Fraunhofer Institute for Manufacturing Engineering and Automation IPA

ROS2 CANopen

Supporting CANopen in ROS2

Design goals

Why complete reimplementation for ROS2?

Understandability and Extendability

Maintainability

Robust Parallel Requests

Plain ROS2 Interface

ros2_control Interface

Extensive Documentation

Overview

General

- Based on the Lely Core CANopen Stack (https://gitlab.com/lely_industries/lely-core)
- Based on ROS2 components concept, CANopen master and drivers are components
- Don’t hardcode, configure
- Licensed under Apache 2.0 where possible (currently only canopen_402_driver needs to be under LGPLv3)

canopen
Package aggregating ros2_canopen packages

canopen_core
Device containers, master and driver interfaces, standard master.

canopen_interfaces
ROS interface descriptions

canopen_base_driver
Abstract driver for interacting with lely_core_libraries.

canopen_proxy_driver
Generic driver with interface for nmt, sdo, pdo communications

canopen_402_driver
A driver for motion controllers implementing CIA402 profile.

canopen_ros2_control
ros2_control system interface

canopen_ros2_controller
Controller for sending generic commands

canopen_tests
Contains tests for canopen stack that need mock slaves.

lely_core_libraries
A ros2 wrapper for lely core libraries.

canopen_fake_slaves
Contains slaves that mock the behaviour of real devices

…

Configuration

Different Configuration Options

Master

- Definition node id
- Definition of component that provides the master driver
- Further master configuration options such as sync period or heartbeat are available

Driver

- Definition of name
- Definition of node id
- Definition of component that provides the master
- Definition of SDO calls that are automatically executed after device boot
- Definition of rpdo and tpdo configuration that is automatically configured after device boot

```yaml
master:
  node_id: 1
  driver: "ros2_canopen::MasterDriver"
  package: "canopen_master_driver"
  sync_period: 20000

cia402_device_1:
  node_id: 2
  dcf: "cia402_slave.eds"
  dcf_path: "install/canopen_tests/share/canopen_tests/config/cia402"
  driver: "ros2_canopen::Cia402Driver"
  package: "canopen_402_driver"
  period: 20
  enable_lazy_load: false
  revision_number: 0
  sdo:
    - {index: 0x60C2, sub_index: 1, value: 50} # Set interpolation time for
    - {index: 0x60C2, sub_index: 2, value: -3} # Set base 10-3s
    - {index: 0x6081, sub_index: 0, value: 1000}
    - {index: 0x6083, sub_index: 0, value: 2000}
  tpdo: # TPDO needed statusword, actual velocity, actual position, mode of operation
    1:
      enabled: true
      cob_id: "auto"
      transmission: 0x01
      mapping:
        - {index: 0x6041, sub_index: 0} # status word
        - {index: 0x6061, sub_index: 0} # mode of operaiton display
    2:
      enabled: true
      cob_id: "auto"
      transmission: 0x01
      mapping:
        - {index: 0x6064, sub_index: 0} # position actual value
        - {index: 0x606c, sub_index: 0} # velocity actual position
    3:
      enabled: false
    4:
      enabled: false
  rpdo: # RPDO needed controlword, target position, target velocity, mode of operation
    1:
      enabled: true
      cob_id: "auto"
      mapping:
      - {index: 0x6040, sub_index: 0} # controlword
      - {index: 0x6060, sub_index: 0} # mode of operation
    2:
```

