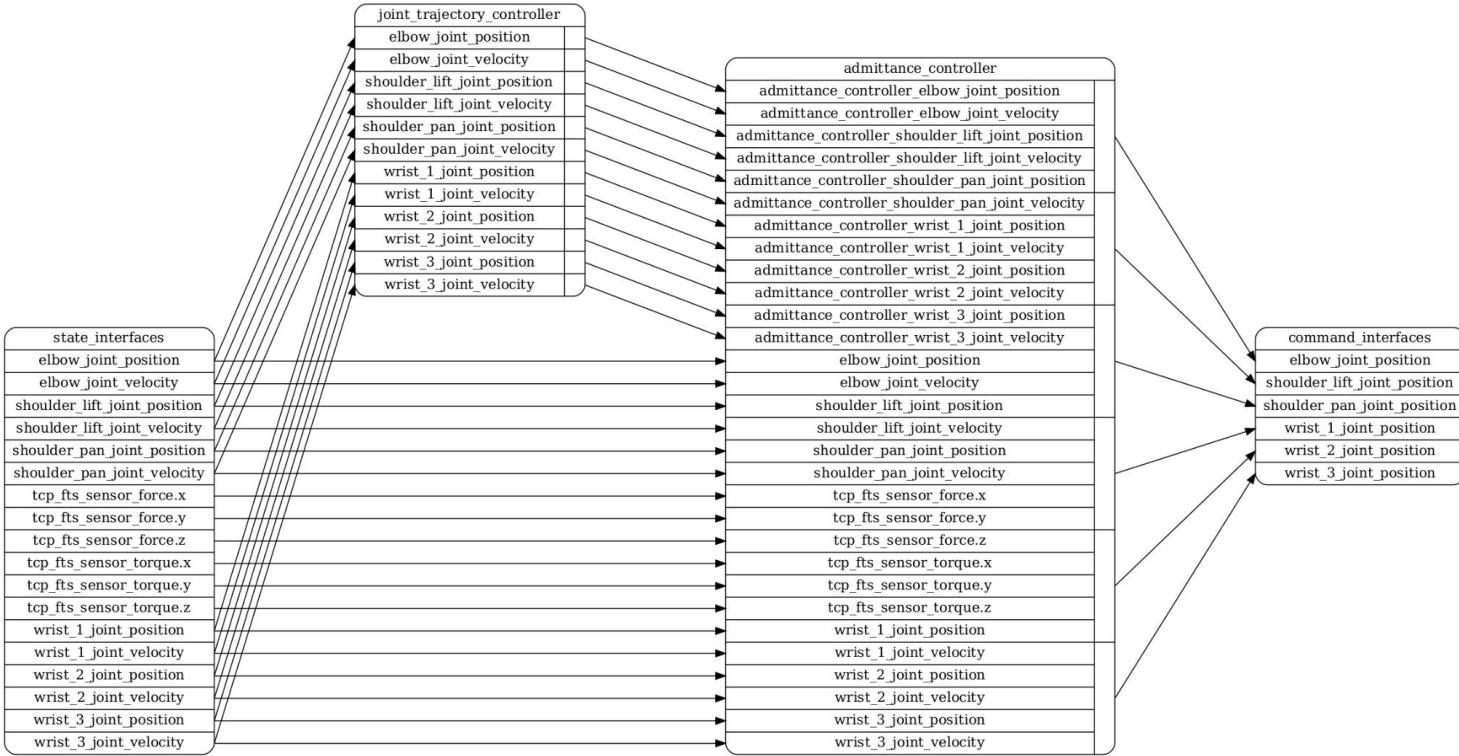
- Position
- ◆ Velocity
- Acceleration
- ▲ Force

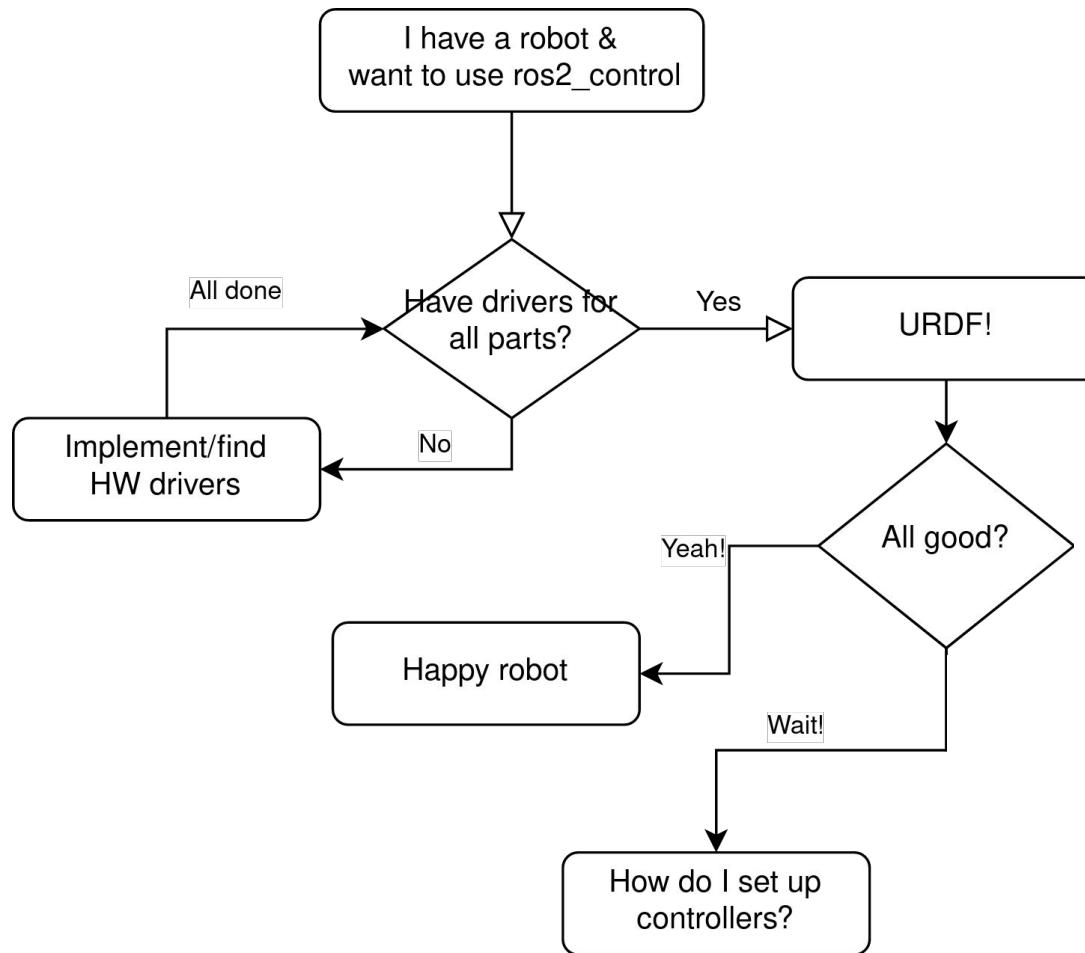
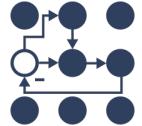


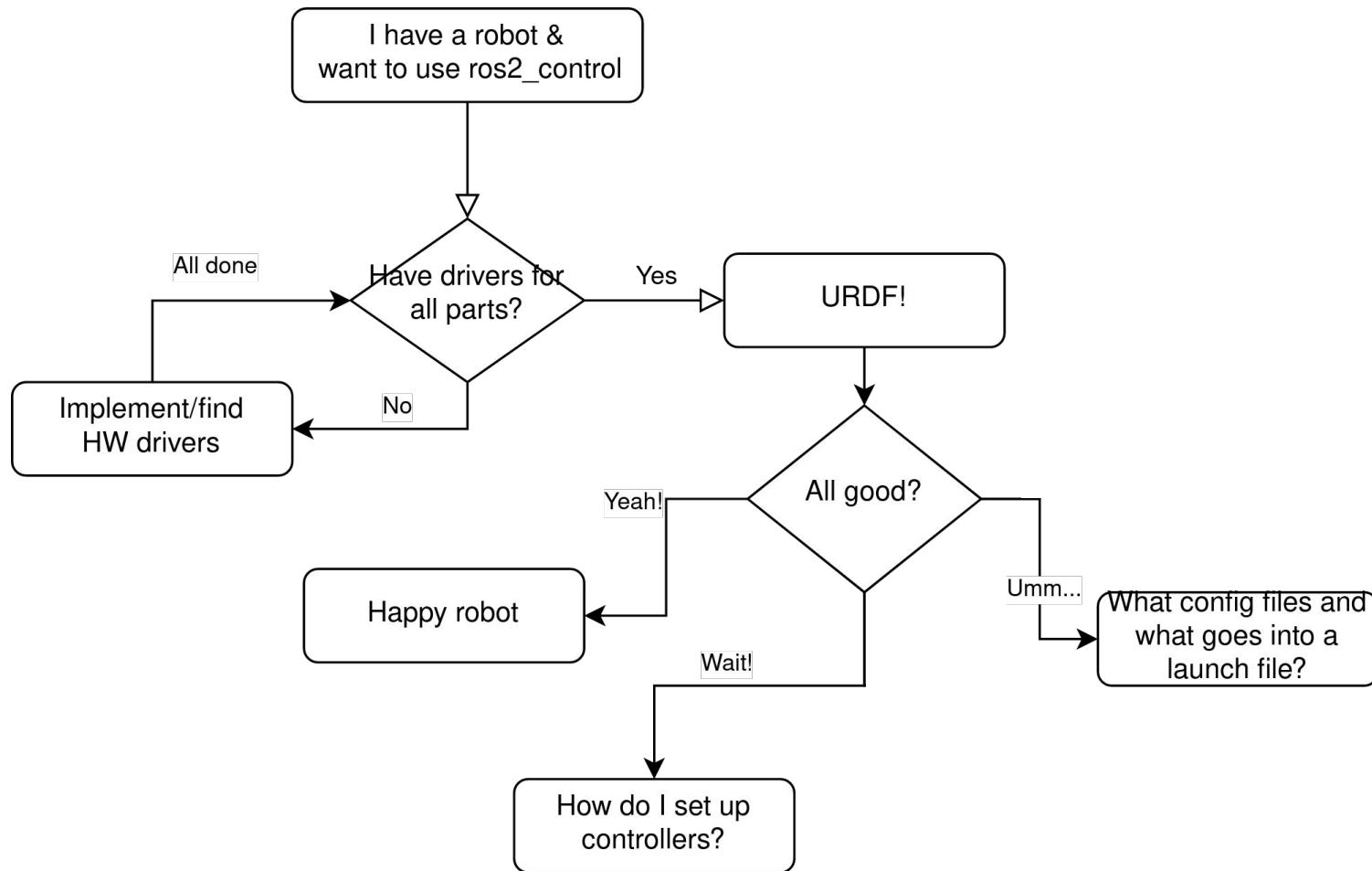
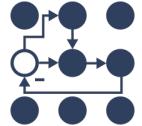
```
controller_manager:  
  update_rate: 500 # Hz  
  
joint_trajectory_controller:  
  type: joint_trajectory_controller/JointTrajectoryController  
  
admittance_controller:  
  type: admittance_controller/AdmittanceController  
  
limiting_controller:  
  type: limiting_controllers/JointLimitingController  
  
joint_trajectory_controller:  
  joints:  
    - joint1  
    - ...  
  command_joints:  
    - admittance_controller/joint1  
    - ...  
  command_interfaces:  
    - position  
  state_interfaces:  
    - position  
    - velocity  
  
# export reference interfaces: "<controller_name>/<joint_name>/<interface_name>"  
admittance_controller:  
  joints:  
    - joint1  
    - ...  
  command_joints:  
    - limiting_controller/joint1  
    - ...  
  command_interfaces:  
    - position  
  state_interfaces:  
    - position  
    - velocity  
  
# export reference interfaces: "<controller_name>/<joint_name>/<interface_name>"  
limiting_controller:  
  joints:  
    - joint1  
    - ...  
  interfaces:  
    - position  
limiting_controller:  
  joints:  
    - joint1  
    - ...  
  interfaces:  
    - position
```

