7th ROS-Industrial Conference
Stuttgart, Germany (EU)
rosindustrial.org/riceu2019

ROS2 on VxWorks

One project on Wind River Labs

ANDREI KHOLODNYI, WIND RIVER

AEROSPACE AND DEFENSE SECTOR

CHEMICAL SECTOR

COMMERCIAL BUILDING SECTOR

COMMUNICATIONS SECTOR

CRITICAL MANUFACTURING SECTOR

DAMS SECTOR

EMERGENCY SERVICES SECTOR

ENERGY SECTOR

FINANCIAL SERVICES SECTOR

FOOD AND AGRICULTURE SECTOR

GOVERNMENT BUILDING SECTOR

IT SECTOR

MEDICAL SECTOR

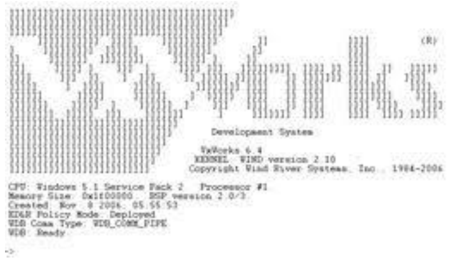
NUCLEAR SECTOR

TRANSPORTATION/AUTO SECTOR

WATER AND WASTEWATER SECTOR

What is VxWorks RTOS?

- 32/64 bits on Arm/Intel/MIPS/PowerPC
- Proprietary real-time OS, POSIX PSE52
- Kernel/user space separation, user space optional
- C/C++11/14, possible to develop kernel C++ modules and user apps
- Safety certifiable: DO-178, ISO 26262, IEC 61508
- Toolchain LLVM 8, Dinkumware C/C++ libs
- Proprietary build system
- Kernel shell
- Eclipse-based IDE, Windows/Linux hosts



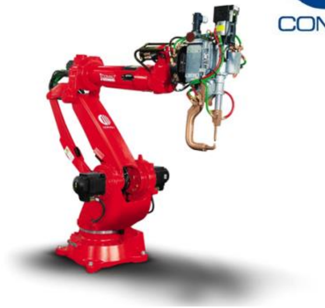
```
                         Development System

                        VxWorks 6.4
                      KERNEL: WIND version 2.10
                    Copyright Wind River Systems, Inc., 1984-2006

CPU: Windows 5.1 Service Pack 2   Processor #1
Memory Size: 0x1f00000   BSP version 2.0/3
Created: Nov  8 2006, 05:55:53
ED&R Policy Mode: Deployed
WDB Comm Type: WDB_COMM_PIPE
WDB: Ready

->
```

