Migration of F1/10 Autonomous Driving Stack to ROS 2

Martin Endler

https://github.com/pokusew/fel-project

Project **Goal**

- **main:** migrate a selected part of the CTU's F1/10 project from ROS 1 to ROS 2
  1. study and describe ROS 1 and ROS 2, **compare their differences**
  2. select a part for the migration and port the code
  3. solve compatibility problems / missing drivers, etc.
  4. setup hardware (NVIDIA Jetson TX2) for ROS 2


HOKUYO

Work

1. study ROS 1 (Kinetic Kame)
2. study ROS 2 (Foxy Fitzroy)
3. port the selected part of the f1tenth codebase (Follow The Gap) to ROS 2
4. verify its workings in the Stage simulator (Ubuntu and macOS)
5. prepare an Ubuntu (JetPack / L4T) image for NVIDIA Jetson TX2
    a. build ROS 2 from sources
    b. setup boot from SD Card

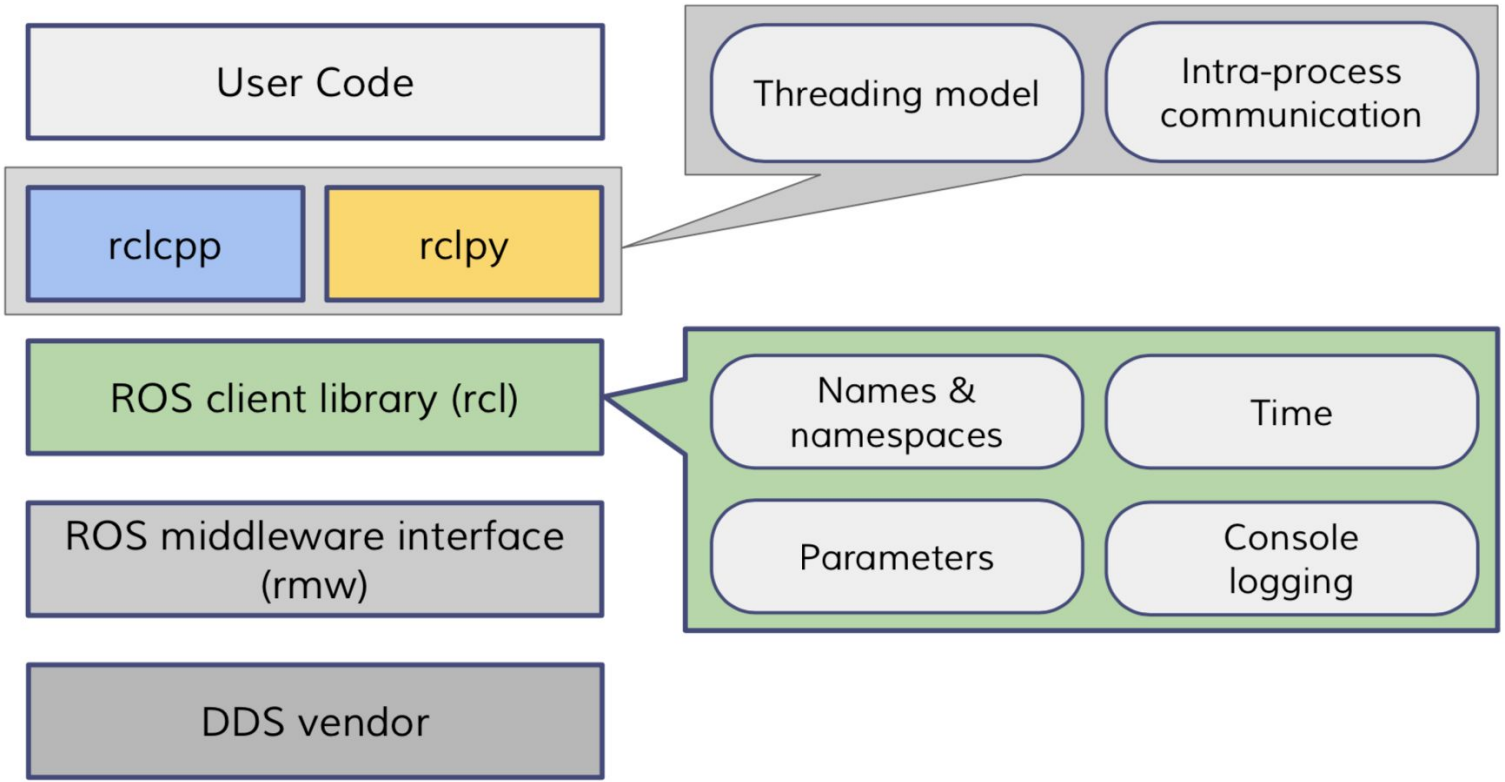
ROS History

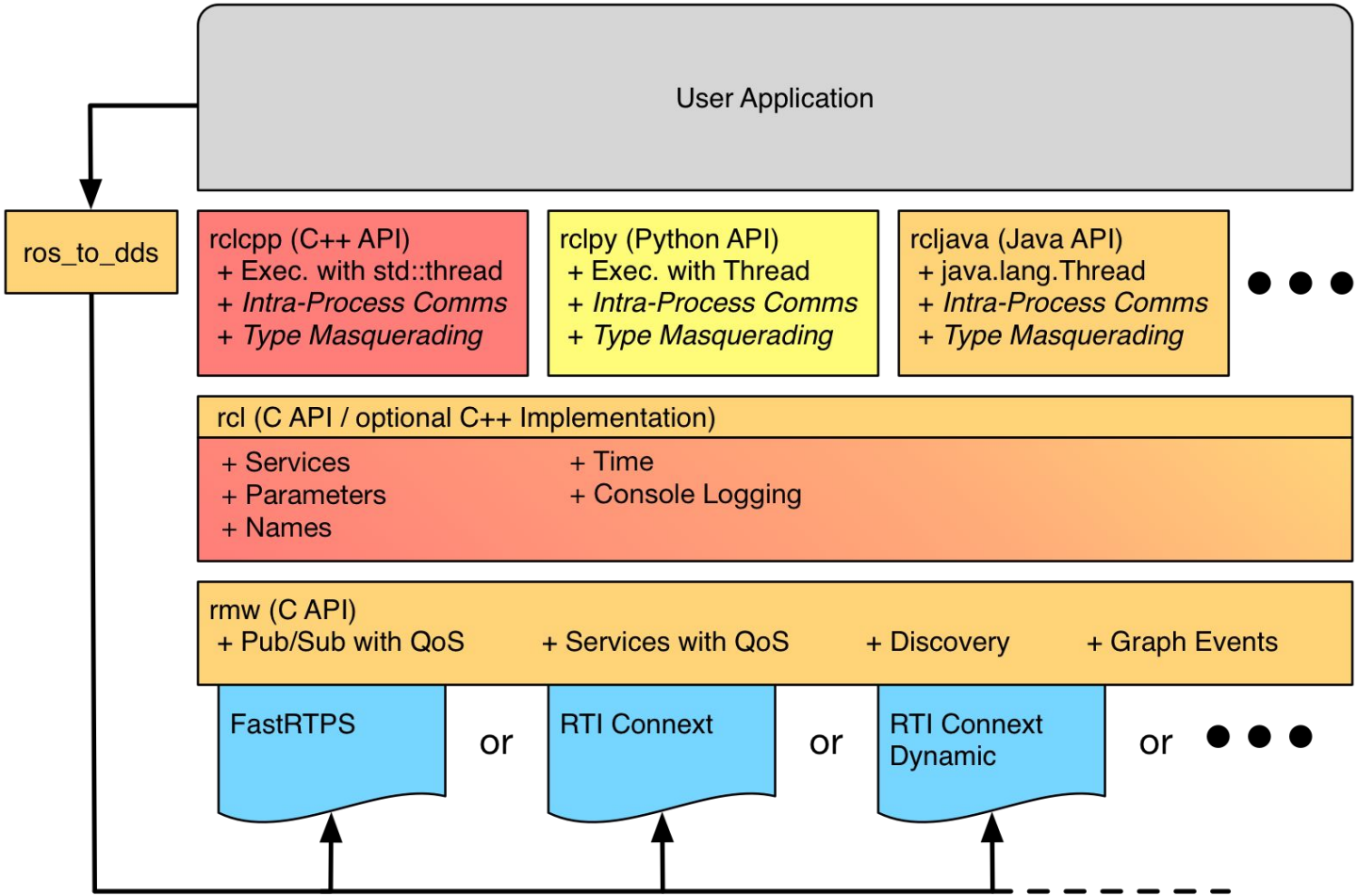
- 2007: ROS project started
- 2010: ROS 1.0
- 2016: ROS Kinetic Kame (Ubuntu 16)
- **2017**: 1st release ROS 2 Ardent Apalone
- 2020: 6th release ROS 2 Foxy Fitzroy (Ubuntu 20)