Configure Master Settings

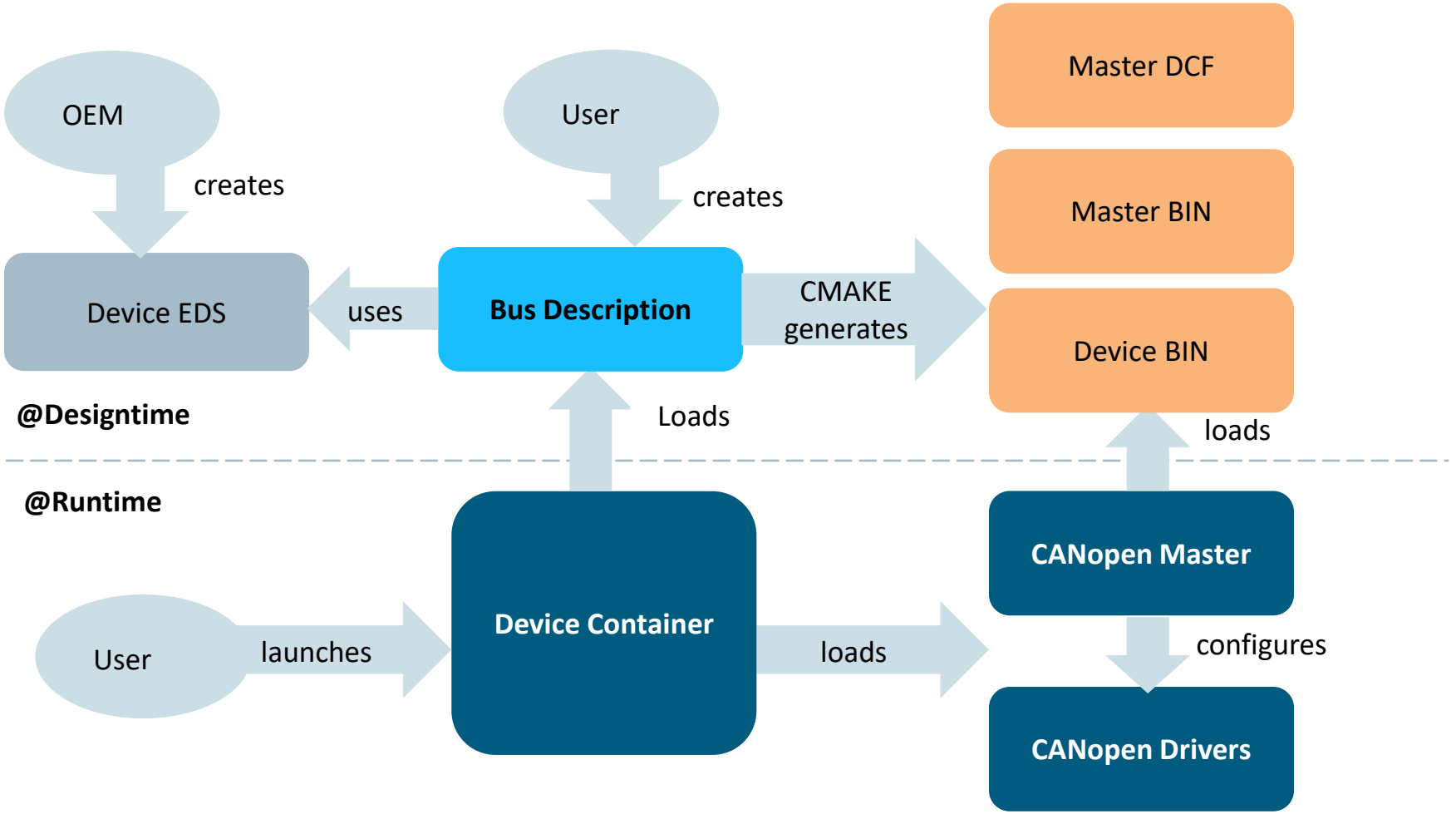
Configure Driver Settings

Configure Device Objects

Configure PDO

Bus Configuration

Automatic configuration artefact generation



Operation

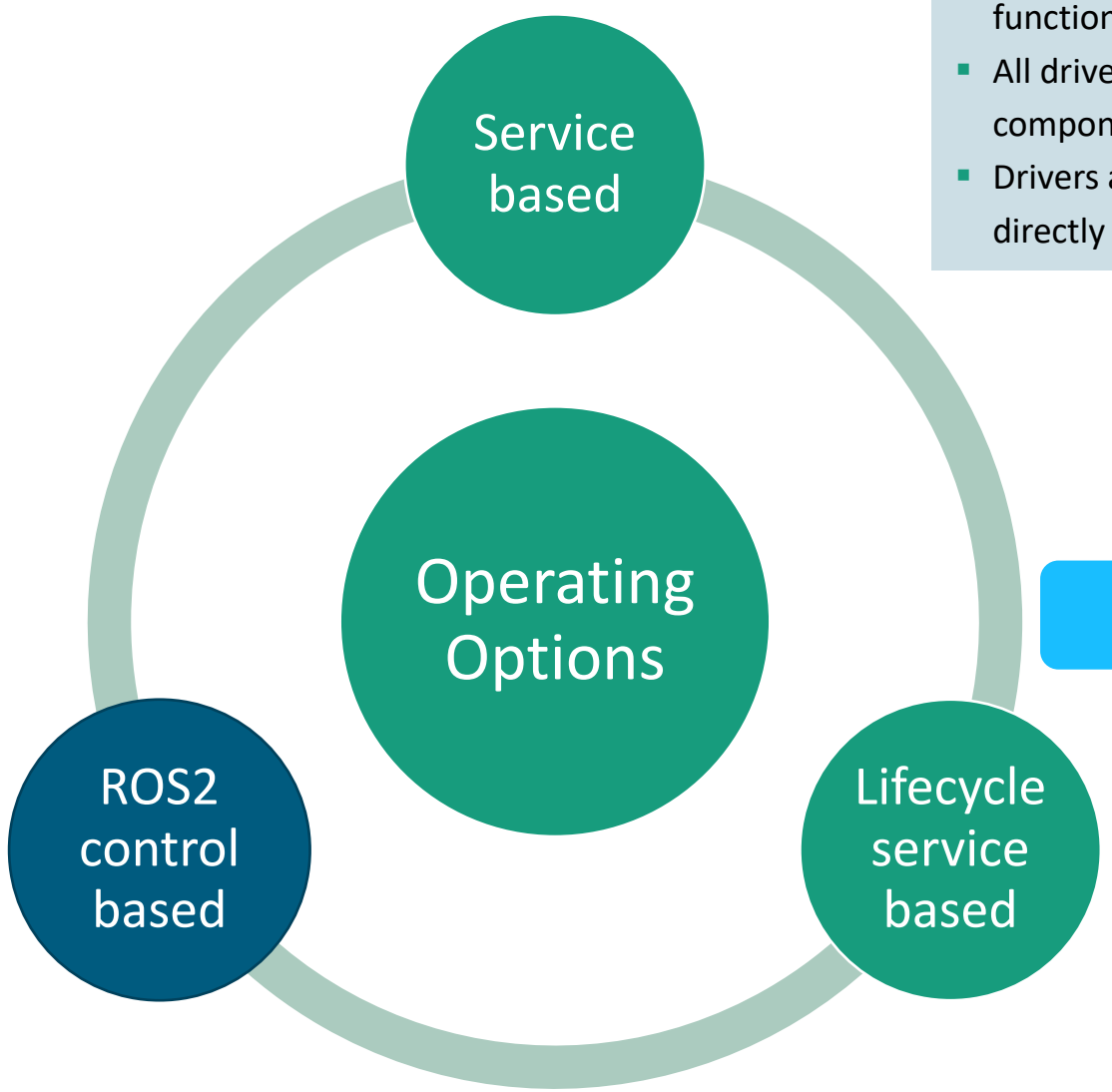
Different Operating Options

Working

Under Development

