



# ros2\_control

## The future of ros\_control

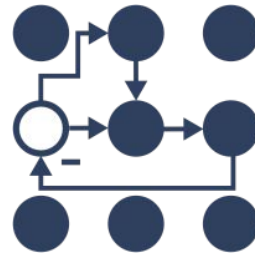




Image from [ros\\_control paper](#)

# What & where

pr2\_controller\_manager

(pr2\_mechanism)

2009



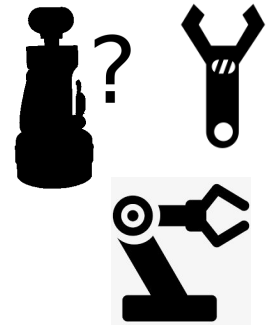
ros\_control

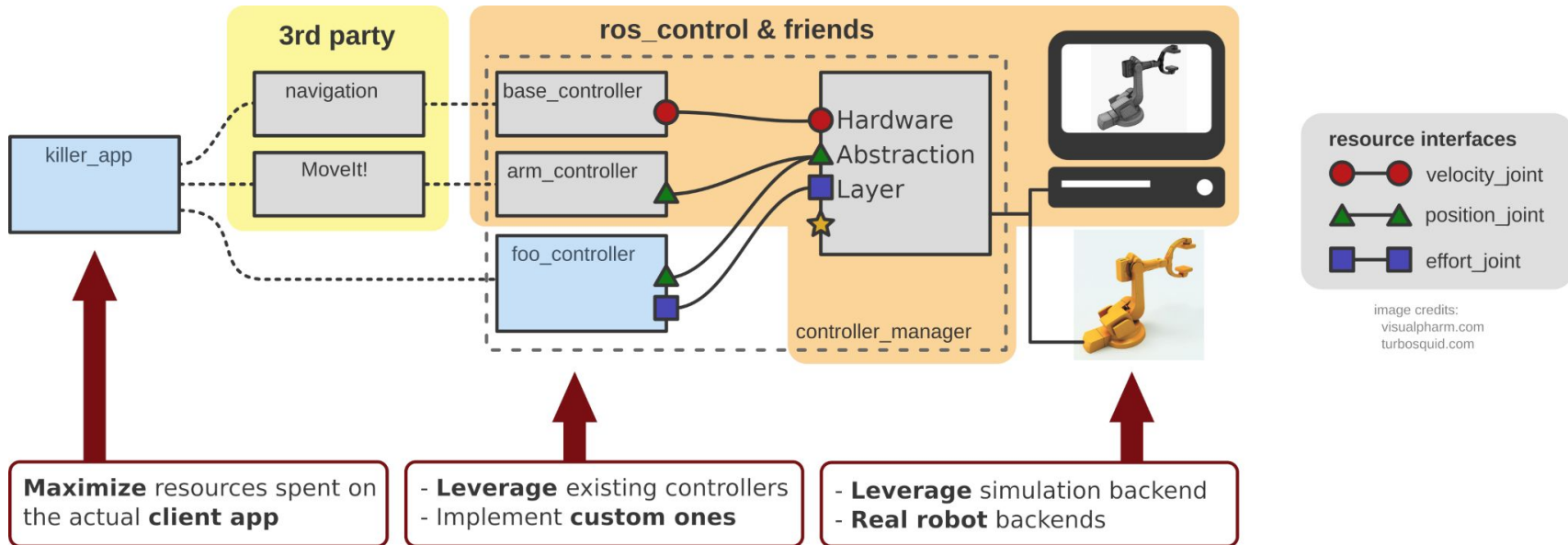
2012/2013



ros2\_control

2017/2021





# Mobile manipulator example

Head controller / gaze controller



End-effector controller



Arm controller / arm-torso controller



Mobile-base controller

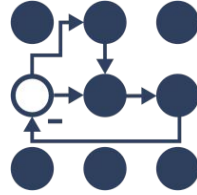


State  
broadcaster(s)