ROS 2

- significant improvements over ROS 1
- **same concepts** but redesigned internal **architecture**
- emphasis on code quality, performance, Real-Time support, …
- solves many pain points of ROS 1
- **better build system** (colcon + ament_cmake, instead of catkin)
  - support for pure Python packages
  - support for pure CMake packages

ROS 2 Architecture



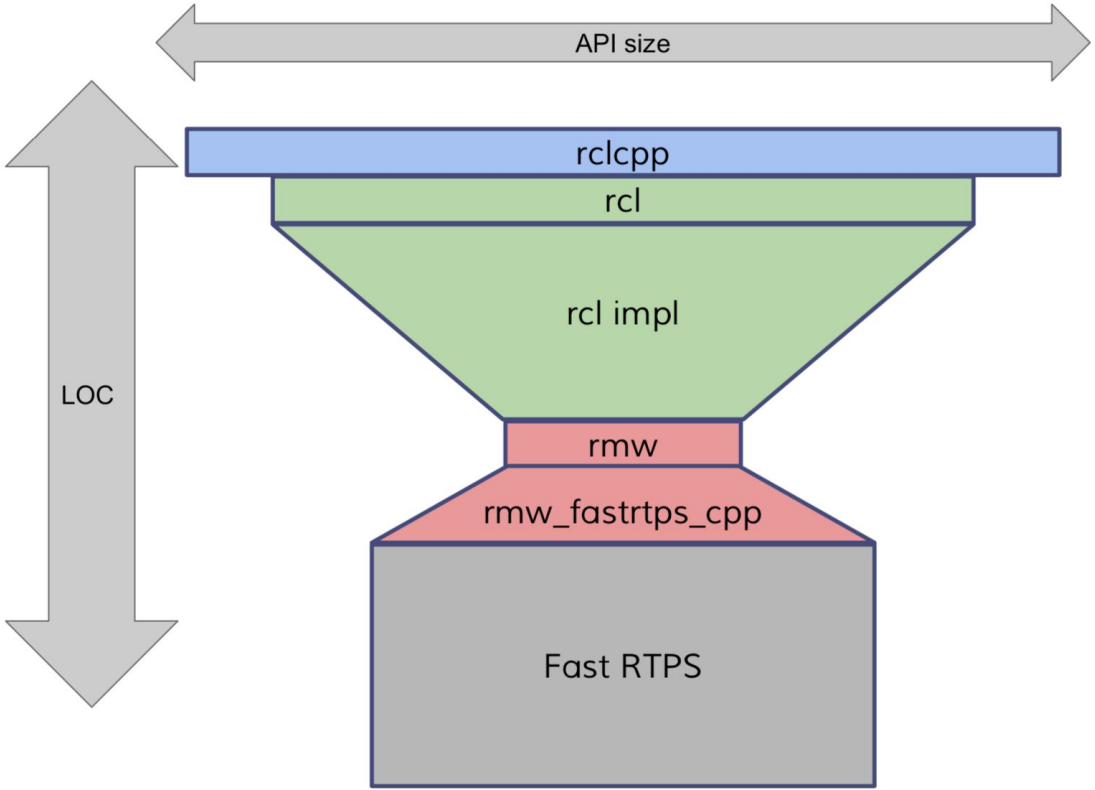


* *Intra-Process Comms* and *Type Masquerading* could be implemented in the client library, but may not currently exist.