Simplicity

- All drivers expose their functionality via ROS interface
- All drivers and master are components
- Drivers and master are enabled directly after launch



Real-time Control

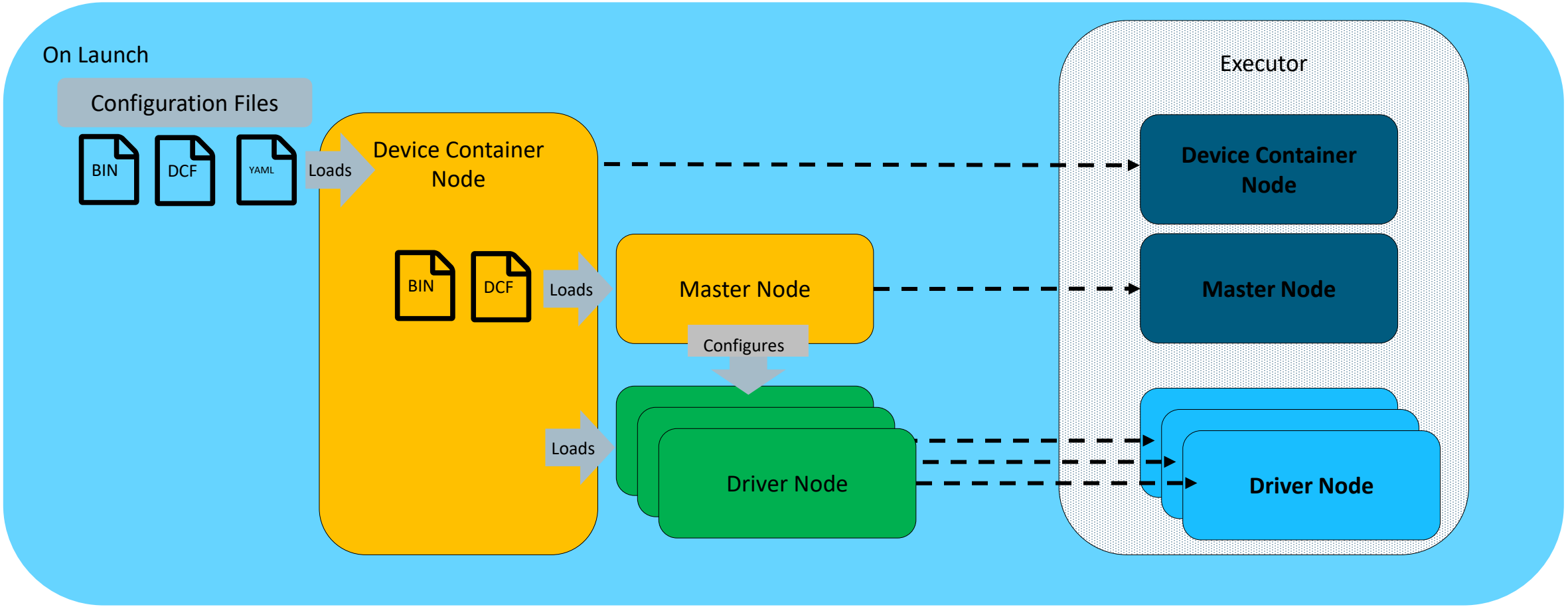
- Driver's functionality is exposed via ros2_control system interface