Industry Examples



COMAU

KUKA

SIEMENS

MITSUBISHI ELECTRIC

ABB

NASA

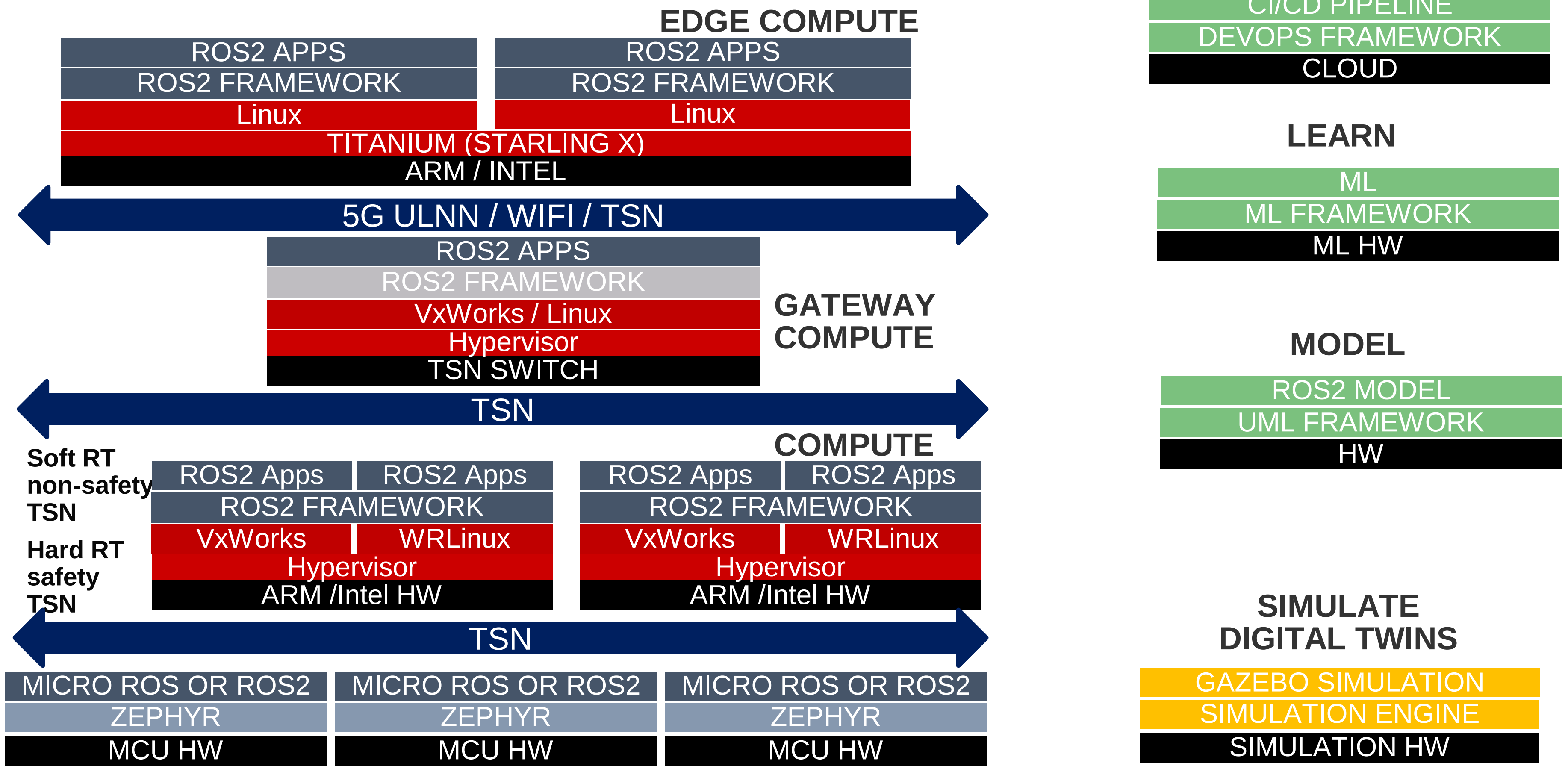
YASKAWA

NXC100 robot controller offers high performance, open communication, and integrated cell control
NXC100
ROBOT CONTROLLER

BOSCH

Schneider Electric

Industrial Profile



- ❑ **Soft RT**
- ❑ **non-safety**
- ❑ **TSN**

- ❑ **Hard RT**
- ❑ **safety**
- ❑ **TSN**

Embedded Development Landscape

- Heterogeneous HW: MCU, CPU, GPU, VPU, TCU, FPGA, SOC
- Embedded, Edge, Fog, Cloud
- Sensors: Camera, Lidar, Radar, IMU, Ultrasonic
- Real-time: hard, soft, best effort
- C/C++ programming language
- Not enough SW engineers with embedded development skills
- Many people can‘t program (embedded)