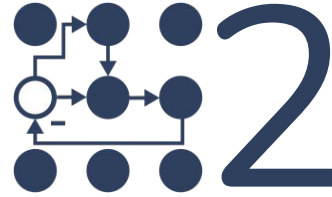
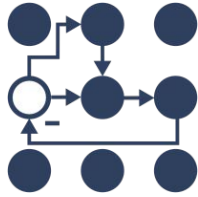


# Standard controllers

- joint\_state\_broadcaster
- diff\_drive\_controller
- joint\_trajectory\_controller
- gripper\_controllers
- Forwarding controllers for (groups of) joints
  - position\_controllers
  - velocity\_controllers
  - effort\_controllers



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



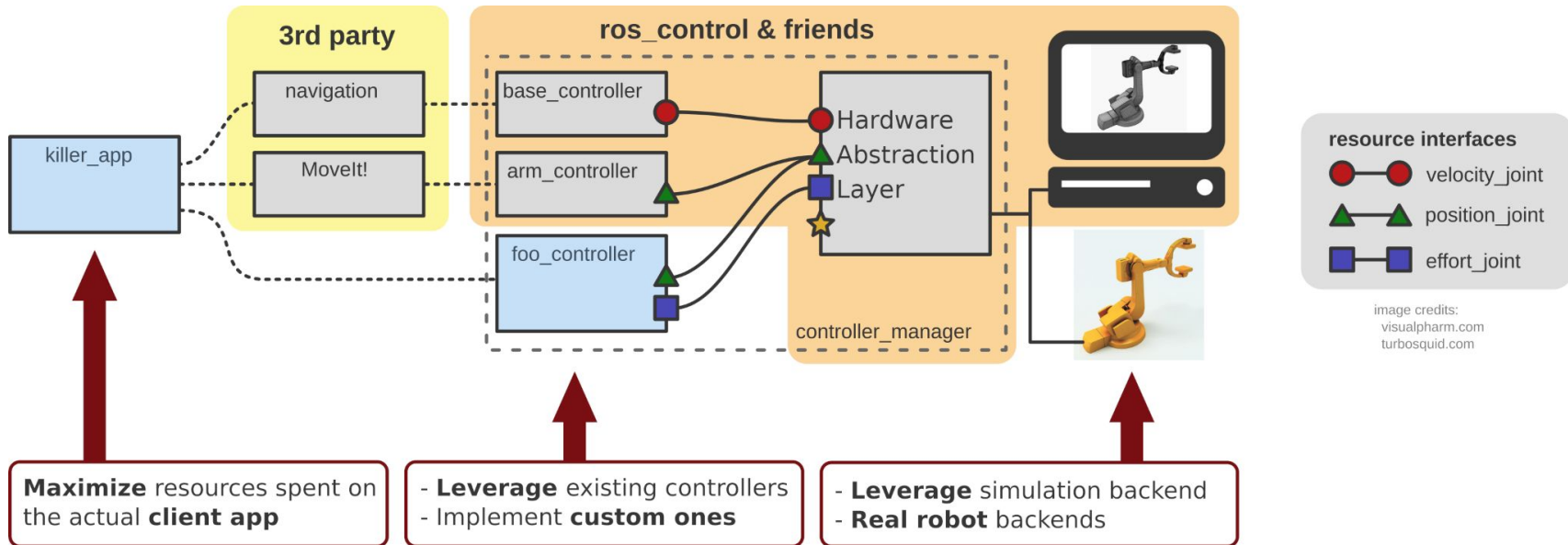
- General, robot-agnostic framework
- Supported joint interfaces: position, velocity, effort
- Collection of official controllers, defining de-facto standard ROS interfaces to 3rd party
- Off-the-shelf Gazebo integration
- Code complexity high, lots of templating and inheritance
- Controller lifecycle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