# CLI extra

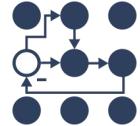
```
$ ros2 control view_controller_chains
```



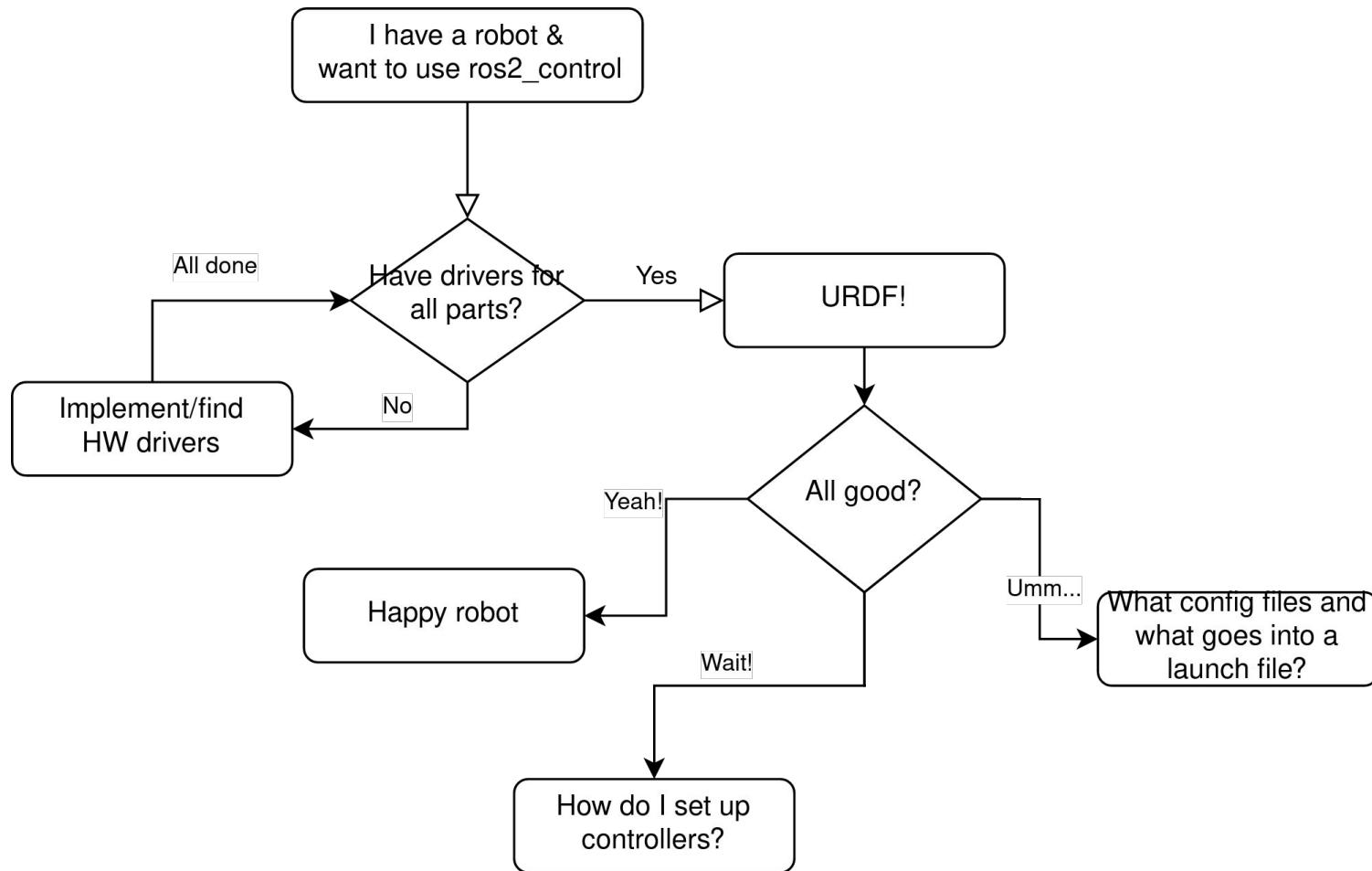
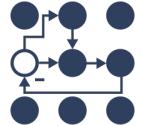


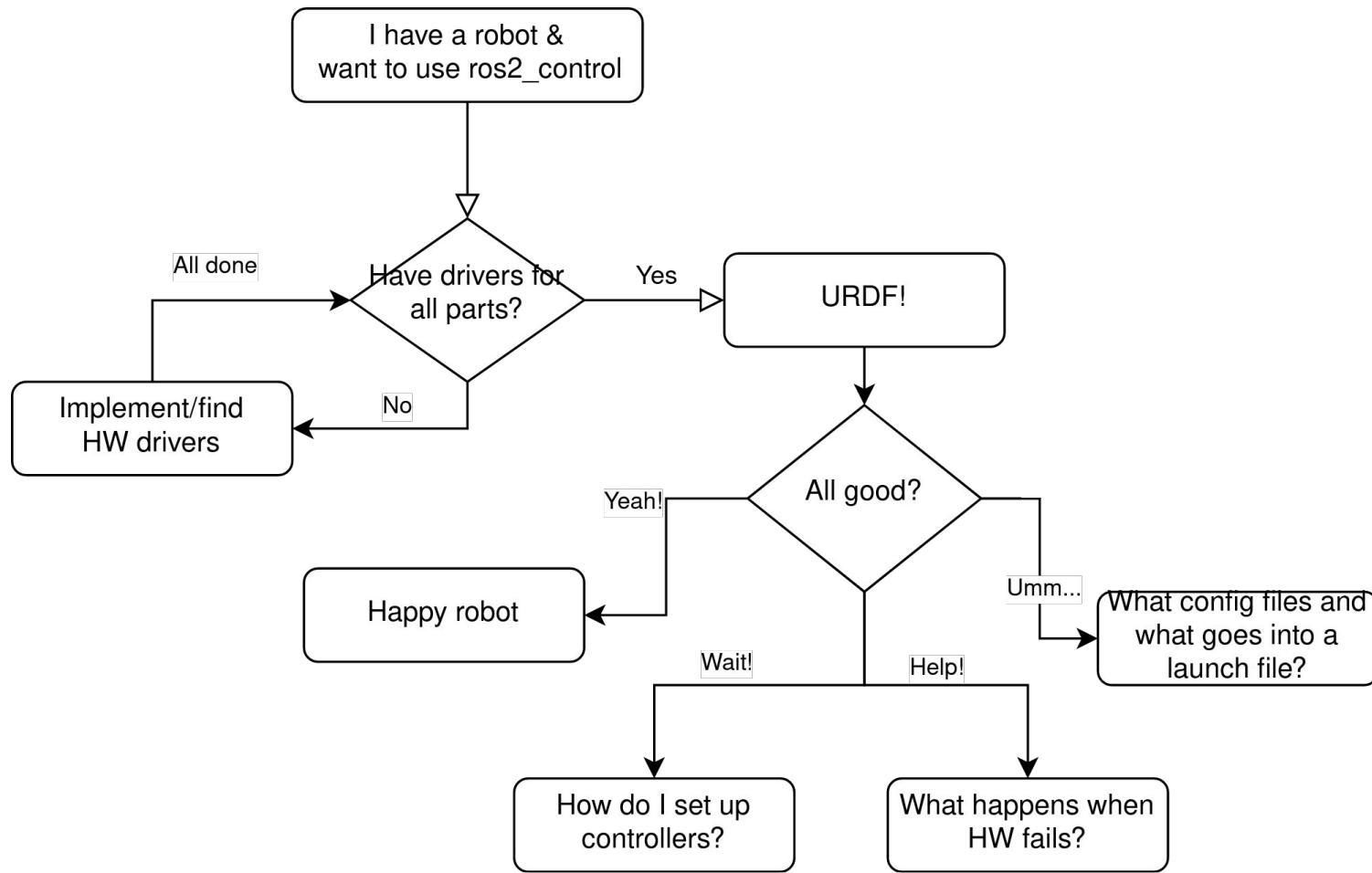
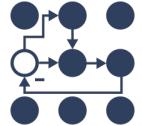