ROS 2

- **no master, distributed discovery**
- nodes, topics, services + **actions**
- no global parameter server
- parameters implemented as services of nodes
- DDS instead of TCPROS/UDPROS
  - advanced QoS settings
  - Real-Time friendly
  - support for multiple vendors
    (Eclipse Cyclone DDS, eProsima Fast DDS, and RTI Connext DDS)
- Composable Nodes (ROS 1: Nodelets)

ROS 2 Architecture

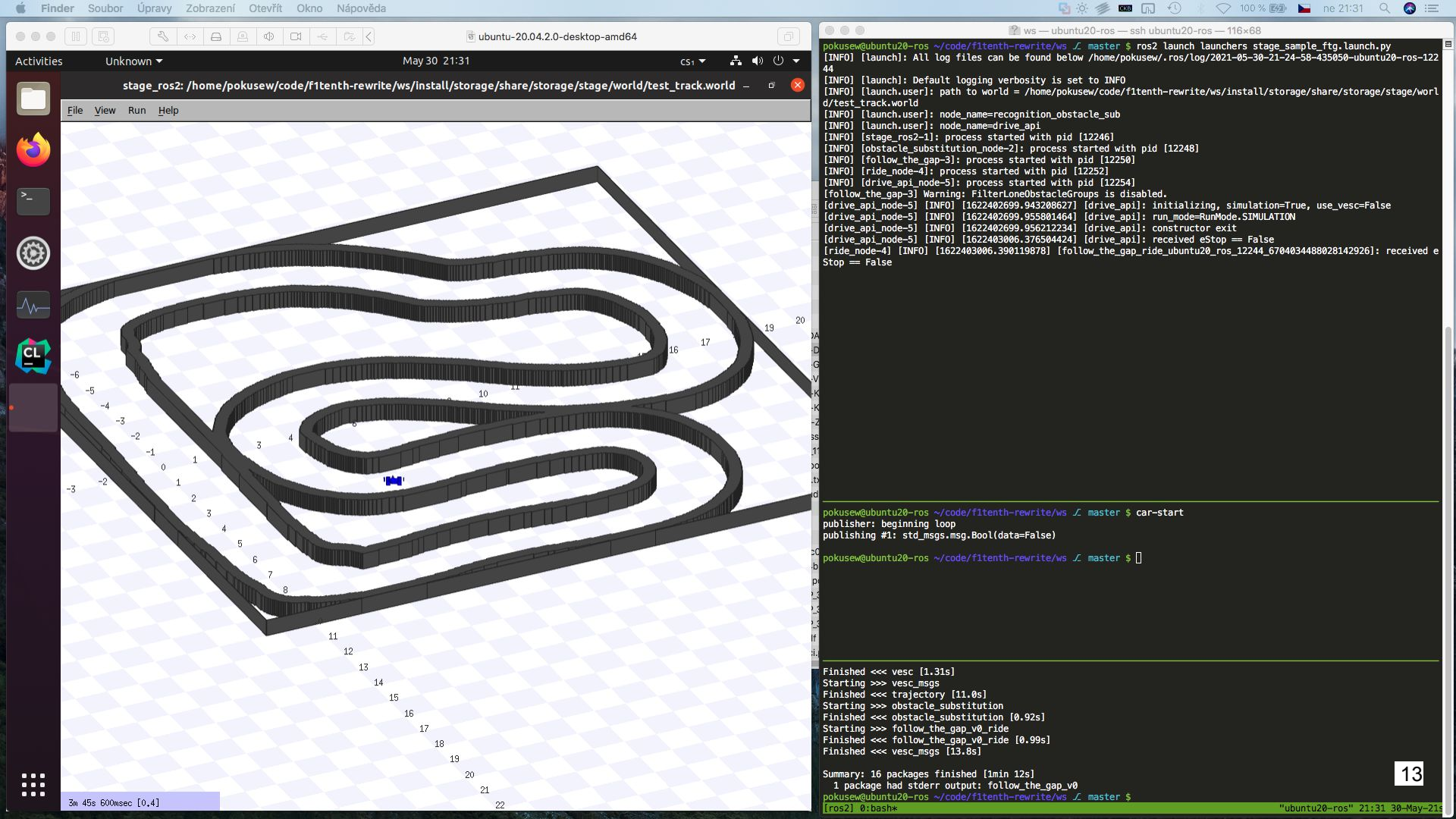


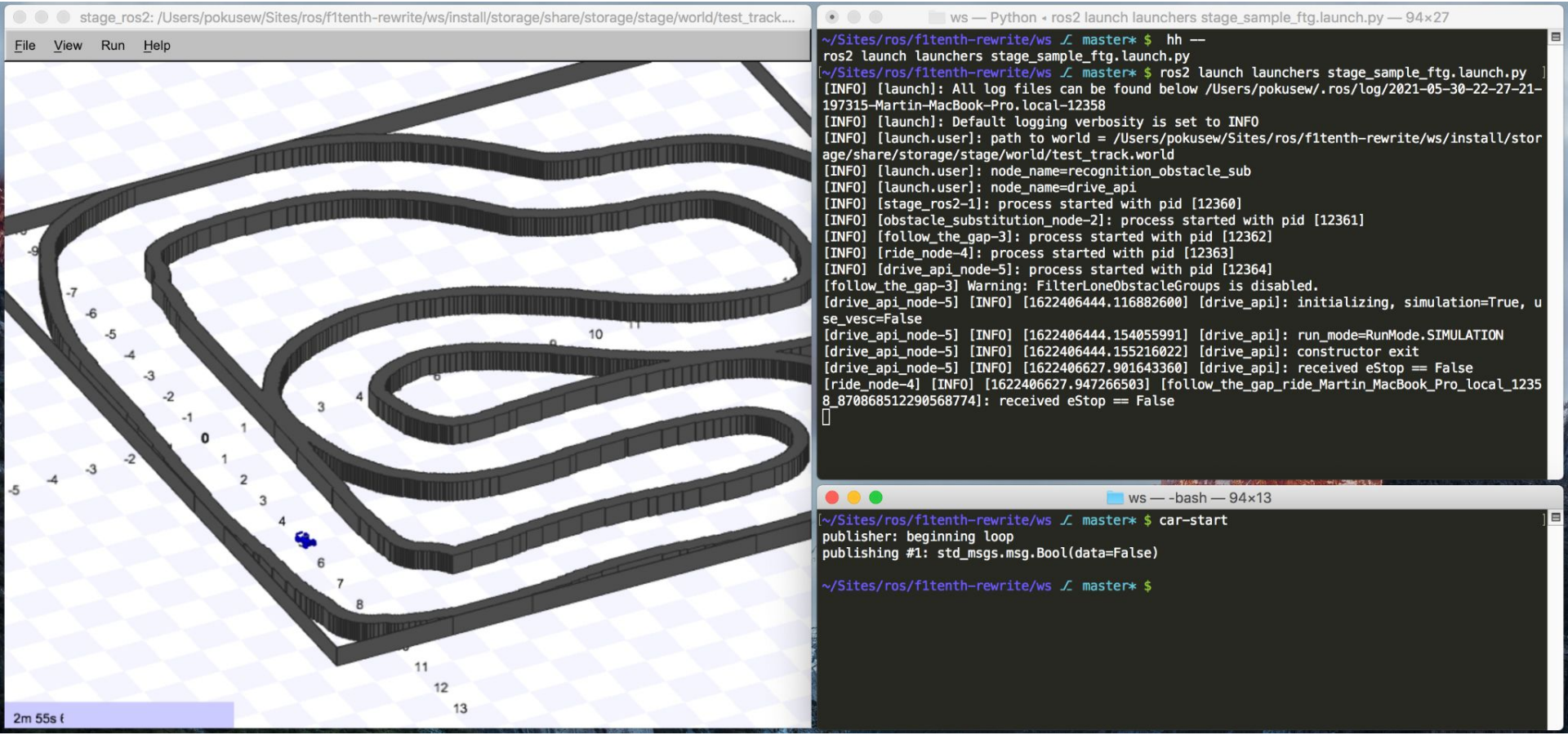
Project **Results**

- working Follow The Gap on **ROS 2** in **Stage** simulator
- **documentation** for setting up Ubuntu / macOS / Jetson TX2 for ROS 1 and 2
- instructions for **building ROS 2 from sources** incl. **Stage simulator**
- documentation for **using IDEs with ROS 1 and ROS 2** (Ubuntu and macOS)
  - **CLion**
  - **Visual Studio Code** (incl. Remote Mode over SSH)
- NVIDIA Jetson TX2 setup
  - latest Jetpack 4.5.1 [L4T 32.5.1] (Ubuntu 18)
  - setup **booting of Jetson TX2 from SD Card**

f1tenth Follow The Gap – ported packages

- *decision_and_control/*
  - **follow_the_gap_v0** (C++)
  - **follow_the_gap_v0_ride** (Python, launch files)
- **launchers** (Python, launch files)
  - `ros2 launch launchers stage_sample_ftg.launch.py`
- *messages/*
  - **command_msgs** (IDL)
  - **f1tenth_race** (IDL)