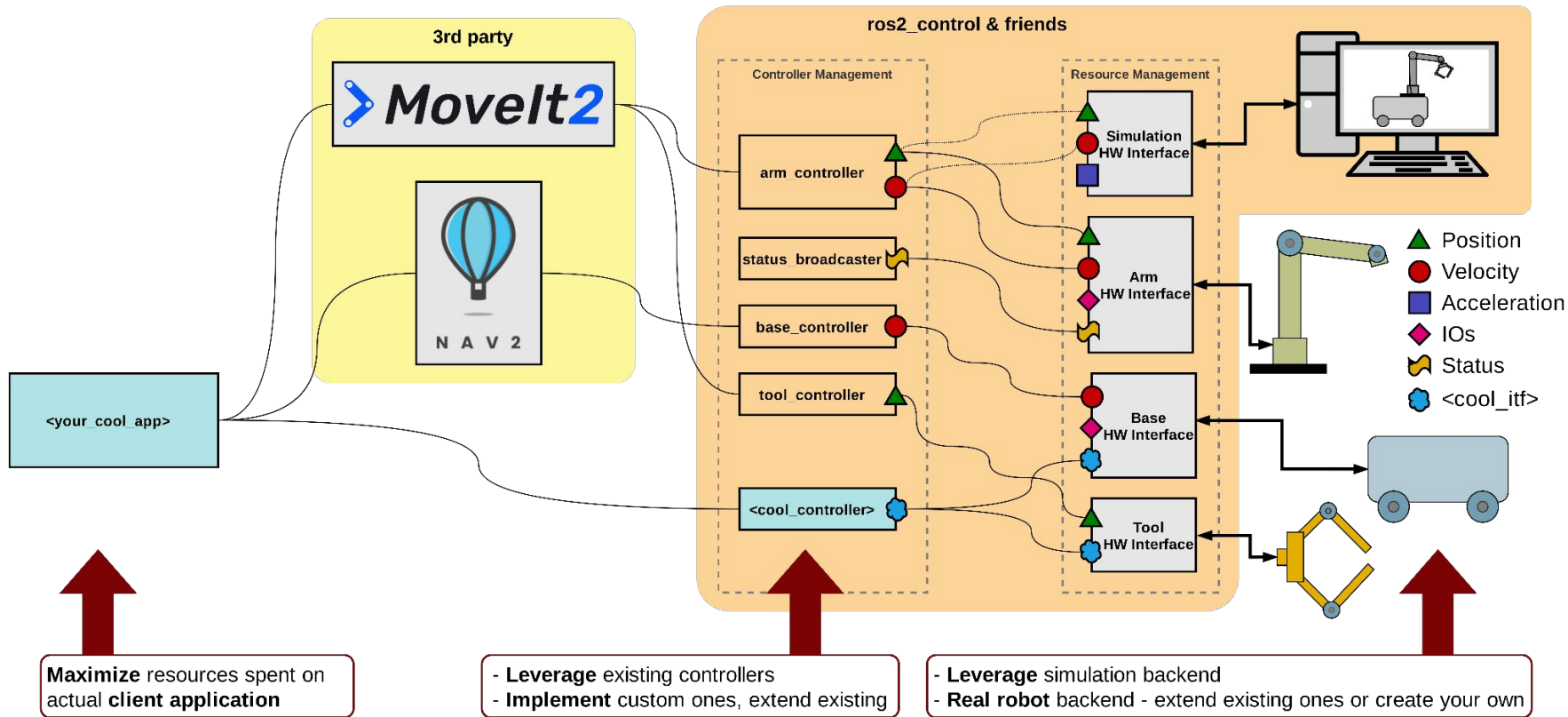
- Supported joint interfaces: no limitations
- 
- 
- Code leaner, more modern C++
- Controller lifecycle via ROS2 LifecycleNode
- [System|Actuator|Sensor]Component, Controller and Broadcaster
- Hardware Composition is first class citizen
- Default *ros2\_control\_node*
- Hardware lifecycle
- Synchronous but variable rate for controllers
- Stay tuned!