# What config files and where?

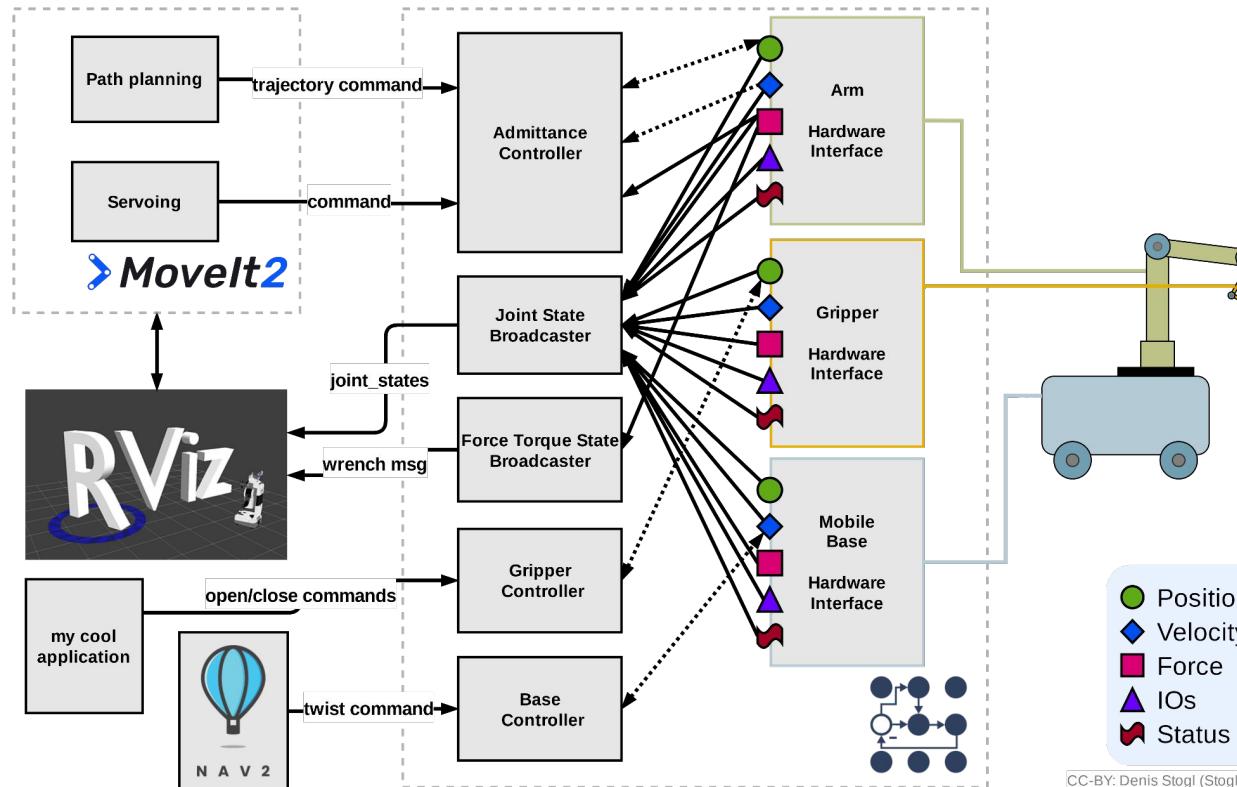


```
controller_manager:  
ros_parameters:  
update_rate: 10 # Hz  
  
joint_state_broadcaster:  
type: joint_state_broadcaster/JointStateBroadcaster  
  
position_trajectory_controller:  
type: joint_trajectory_controller/JointTrajectoryController  
  
position_trajectory_controller:  
ros_parameters:  
joints:  
- joint1  
- joint2  
  
command_interfaces:  
- position  
  
state_interfaces:  
- position  
  
state_publish_rate: 200.0 # Defaults to 50  
action_monitor_rate: 20.0 # Defaults to 20  
  
allow_partial_joints_goal: false # Defaults to false  
open_loop_control: true  
allow_integration_in_goal_trajectories: true  
constraints:  
    stopped_velocity_tolerance: 0.01 # Defaults to 0.01  
    goal_time: 0.0 # Defaults to 0.0 (start immediately)  
  
<?xml version="1.0"?>  
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">  
  
    <xacro:macro name="rrbot_ros2_control" params="name prefix">  
  
        <ros2_control name="${name}" type="system">  
            <hardware>  
                <plugin>ros2_control_demo_hardware/RRBotSystemPositionOnlyHardware</plugin>  
                <param name="example_param_hw_start_duration_sec">0</param>  
                <param name="example_param_hw_stop_duration_sec">3.0</param>  
                <param name="example_param_hw_slowdown">100</param>  
            </hardware>  
  
            <joint name="${prefix}joint1">  
                <command_interface name="position">  
                    <param name="min">-1</param>  
                    <param name="max">1</param>  
                </command_interface>  
                <state_interface name="position"/>  
            </joint>  
            <joint name="${prefix}joint2">  
                <command_interface name="position">  
                    <param name="min">-1</param>  
                    <param name="max">1</param>  
                </command_interface>  
                <state_interface name="position"/>  
            </joint>  
        </ros2_control>  
  
    </xacro:macro>  
  
</robot>  
  
control_node = Node(  
    package="controller_manager",  
    executable="ros2_control_node",  
    parameters=[robot_description, robot_controllers],  
    remappings=[  
        (/  
            "forward_position_controller/commands",  
            "/position_commands",  
        ),  
    ],  
    output="both",  
)  
  
robot_state_pub_node = Node(  
    package="robot_state_publisher",  
    executable="robot_state_publisher",  
    output="both",  
    parameters=[robot_description],  
)  
  
joint_state_broadcaster_spawner = Node(  
    package="controller_manager",  
    executable="spawner",  
    arguments=["joint_state_broadcaster", "--controller-manager", "/controller_manager"],  
)  
  
robot_controller_spawner = Node(  
    package="controller_manager",  
    executable="spawner",  
    arguments=[ "forward_position_controller", "-c", "/controller_manager"],  
)  
  
nodes = [  
    control_node,  
    robot_state_pub_node,  
    joint_state_broadcaster_spawner,  
    robot_controller_spawner,  
]  
  
return LaunchDescription(nodes)
```