  - **obstacle_msgs** (IDL)
  - **plan_msgs** (IDL)
  - **trajectory** (IDL)
  - **vesc_msgs** (IDL)
- *perception/*
  - *recognition/*
    - **obstacle_substitution** (Python, launch files)
- **storage** (data files)
- *vehicle_platform/*
  - **drive_api** (Python, launch files)
  - **drive_api_msgs** (IDL)


Finder Soubor Úpravy Zobrazení Otevřít Okno Nápověda
ne 21:31
ubuntu-20.04.2.0-desktop-amd64
Activities
Unknown
May 30 21:31
stage_ros2: /home/pokusew/code/f1tenth-rewrite/ws/install/storage/share/storage/stage/world/test_track.world
File View Run Help
3m 45s 600msec [0,4]
ws — ubuntu20-ros — ssh ubuntu20-ros — 116×68
pokusew@ubuntu20-ros ~/code/f1tenth-rewrite/ws master $ ros2 launch launchers stage_sample_ftg.launch.py
[INFO] [launch]: All log files can be found below /home/pokusew/.ros/log/2021-05-30-21-24-58-435050-ubuntu20-ros-12244
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [launch.user]: path to world = /home/pokusew/code/f1tenth-rewrite/ws/install/storage/share/storage/stage/world/test_track.world
[INFO] [launch.user]: node_name=recognition_obstacle_sub
[INFO] [launch.user]: node_name=drive_api
[INFO] [stage_ros2-1]: process started with pid [12246]
[INFO] [obstacle_substitution_node-2]: process started with pid [12248]
[INFO] [follow_the_gap-3]: process started with pid [12250]