Robustness

- All drivers expose their functionality via ROS interface
- All drivers and master are lifecycle nodes
- Lifecycle device manager enables managing overall lifecycle

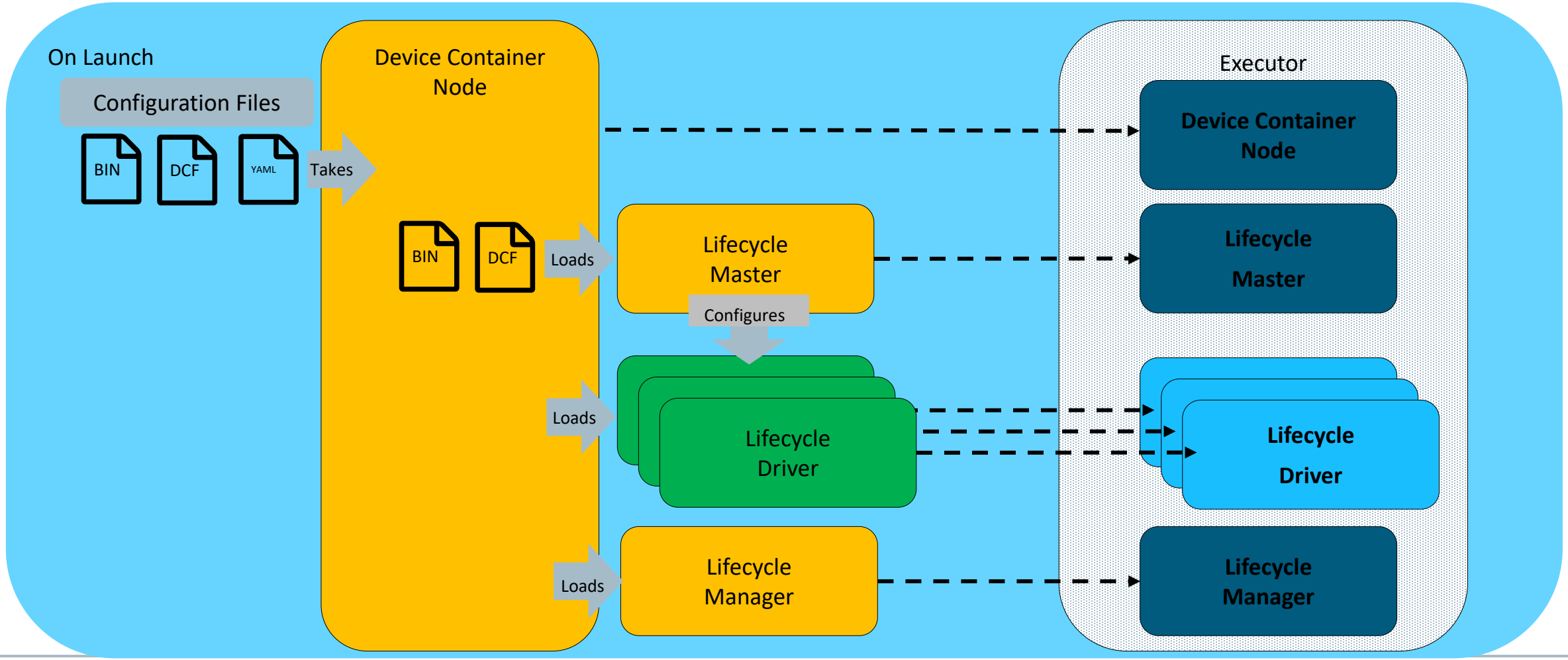
Service-based interface

Simplicity



Lifecycle service-based interface

Robustness

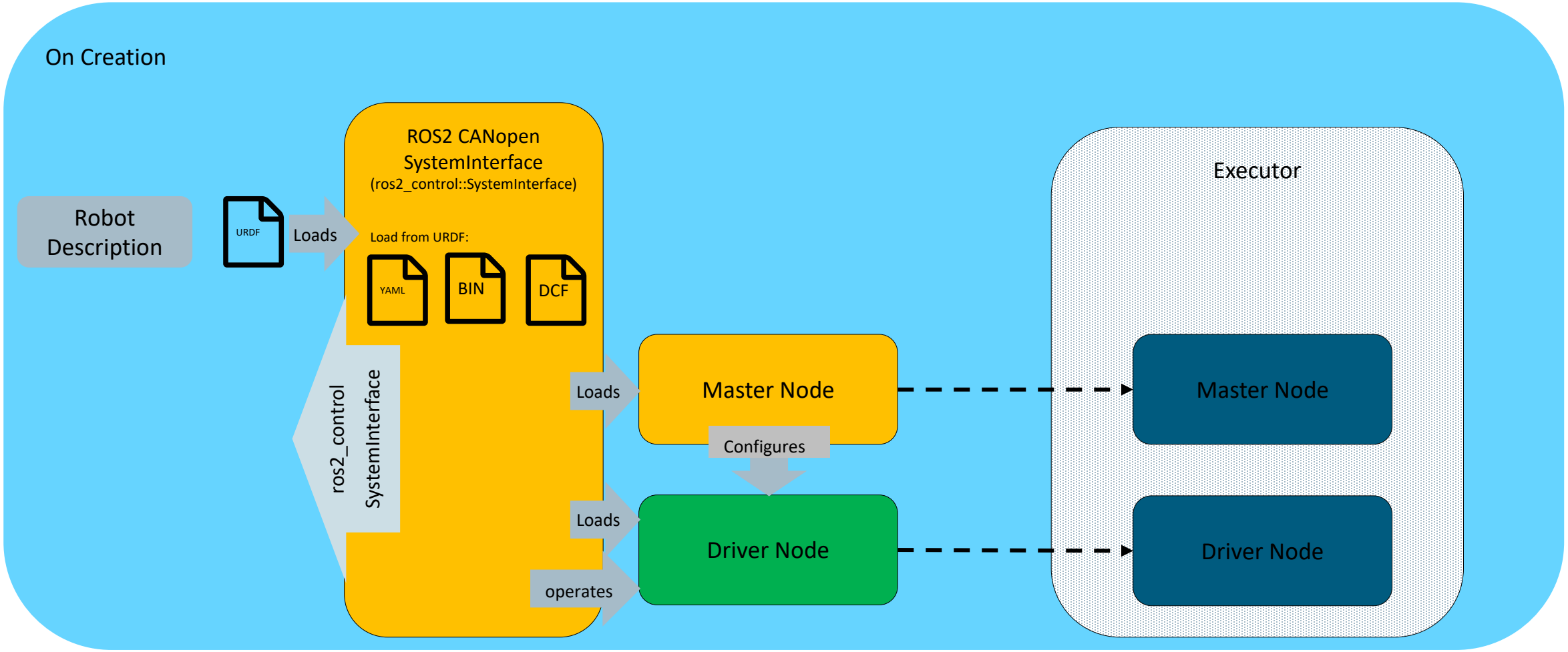


  Open

Fraunhofer IPA