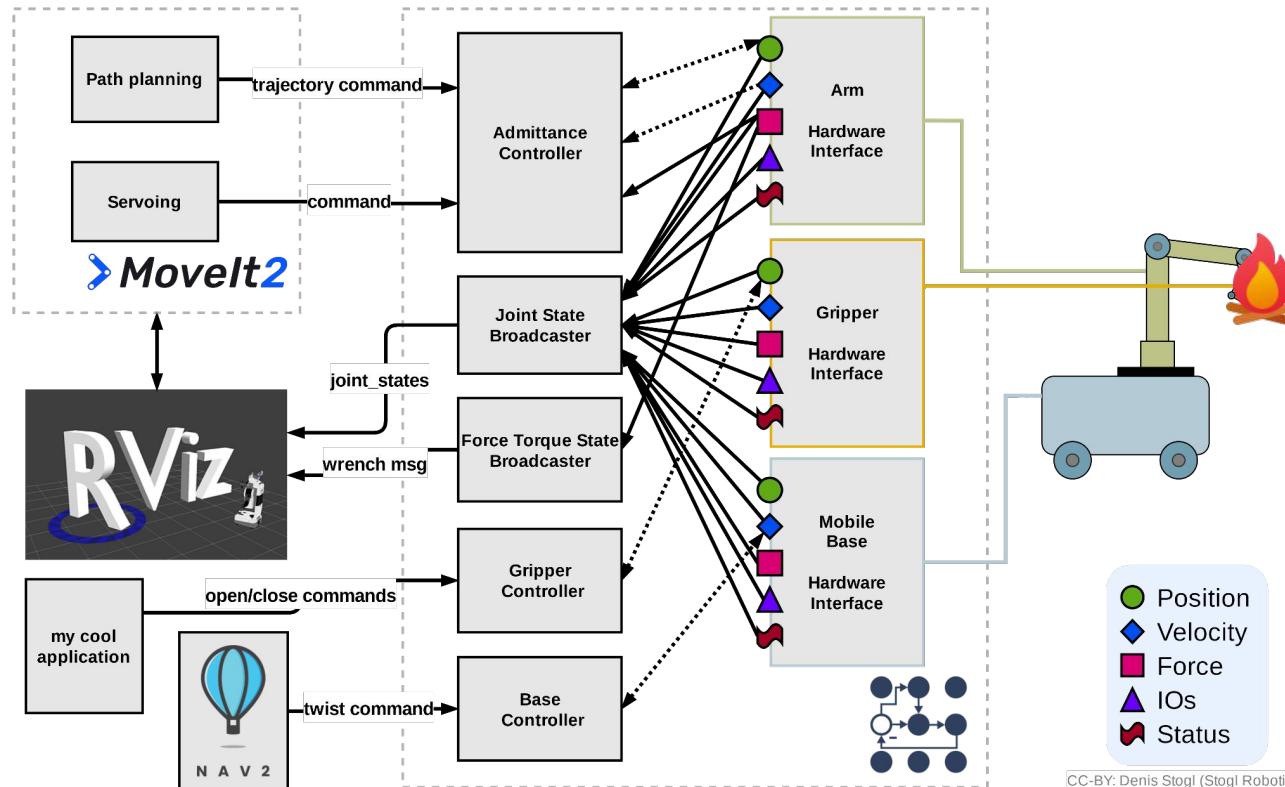




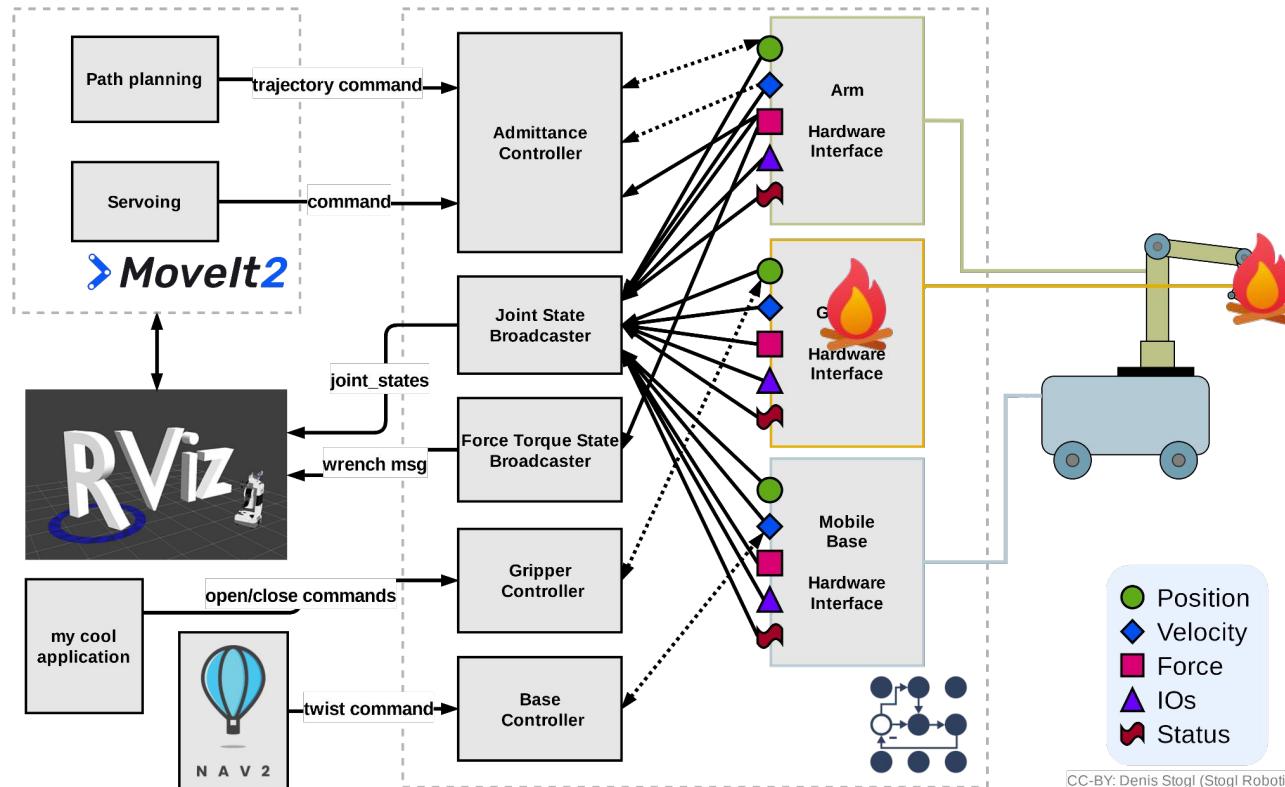
# HW error handling



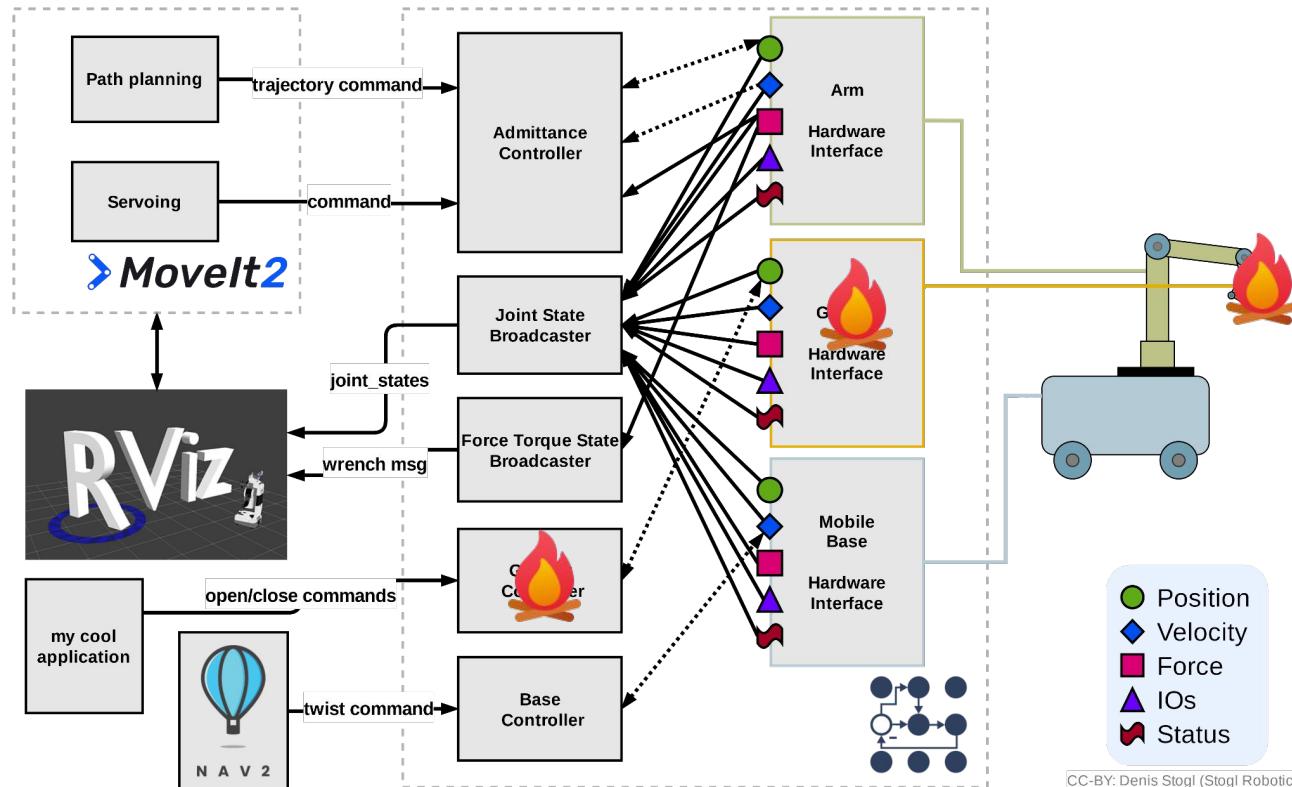
# HW error handling



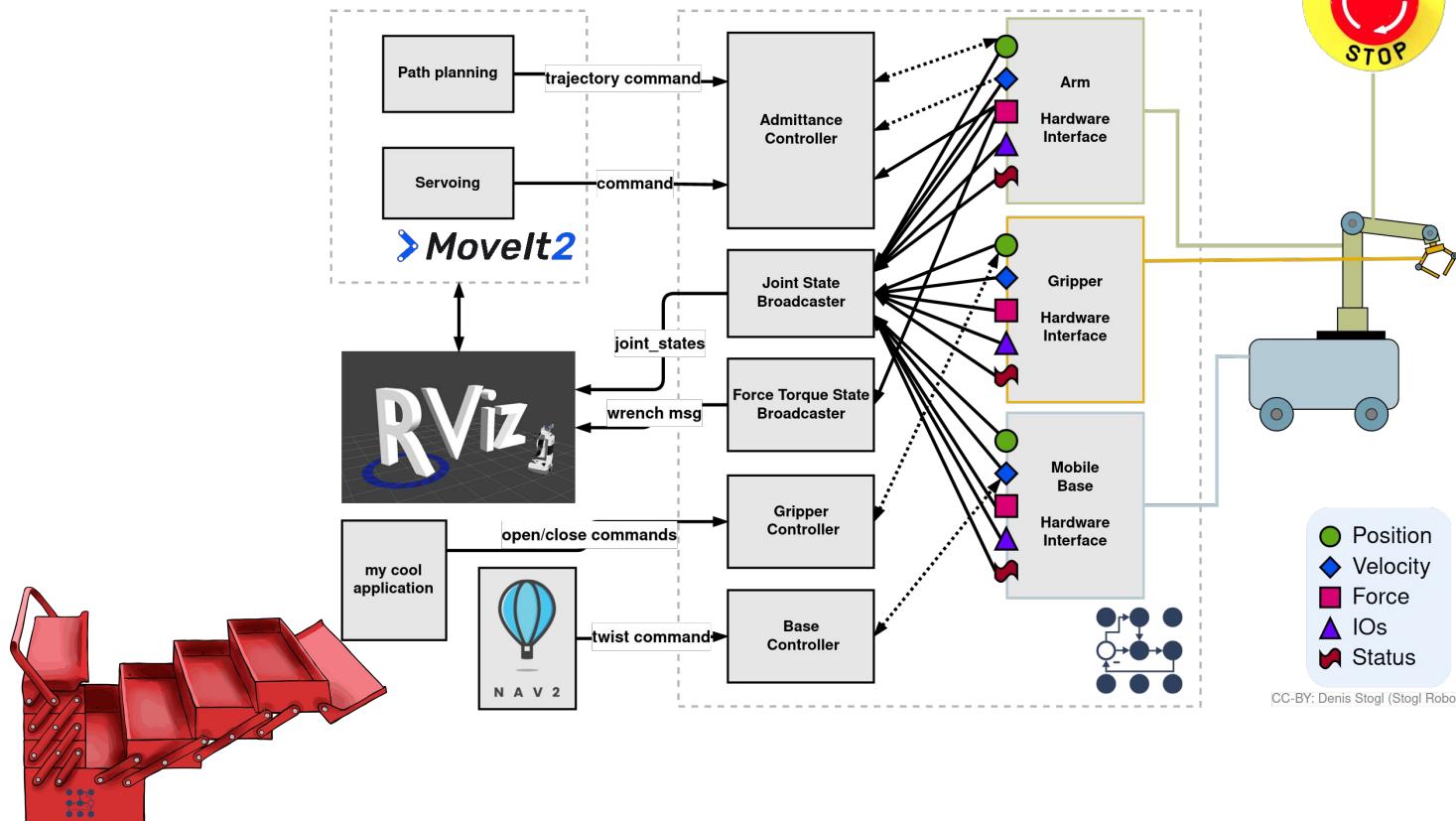