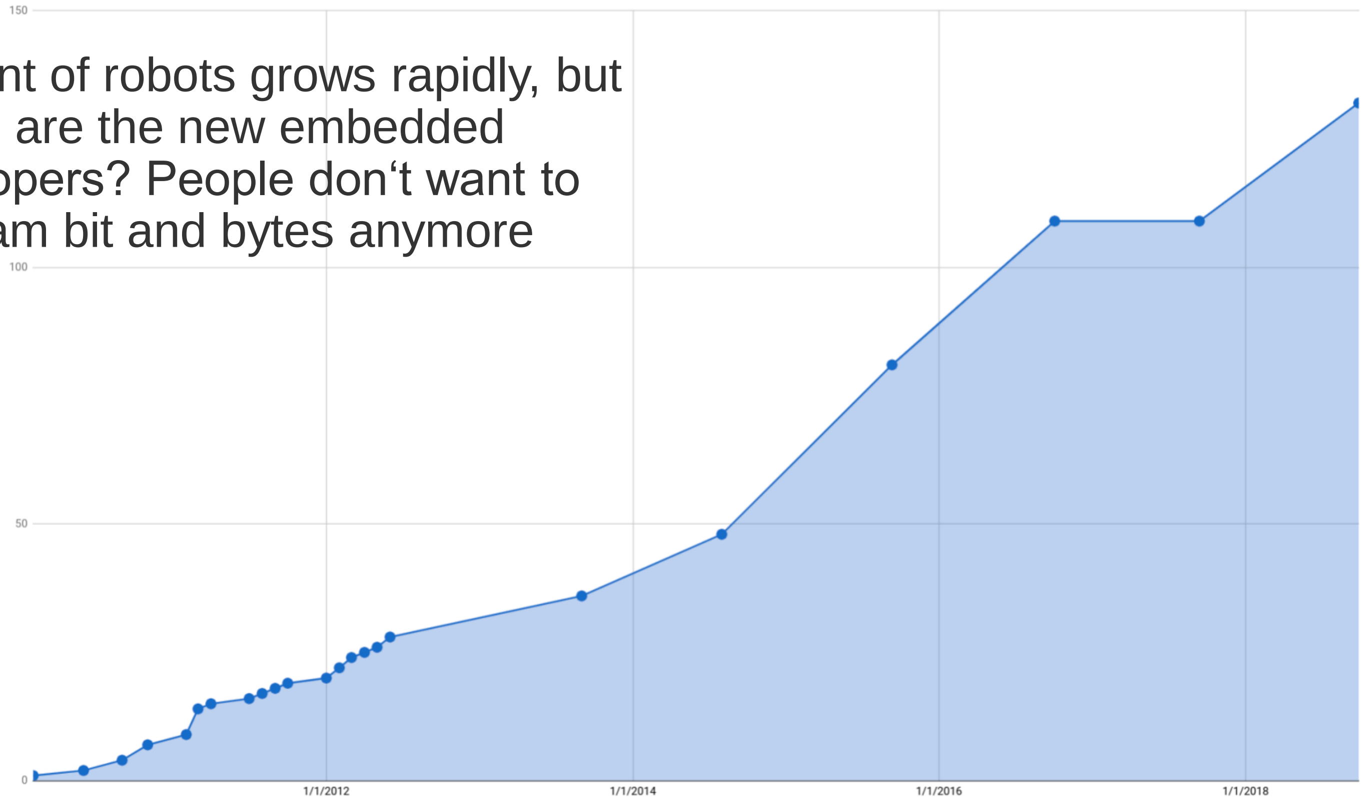
Embedded Development Landscape

- Heterogeneous HW: MCU, CPU, GPU, VPU, TCU, FPGA, SOC
- Embedded, Edge, Fog, Cloud
- Sensors: Camera, Lidar, Radar, IMU, Ultrasonic
- Real-time: hard, soft, best effort
- C/C++ programming language
- Not enough SW engineers with embedded development skills
- Many people ~~can‘t~~ do not want to program (embedded)
- Stop scaring people with real-time, safety and security

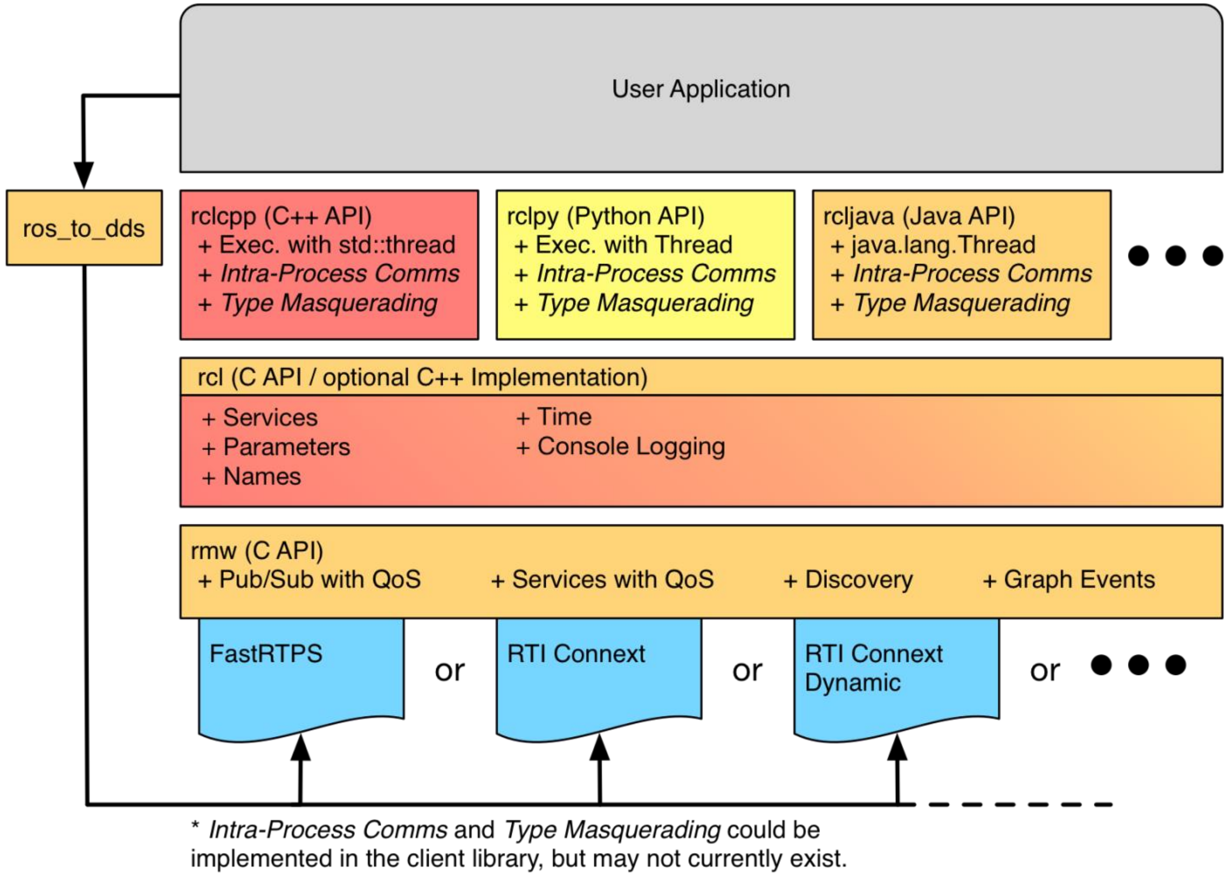
Documented ROS Robots

https://robots.ros.org/

Amount of robots grows rapidly, but where are the new embedded developers? People don‘t want to program bit and bytes anymore