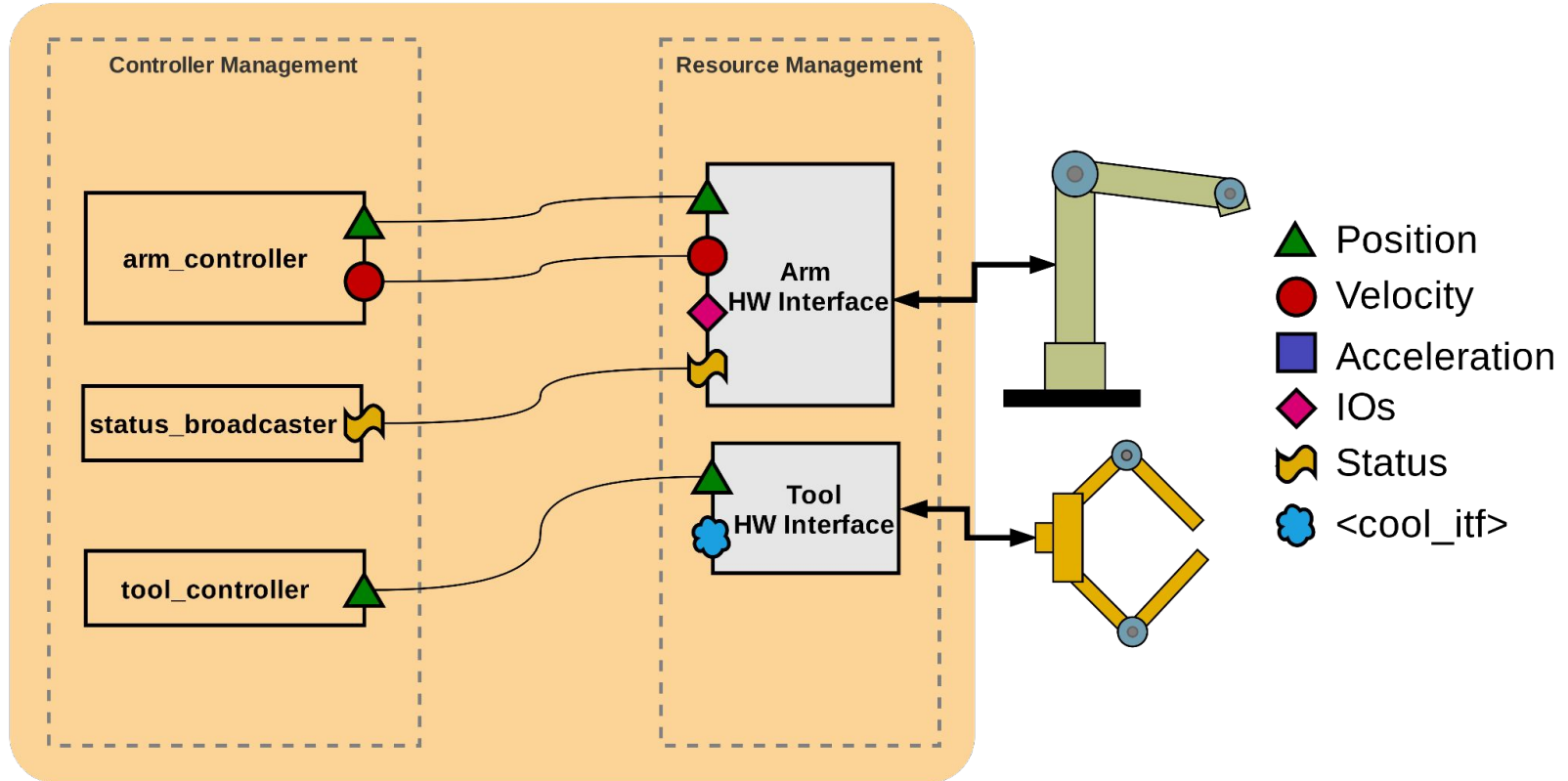


- General, robot-agnostic framework
- Supported joint interfaces: no limitations
- Collection of official controllers, defining de-facto standard ROS interfaces to 3rd party
- Off-the-shelf Gazebo integration
- Lean and extendable structure using modern C++
- Controller lifecycle via ROS2 LifecycleNode
- [System|Actuator|Sensor]Component, Controller and Broadcaster
- Full Hardware Composition support
- Default *ros2\_control\_node*
- Hardware lifecycle
- Synchronous but variable rate for controllers
- Asynchronous controllers
- Joint limiting plugin
- Emergency stop handler plugin