[INFO] [ride_node-4]: process started with pid [12252]
[INFO] [drive_api_node-5]: process started with pid [12254]
[follow_the_gap-3] Warning: FilterLoneObstacleGroups is disabled.
[drive_api_node-5] [INFO] [1622402699.943208627] [drive_api]: initializing, simulation=True, use_vesc=False
[drive_api_node-5] [INFO] [1622402699.955801464] [drive_api]: run_mode=RunMode.SIMULATION
[drive_api_node-5] [INFO] [1622402699.956212234] [drive_api]: constructor exit
[drive_api_node-5] [INFO] [1622403006.376504424] [drive_api]: received eStop == False
[ride_node-4] [INFO] [1622403006.390119878] [follow_the_gap_ride_ubuntu20_ros_12244_6704034488028142926]: received eStop == False
pokusew@ubuntu20-ros ~/code/f1tenth-rewrite/ws master $ car-start
publisher: beginning loop
publishing #1: std_msgs.msg.Bool(data=False)
pokusew@ubuntu20-ros ~/code/f1tenth-rewrite/ws master $
Finished <<< vesc [1.31s]
Starting >>> vesc_msgs
Finished <<< trajectory [11.0s]
Starting >>> obstacle_substitution
Finished <<< obstacle_substitution [0.92s]
Starting >>> follow_the_gap_v0_ride
Finished <<< follow_the_gap_v0_ride [0.99s]
Finished <<< vesc_msgs [13.8s]
Summary: 16 packages finished [1min 12s]
1 package had stderr output: follow_the_gap_v0
pokusew@ubuntu20-ros ~/code/f1tenth-rewrite/ws master $
[ros2] 0:bash*
"ubuntu20-ros" 21:31 30-May-21
13