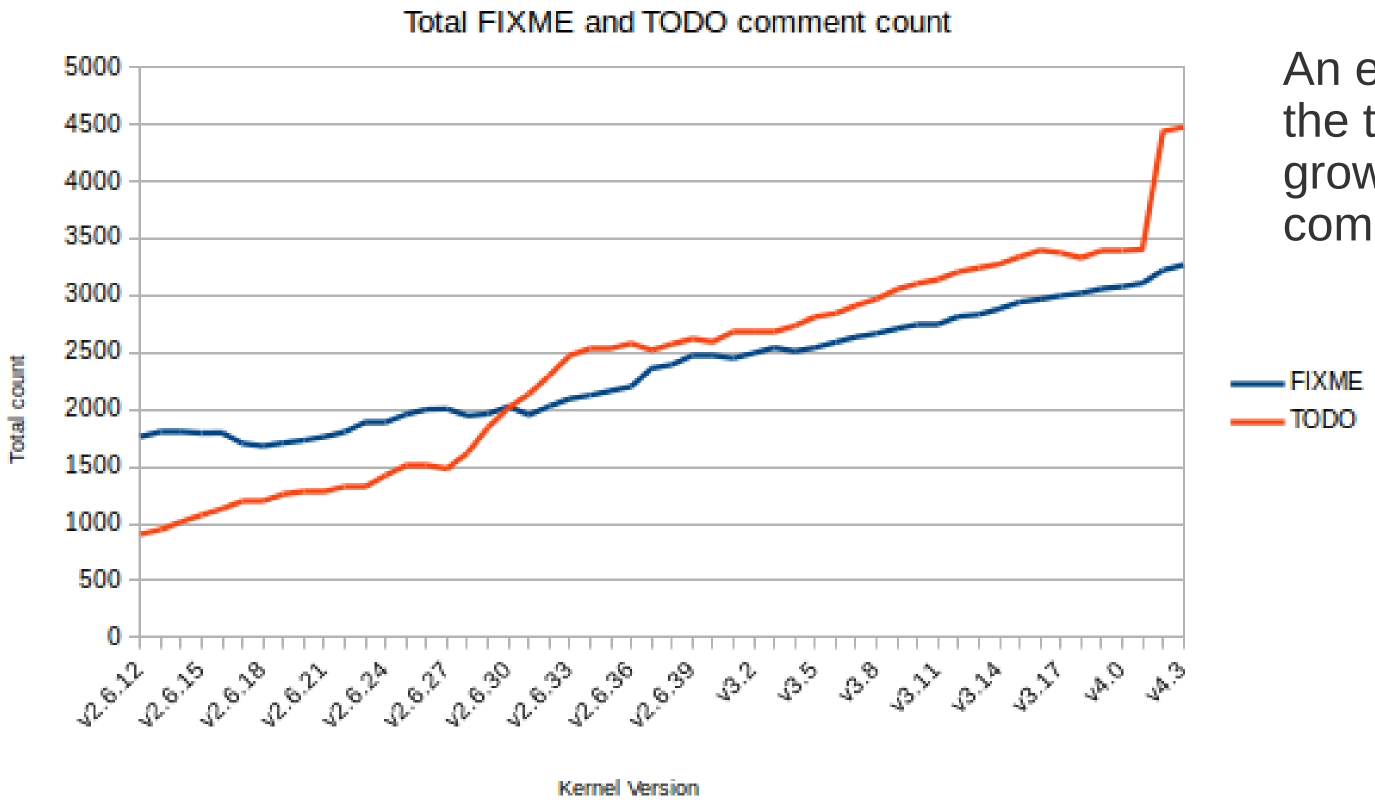
ROS2 Architecture



C/C++ in an embedded API.