# Command and state interfaces



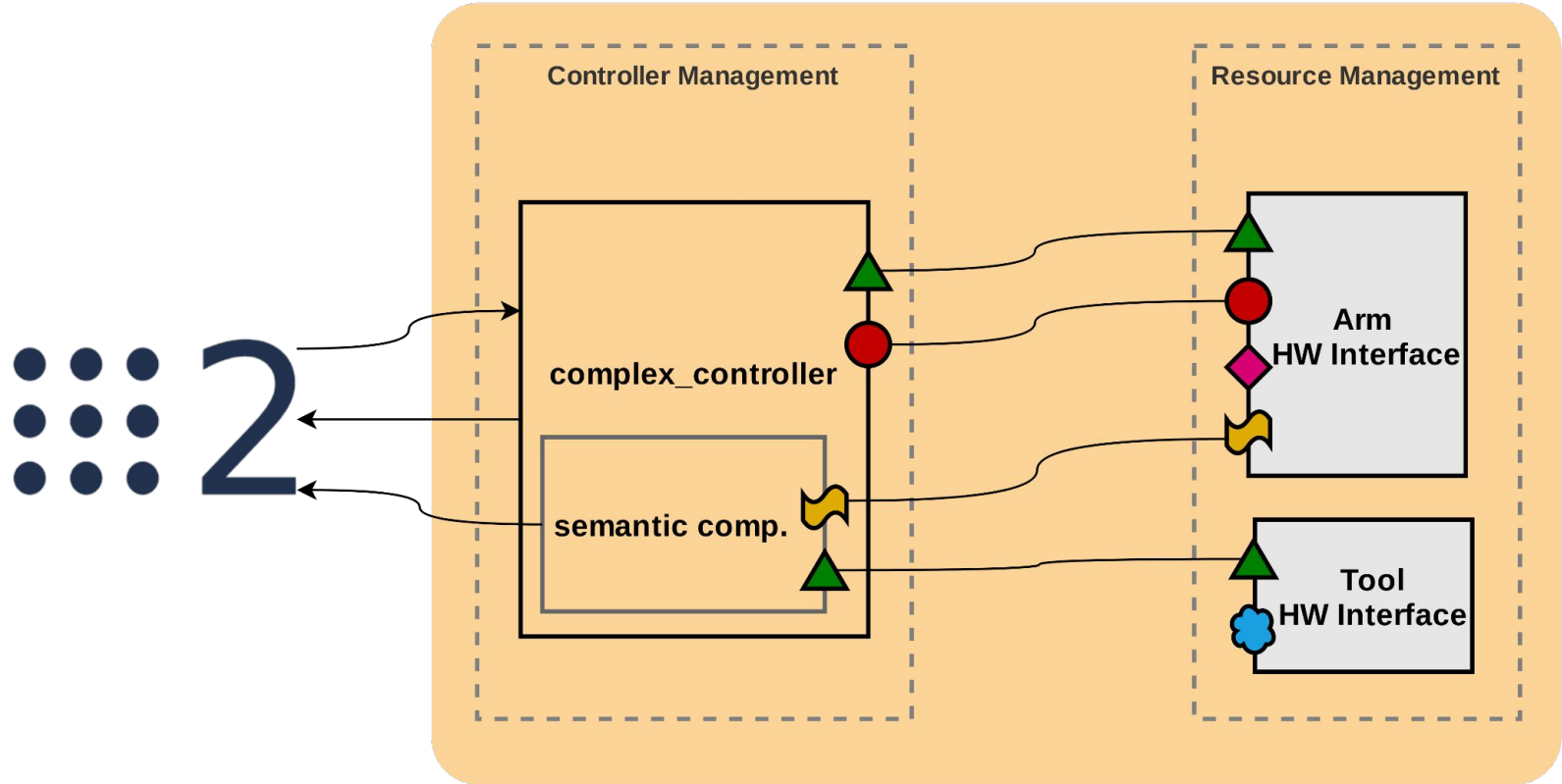
# URDF extension with <ros2\_control>-tag

```
<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>
  . . .
  <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>
```