# HW error handling



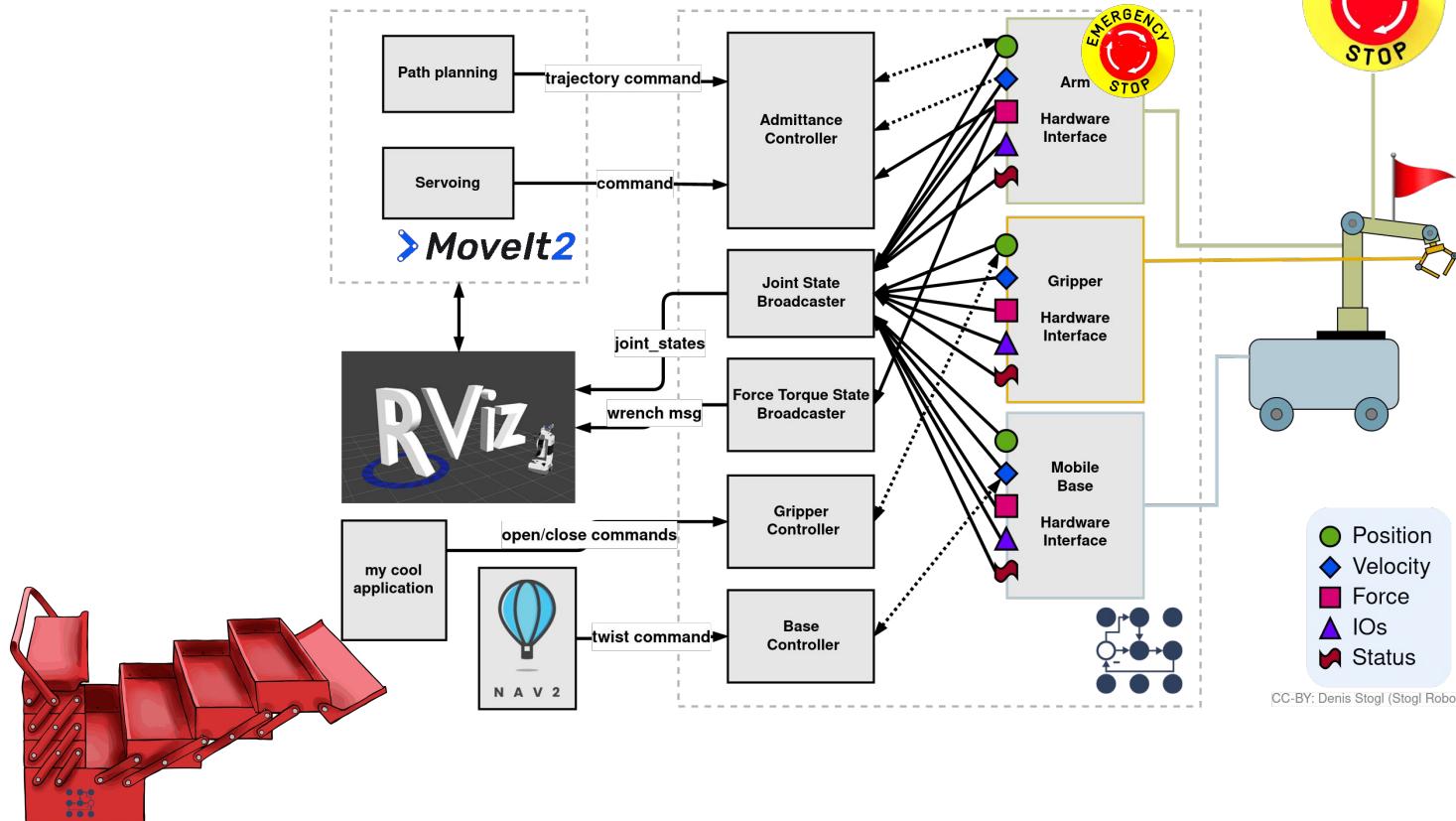
# HW error handling



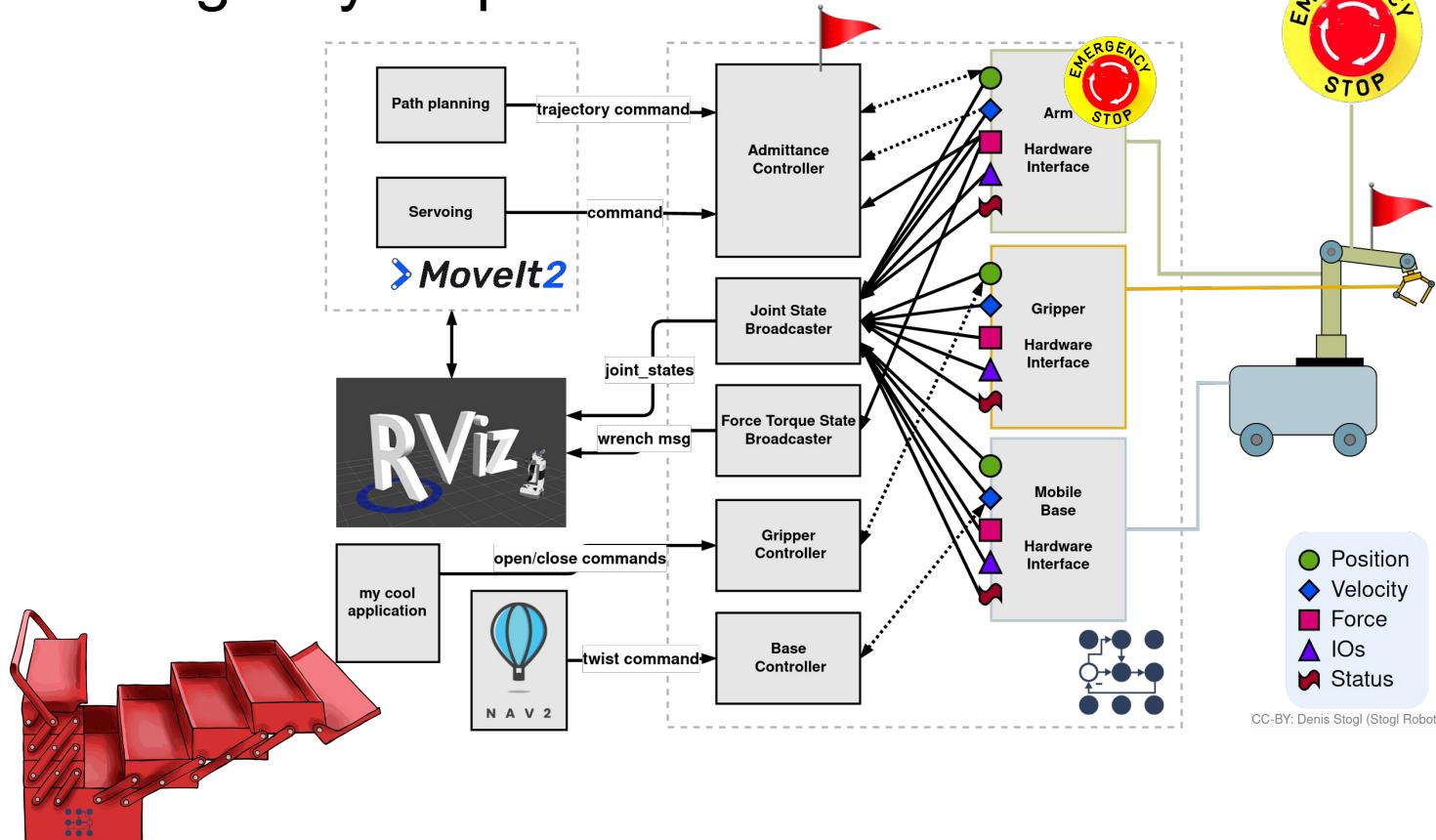
# Emergency stops\*



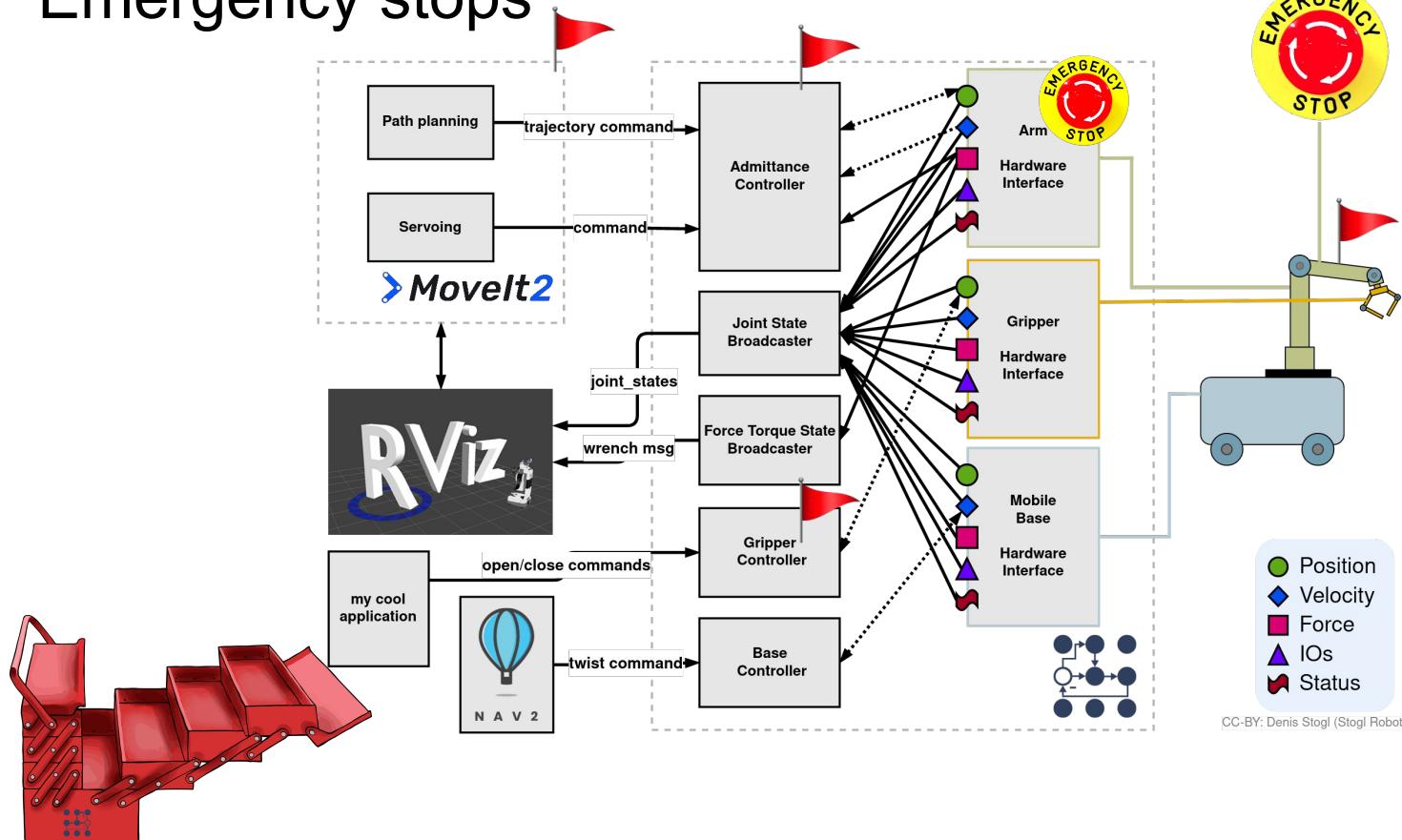