A Robotics domain specific API is needed instead

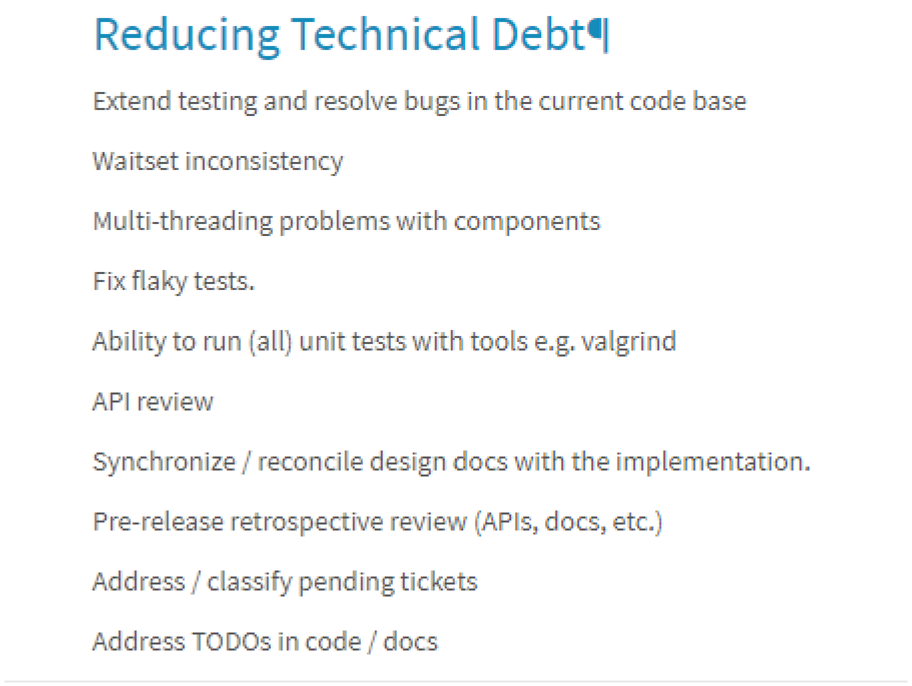
Technical Debt grows



An example of the technica debt grow in the kernl community

http://smackerelofopinion.blogspot.com/2016/01/fixme-and-todo-comments-in-linux-kernel.html

ROS2 Technical Debt



Reducing Technical Debt¶

Extend testing and resolve bugs in the current code base

Waitset inconsistency

Multi-threading problems with components

Fix flaky tests.

Ability to run (all) unit tests with tools e.g. valgrind

API review

Synchronize / reconcile design docs with the implementation.

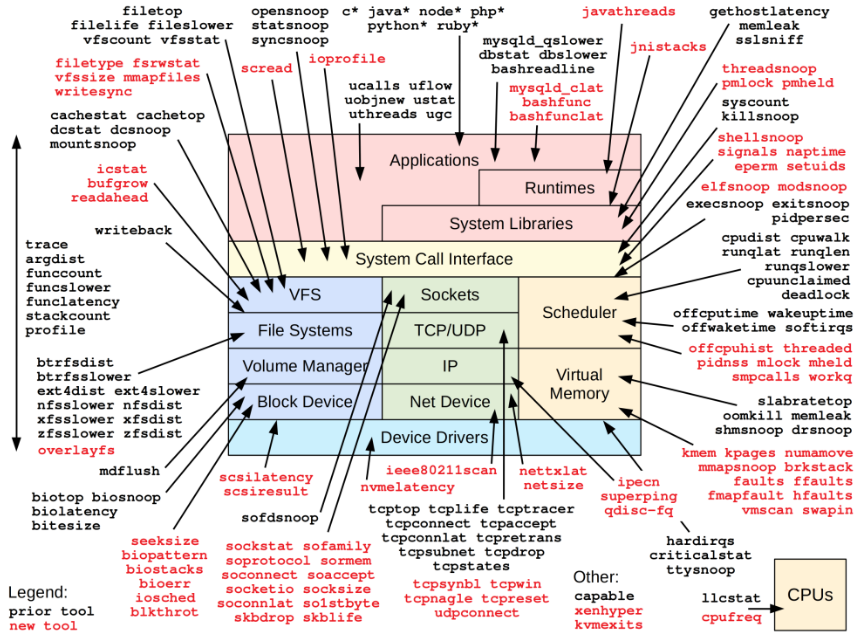
Pre-release retrospective review (APIs, docs, etc.)

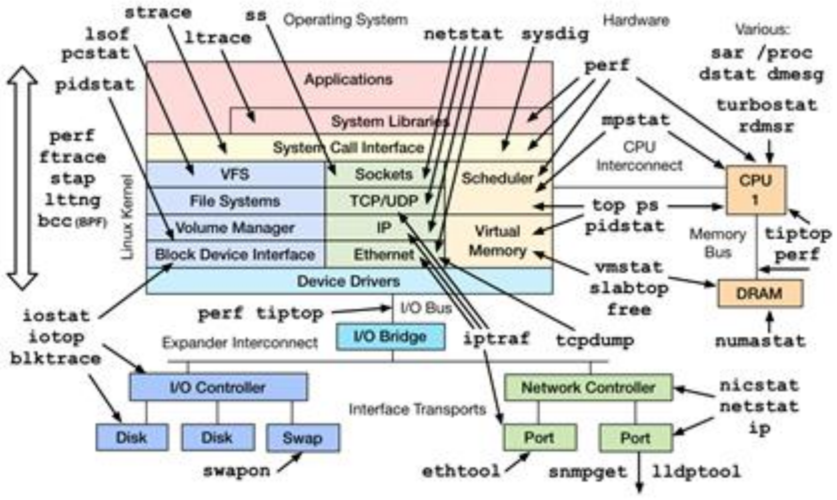
Address / classify pending tickets

Address TODOs in code / docs

It is probably the same for ROS2

Run-time optimization (performance, footprint, RAM, I/O etc.) is very difficult to handle