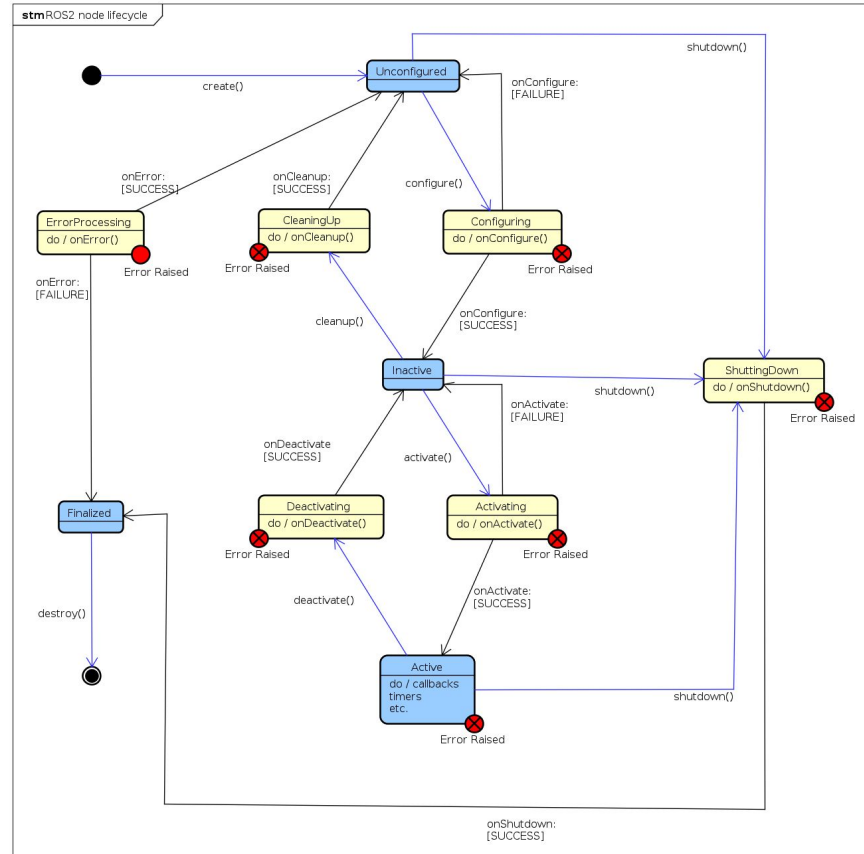
```
<ros2_control name="robot" type="system">
  <hardware>
    <plugin>robot_package/Robot</plugin>
    <param name="hardware_parameter">some_value</param>
  </hardware>
  <joint name="joint_first">
    <command_interface name="position"/>
    <state_interface name="acceleration"/>
  </joint>
  . . .
  <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_inteface"/>
    <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>
```

# Syntax cookies: semantic components



# Lifecycle for controllers and hardware

- Managed nodes - interface
  - Hardware components
  - Controllers



[https://design.ros2.org/articles/node\\_lifecycle.html](https://design.ros2.org/articles/node_lifecycle.html)

# ros2\_control CLI

Integrated with ROS2 CLI

```
$ ros2 control list_controllers  
base_controller[DiffDriveController] active
```

```
$ ros2 control list_hardware_interfaces  
command interfaces  
  joint1/position [unclaimed]  
  joint2/position [unclaimed]  
state interfaces  
  joint1/position  
  joint2/position
```

```
$ ros2 control list_controller_types  
diff_drive_controller/DiffDriveController  
joint_state_controller/JointStateController  
joint_trajectory_controller/JointTrajectoryController  
controller_interface::ControllerInterface  
controller_interface::ControllerInterface  
controller_interface::ControllerInterface
```



# References

- ros\_control [paper](#) in the Journal of Open Source Software
- ros2\_control resources
  - <https://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!



Karsten Knese, Victor Lopez,  
Jordan Palacios, Olivier  
Stasse, Mathias Arbo, Jaron  
Lundwall, Colin MacKenzie,  
Matthew Reynolds, Andy  
Zelenak, Lovro Ivanov, Jafar  
Abdi, Tyler Weaver, Márk  
Szitanics, Anas Abou Allaban,  
Yutaka Kondo, Mateus  
Amarante, Auguste Bourgois  
and many more!