# Emergency stops\*

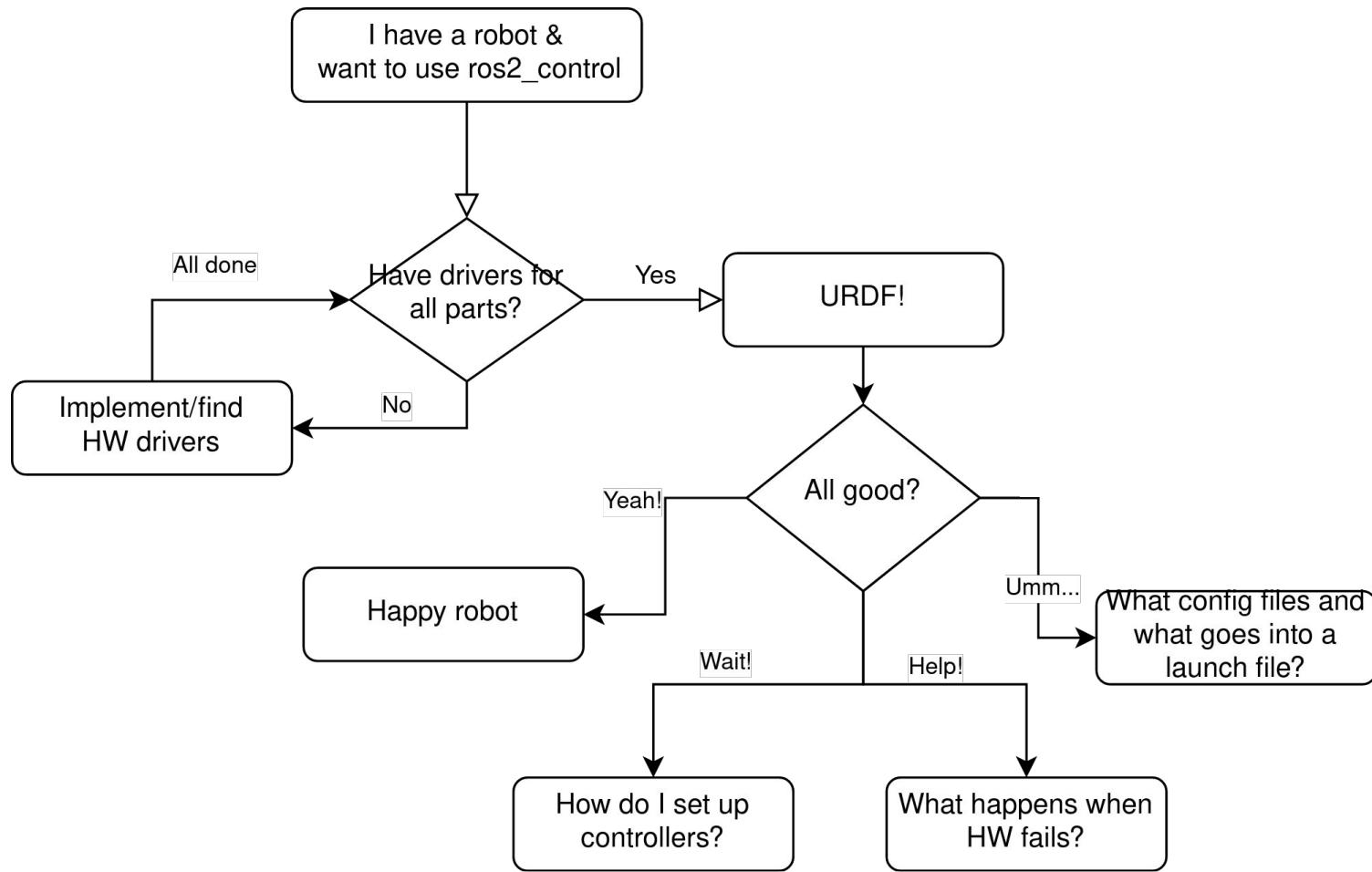
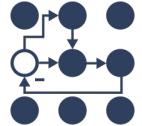


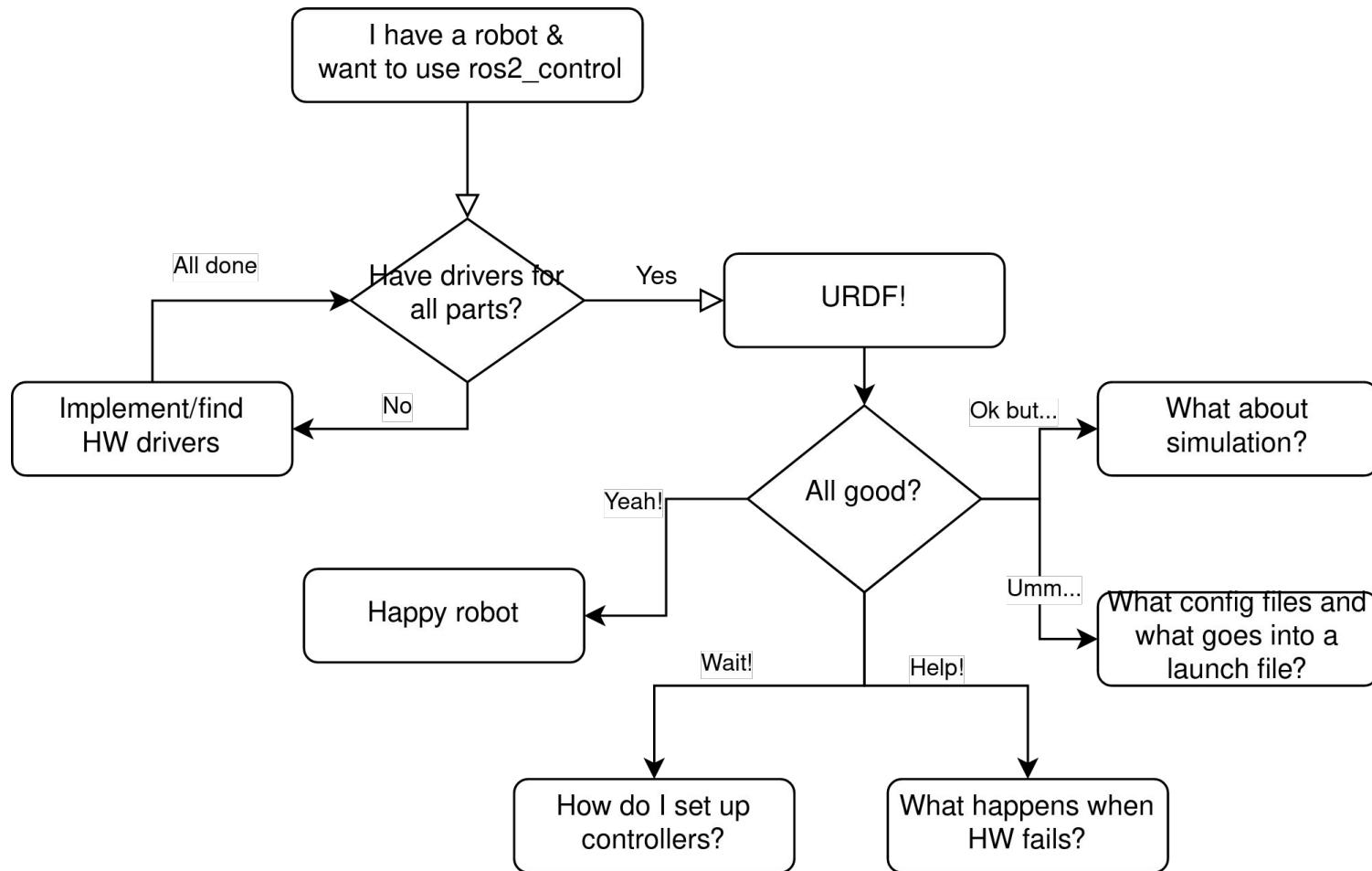
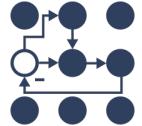
# Emergency stops\*