stage_ros2: /Users/pokusew/Sites/ros/f1tenth-rewrite/ws/install/storage/share/storage/stage/world/test_track....
File View Run Help
2m 55s
ws — Python ◂ ros2 launch launchers stage_sample_ftg.launch.py — 94×27
~/Sites/ros/f1tenth-rewrite/ws ⎇ master* $ hh --
ros2 launch launchers stage_sample_ftg.launch.py
~/Sites/ros/f1tenth-rewrite/ws ⎇ master* $ ros2 launch launchers stage_sample_ftg.launch.py
[INFO] [launch]: All log files can be found below /Users/pokusew/.ros/log/2021-05-30-22-27-21-197315-Martin-MacBook-Pro.local-12358
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [launch.user]: path to world = /Users/pokusew/Sites/ros/f1tenth-rewrite/ws/install/storage/share/storage/stage/world/test_track.world
[INFO] [launch.user]: node_name=recognition_obstacle_sub
[INFO] [launch.user]: node_name=drive_api
[INFO] [stage_ros2-1]: process started with pid [12360]
[INFO] [obstacle_substitution_node-2]: process started with pid [12361]
[INFO] [follow_the_gap-3]: process started with pid [12362]
[INFO] [ride_node-4]: process started with pid [12363]
[INFO] [drive_api_node-5]: process started with pid [12364]
[follow_the_gap-3] Warning: FilterLoneObstacleGroups is disabled.
[drive_api_node-5] [INFO] [1622406444.116882600] [drive_api]: initializing, simulation=True, use_vesc=False
[drive_api_node-5] [INFO] [1622406444.154055991] [drive_api]: run_mode=RunMode.SIMULATION
[drive_api_node-5] [INFO] [1622406444.155216022] [drive_api]: constructor exit
[drive_api_node-5] [INFO] [1622406627.901643360] [drive_api]: received eStop == False
[ride_node-4] [INFO] [1622406627.947266503] [follow_the_gap_ride_Martin_MacBook_Pro_local_12358_870868512290568774]: received eStop == False
ws — -bash — 94×13
~/Sites/ros/f1tenth-rewrite/ws ⎇ master* $ car-start
publisher: beginning loop
publishing #1: std_msgs.msg.Bool(data=False)
~/Sites/ros/f1tenth-rewrite/ws ⎇ master* $

Code

- github.com/pokusew/**fel-project**
  – The project homepage, final report, links to all related repositories
- github.com/pokusew/**f1tenth-rewrite**
  – ROS 2 port of CTU F1/10 project
- github.com/pokusew/**ros-setup**
  – Setup notes for using ROS on different platforms (Ubuntu, macOS, NVIDIA Jetson TX2) and using ROS with IDEs (JetBrains CLion, Visual Studio Code)
- github.com/pokusew/**ros2-build**
  – colcon workspace for building ROS 2 from sources on different platforms together with additional packages (e.g. Stage simulator)
- github.com/pokusew/**rh**
  – A simple helper to make working with different ROS versions and projects easier

Conclusion

- ROS 1 and ROS 2 comparison
- **successful migration** of the Follow The Gap application **to ROS 2**
- demonstration in Stage simulator on Ubuntu and macOS
- NVIDIA Jetson TX2 setup for running ROS 2 applications
- booting of Jetson TX2 from SD Card
- a collection of setup guides and documentation
  that covers various aspects of working with ROS
- everything published on GitHub

Future Work

- port missing pieces so it can run on a real F1/10 model car
  - Orbitty Carrier BSP for L4T image
  - LIDAR
  - VESC
- compare ROS 2 vs ROS 1 (performance, real-time properties, temporal determinism, communication overheads etc.)

Questions?