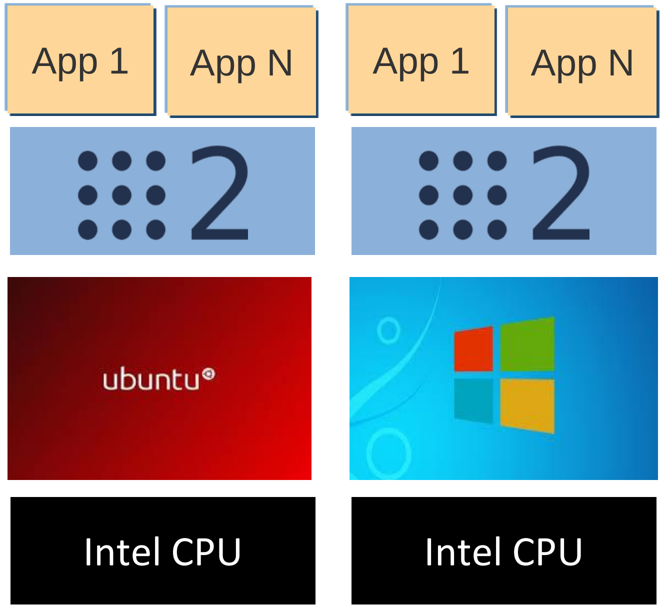


http://www.brendangregg.com/ebpf.html

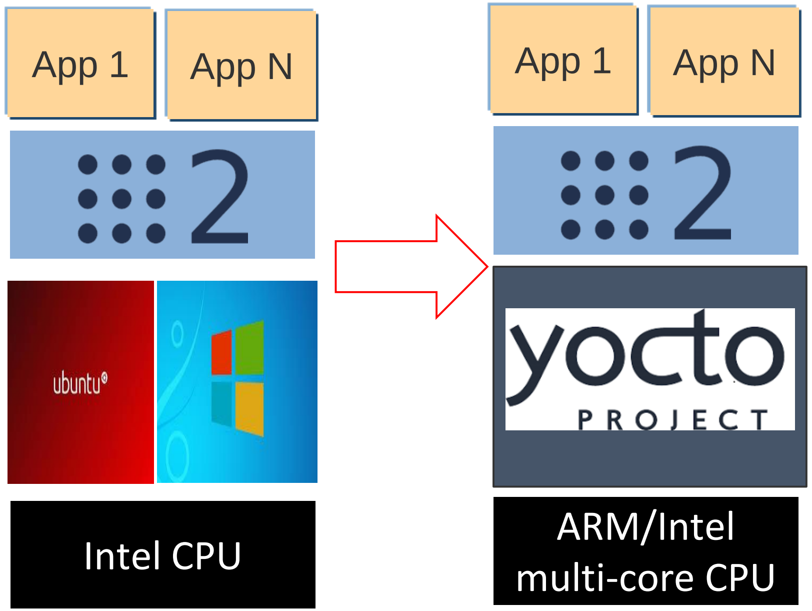
Usability Problem

- Little kids playing with a smartphone, they don‘t know how it works
- Increasing amount of robots versus descreasing amount of embedded software engineers
- Productivity crisis in the embedded software development?
- Look at ROS2 APIs only C++, C and Python – typical embedded
- TSN protocols
- Machine learning networks
- Security
- Safety
- Little kids playing with a robot?

ROS2 Developer Journey Desktop



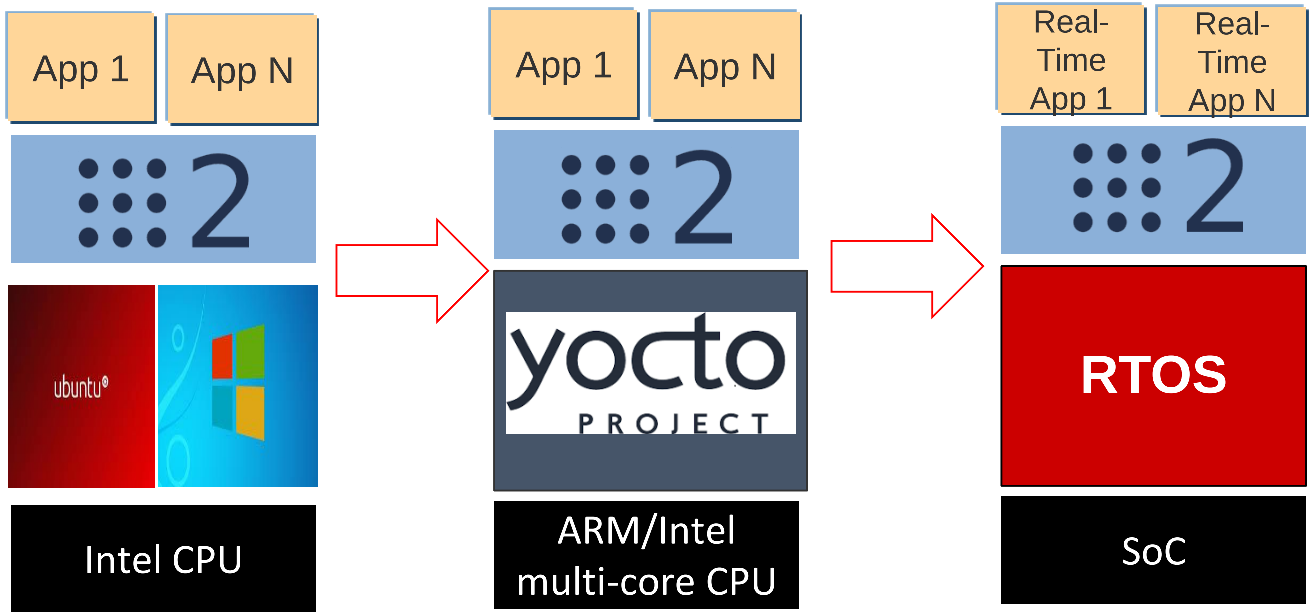
ROS2 Developer Journey embedded Linux development