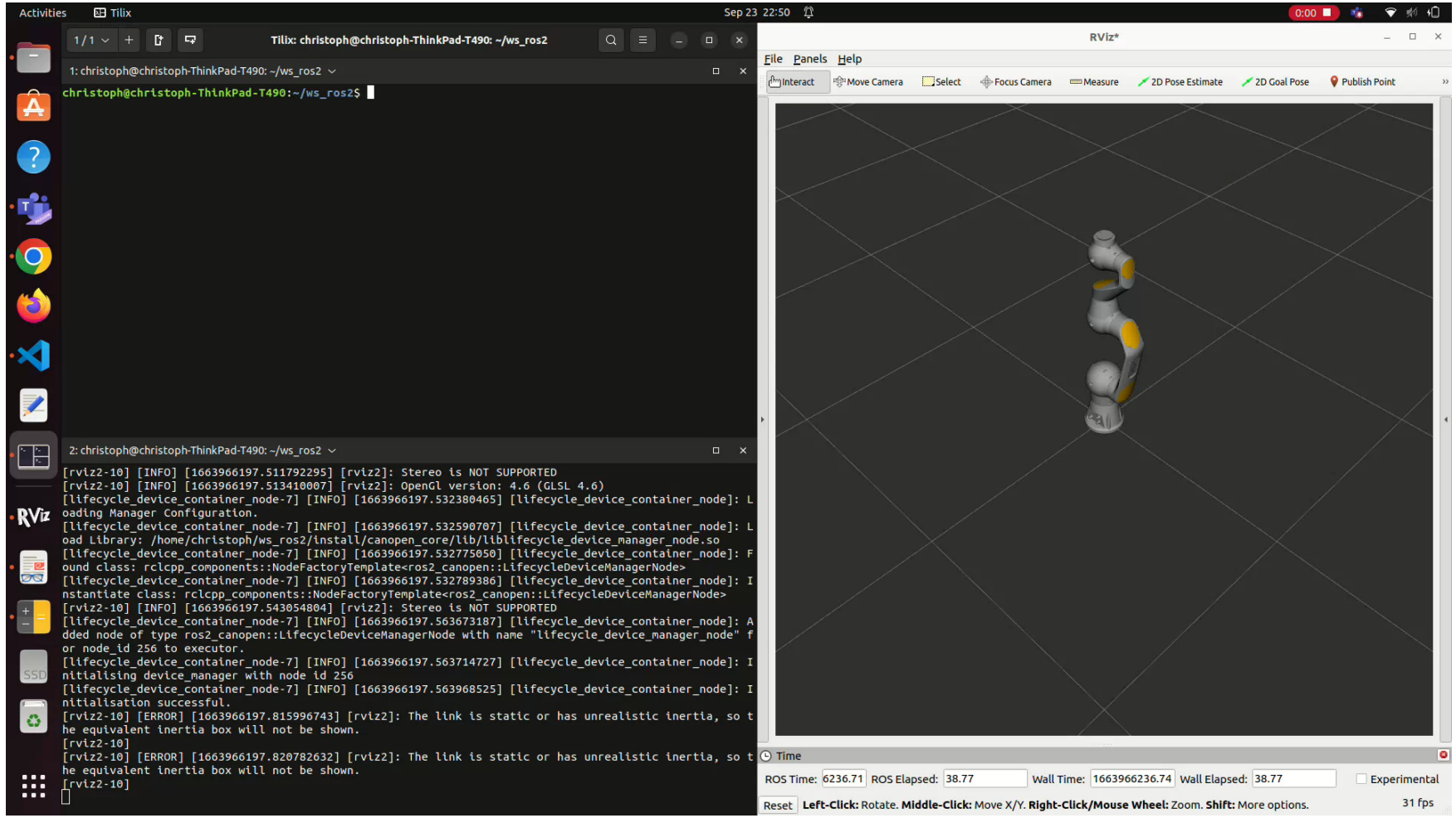
ros2_control System Interface

Controlling – Under Development



Video

ros2_canopen with lifecycle service interface



Further Developments

- Final integration of ros2_control interface
- Streamlining of the different interfaces (removing code duplications etc.)
- Extensive testing (Pilz PRBT, Care-O-Bot and others planned)

Thank you for your attention!

Contact

Harsh Deshpande and Christoph Hellmann Santos
Fraunhofer IPA – Robots and assistive systems
Phone +49 711 970-1097
Christoph.hellmann.santos@ipa.fraunhofer.de

Fraunhofer IPA
Nobelstraße 12
70569 Stuttgart
Germany
www.ipa.fraunhofer.de

Fraunhofer IPA

Fraunhofer Institute for Manufacturing Engineering and Automation IPA

Future is our product

Sustainable. Personalized. Smart.