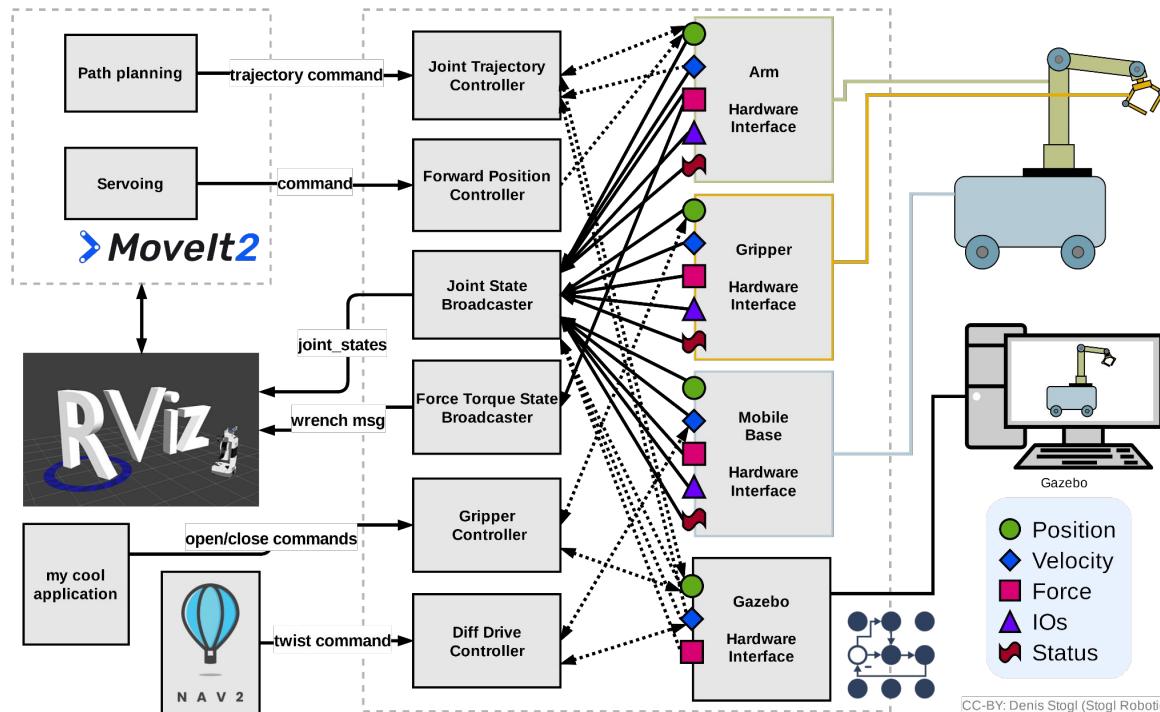
# Emergency stops\*







# Let's check an example



```
<ros2_control name="rrbot_real" type="system">
```

```
<hardware>
```

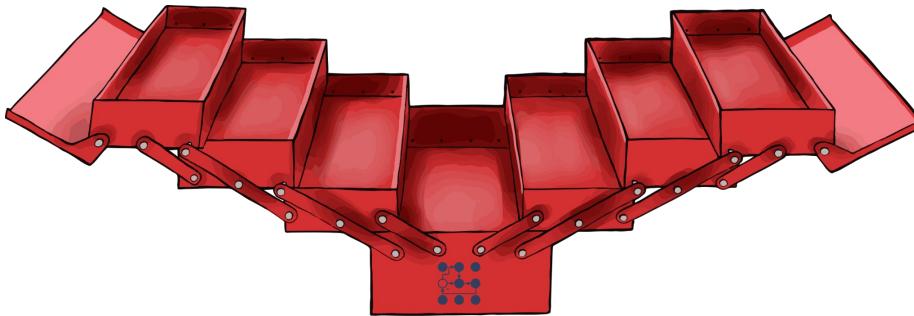
```
  <plugin>ros2_control_demo_hardware/RRBotSystemPositionOnlyHardware</plugin>
  <param name="hw_start_duration_sec">0.0</param>
  <param name="hw_stop_duration_sec">3.0</param>
  <param name="hw_slowdown_factor">2.0</param>
```

```
<hardware>
```

```
<joint name="joint1">
  <command_interface name="position">
    <param name="min">-1</param>
    <param name="max">1</param>
  </command_interface>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
</joint>
<joint name="joint2">
  <command_interface name="position"/>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
</joint>
```

```
</ros2_control>
```

Real hardware



```
<ros2_control name="rrbot_sim" type="system">
```

```
<hardware>
```

```
  <plugin>gazebo_ros2_control/GazeboSystem</plugin>
</hardware>
```

```
<joint name="joint1">
```

```
  <command_interface name="position">
    <param name="min">-1</param>
    <param name="max">1</param>
  </command_interface>
```

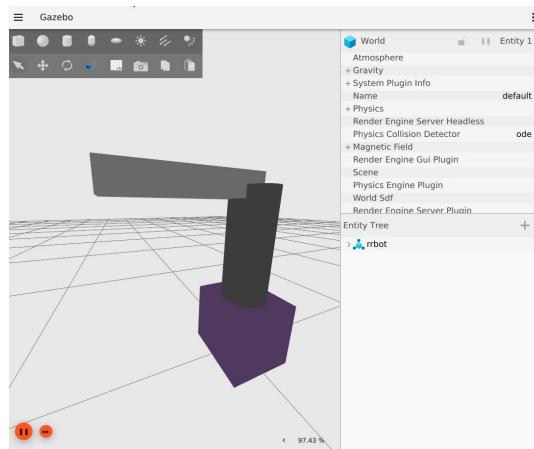
```
  <command_interface name="velocity">
    <param name="min">-1</param>
    <param name="max">1</param>
  </command_interface>
```

```
  <command_interface name="acceleration">
    <param name="min">-1</param>
    <param name="max">1</param>
  </command_interface>
```

```
  <state_interface name="position"/>
  <state_interface name="velocity"/>
  <state_interface name="acceleration"/>
</joint>
```

```
<joint name="joint2">
  <command_interface name="position"/>
  <command_interface name="velocity"/>
  <command_interface name="acceleration"/>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
  <state_interface name="acceleration"/>
</joint>
```

```
</ros2_control>
```



Gazebo  
simulation

```
<ros2_control name="rrbot_real" type="system">
```

```
<hardware>
```

```
  <plugin>ros2_control_demo_hardware/RRBotSystemPositionOnlyHardware</plugin>
  <param name="hw_start_duration_sec">0.0</param>
  <param name="hw_stop_duration_sec">3.0</param>
  <param name="hw_slowdown_factor">2.0</param>
```

```
<hardware>
```

```
<joint name="joint1">
```

```
  <command_interface name="position">
    <param name="min">-1</param>
    <param name="max">1</param>
  </command_interface>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
```

```
</joint>
```

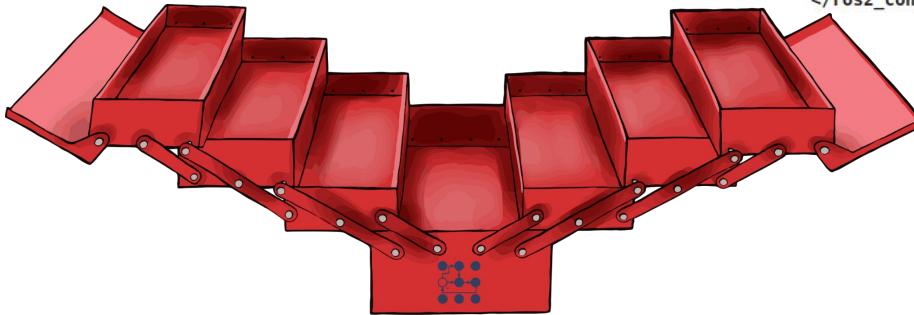
```
<joint name="joint2">
```

```
  <command_interface name="position"/>
  <state_interface name="position"/>
  <state_interface name="velocity"/>
```

```
</joint>
```

```
</ros2_control>
```

Real hardware



```
<ros2_control name="rrbot_mock" type="system">
```

```
<hardware>
```

```
  <plugin>mock_components/GenericSystem</plugin>
  <param name="fake_sensor_commands">True</param>
  <param name="state_following_offset">0.0</param>
```

```
</hardware>
```

```
<joint name="joint1">
```

```
  <command_interface name="position">
    <param name="min">-1</param>
    <param name="max">1</param>
```

```
  </command_interface>
```

```
  <command_interface name="velocity">
    <param name="min">-1</param>
    <param name="max">1</param>
```

```
  </command_interface>
```

```
  <command_interface name="acceleration">
    <param name="min">-1</param>
    <param name="max">1</param>
```

```
  </command_interface>
```

```
  <state_interface name="position"/>
  <state_interface name="velocity"/>
```

```
  <state_interface name="acceleration"/>
```

```
</joint>
```

```
<joint name="joint2">
```

```
  <command_interface name="position"/>
  <command_interface name="velocity"/>
  <command_interface name="acceleration"/>
```

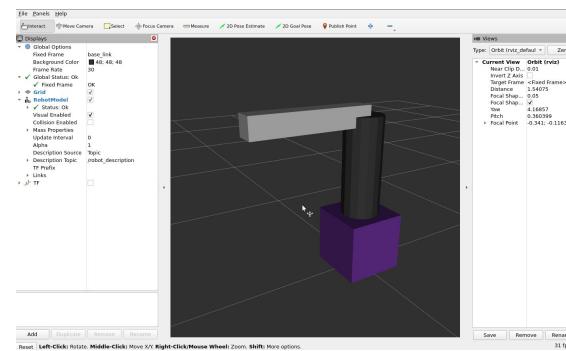
```
  <state_interface name="position"/>
```

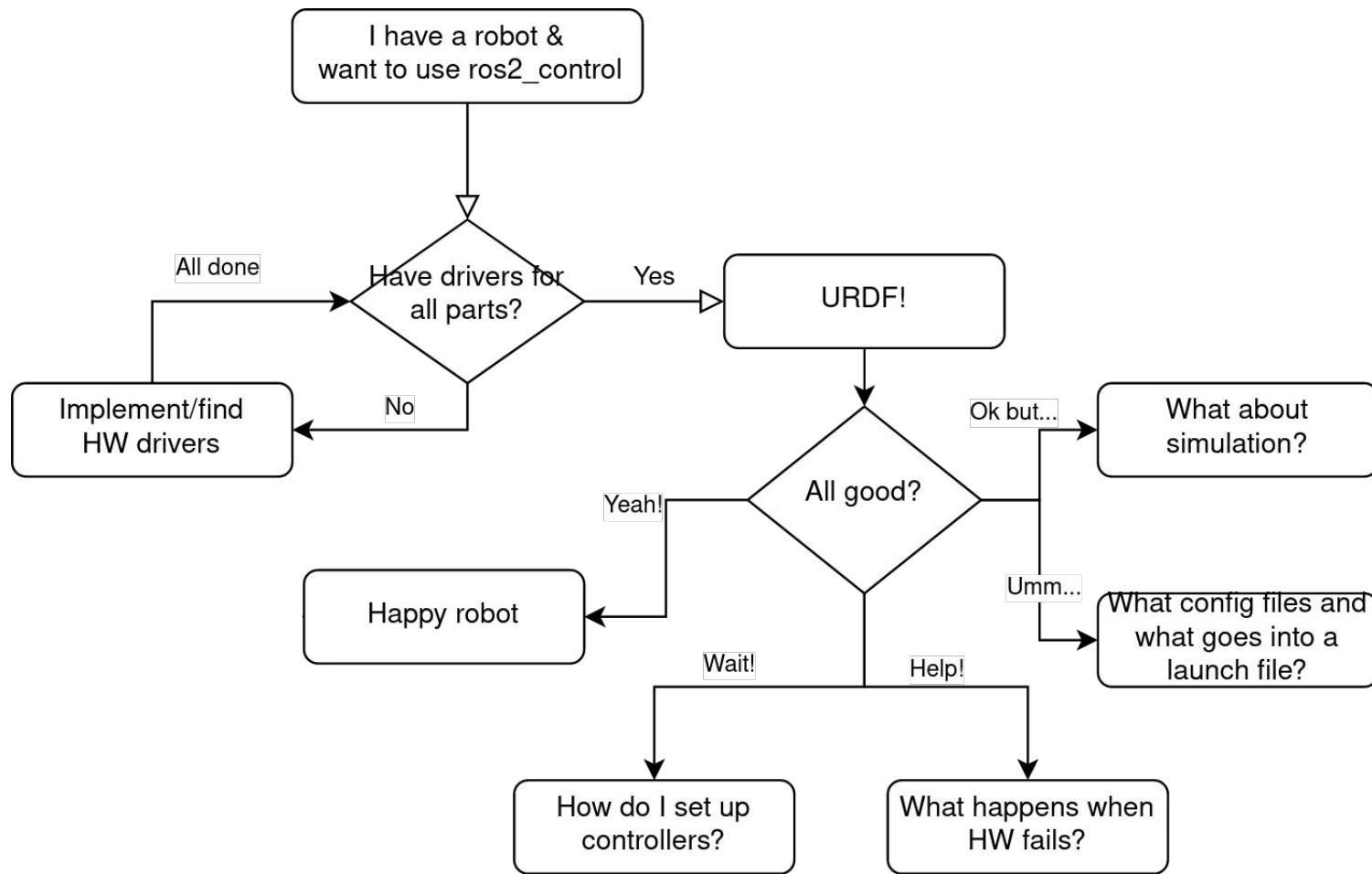
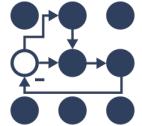
```
  <state_interface name="velocity"/>
  <state_interface name="acceleration"/>
```

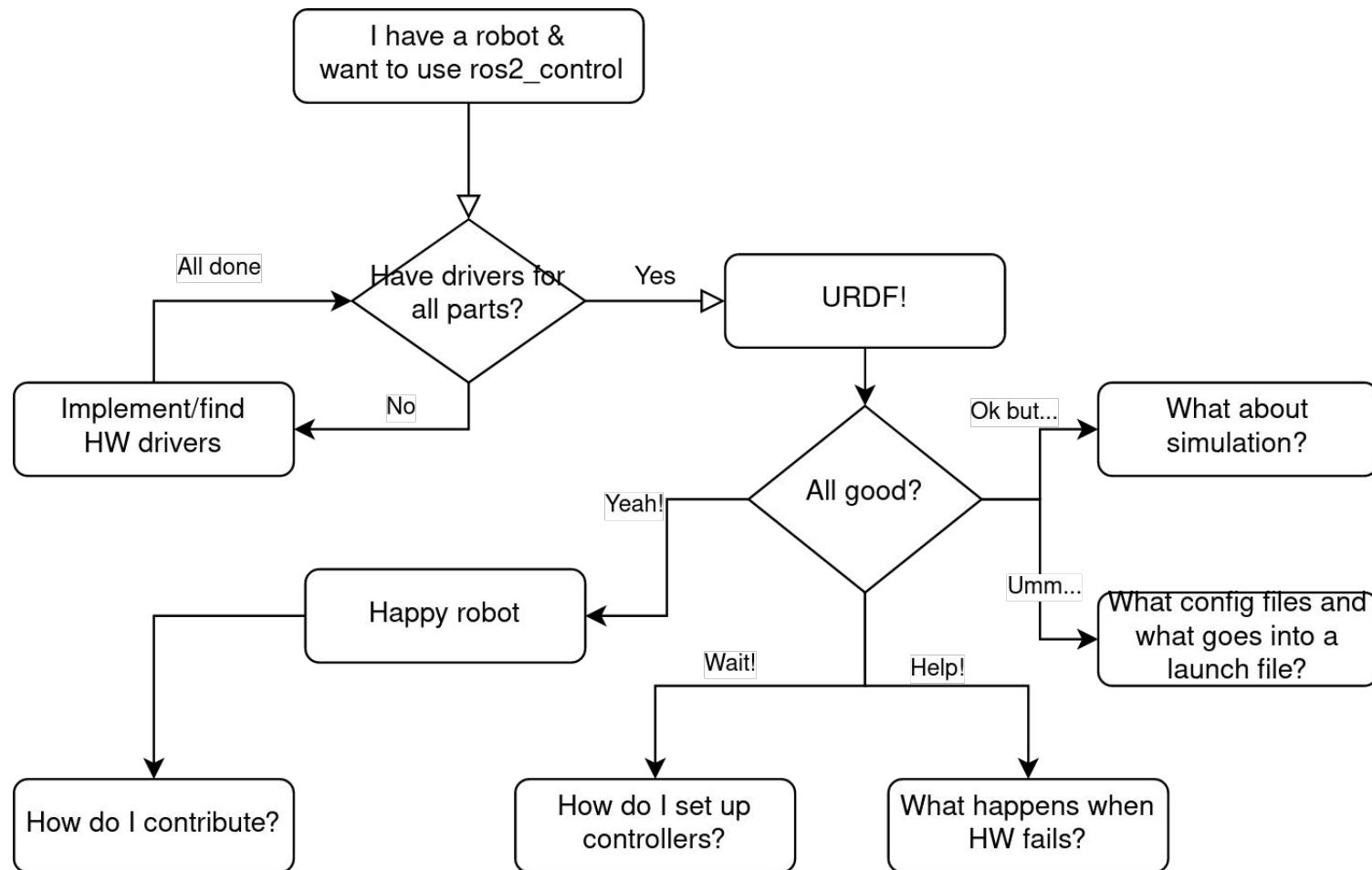
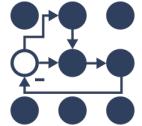
```
</joint>
```

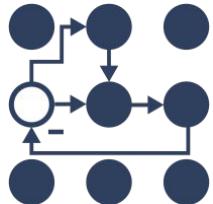
```
</ros2_control>
```

Mock









# Contributing

<https://github.com/ros-controls>

[ros-controls / ros2\\_control](#) Public

[Code](#)  [Issues 97](#)  [Pull requests 23](#)

---

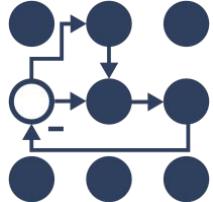
[Add additional return value to the hardware\\_interface::return\\_type](#) good first issue good second issue help wanted

#815 opened 27 days ago by destogl

[ros2\\_control reviewers](#)



25 members



# References

- <https://control.ros.org>
  - ros\_control [paper](#) in the Journal of Open Source Software
  - ros2\_control presentations
    - <https://control.ros.org/master/doc/resources/resources.html>
  - ros2\_control resources
    - <https://ros-controls.github.io/control.ros.org/>
    - [https://github.com/ros-controls/ros2\\_control](https://github.com/ros-controls/ros2_control)
    - [https://github.com/ros-controls/ros2\\_controllers](https://github.com/ros-controls/ros2_controllers)
    - [https://github.com/ros-controls/ros2\\_control\\_demos](https://github.com/ros-controls/ros2_control_demos)
    - [https://github.com/ros-controls/roadmap/blob/master/documentation\\_resources.md](https://github.com/ros-controls/roadmap/blob/master/documentation_resources.md)

# Thank you!



Denis Štogl, Karsten Knese,  
Victor Lopez, Jordan Palacios,  
Tyler Weaver, Márk Szitánics,  
Paul Gesel, Tony Najjar, Andy  
Zelenak, Olivier Stasse, Sachin  
Kumar, Noel Jiménez García,  
Jaron Lundwall, Alejandro  
Hernández Cordero, Colin  
MacKenzie, Tim Clephas,  
Lovro Ivanov, Jafar Abdi,  
Michael Wiznitzer, Patrick  
Roncagliolo, Bence Magyar  
and many more!