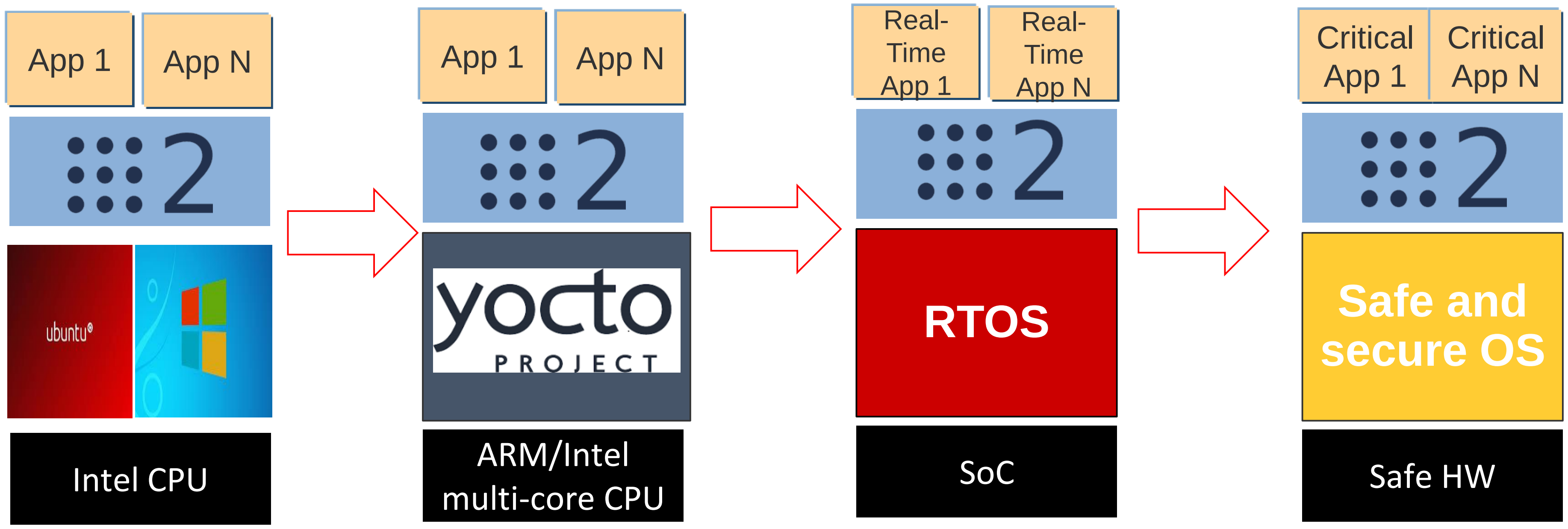
A journey from Ubuntu desktop to the embedded Linux e.g. Yocto is the rocky one:

- Cross compilation
- Complicated Build system
- etc

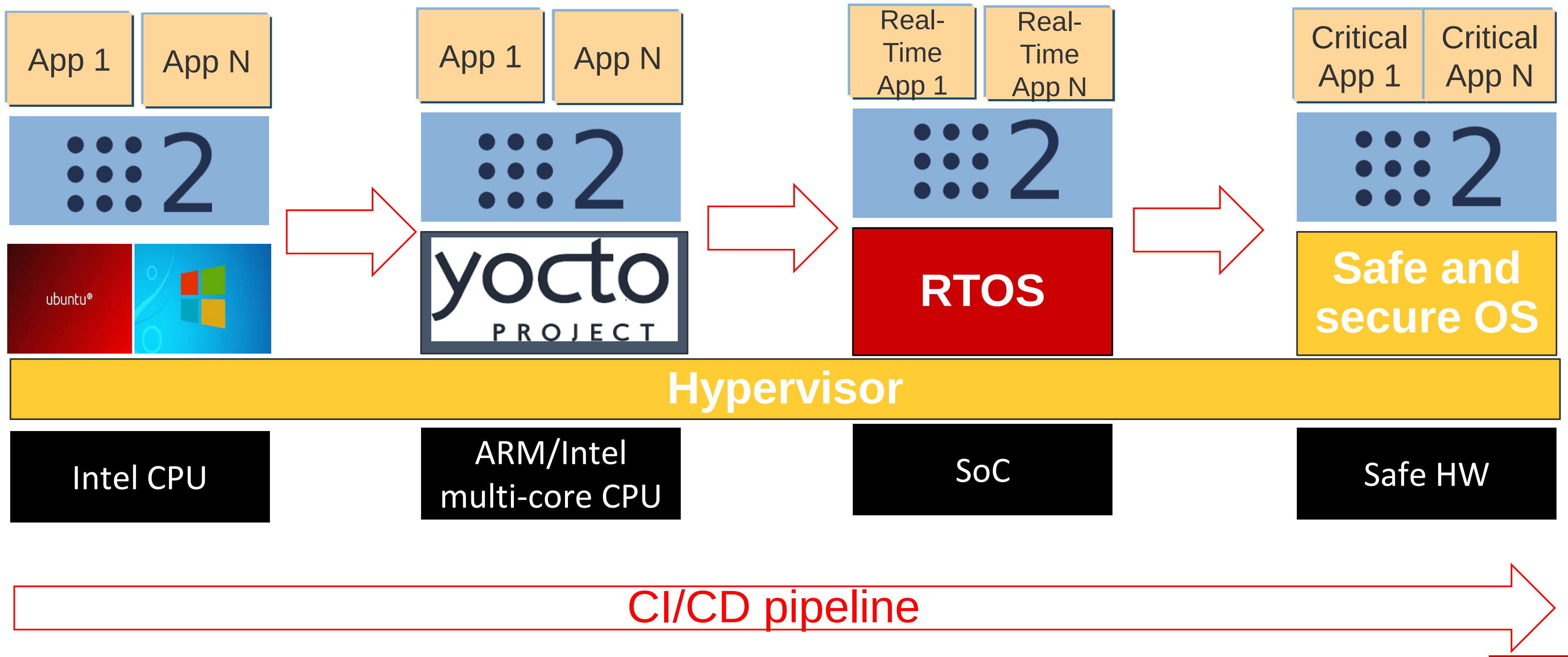
ROS2 Developer Journey embedded real-time



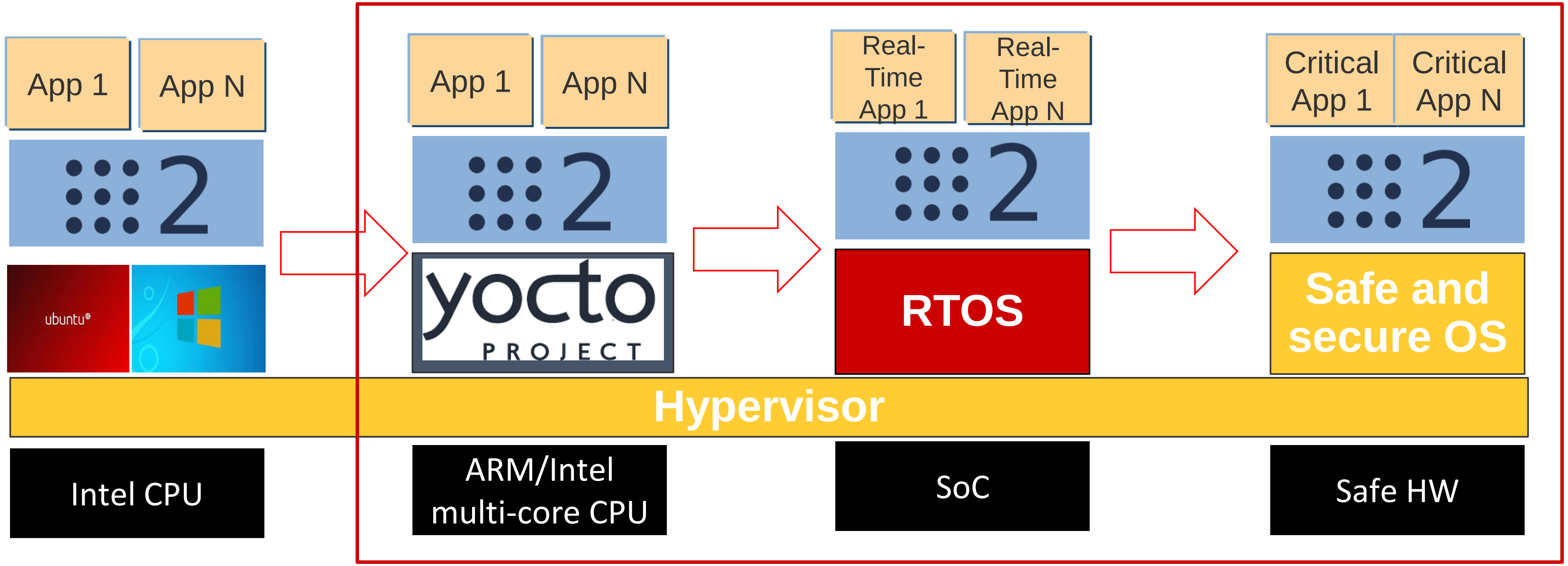
ROS2 Developer Journey embedded real-time, safe and secure



ROS2 Developer Journey hypervisor

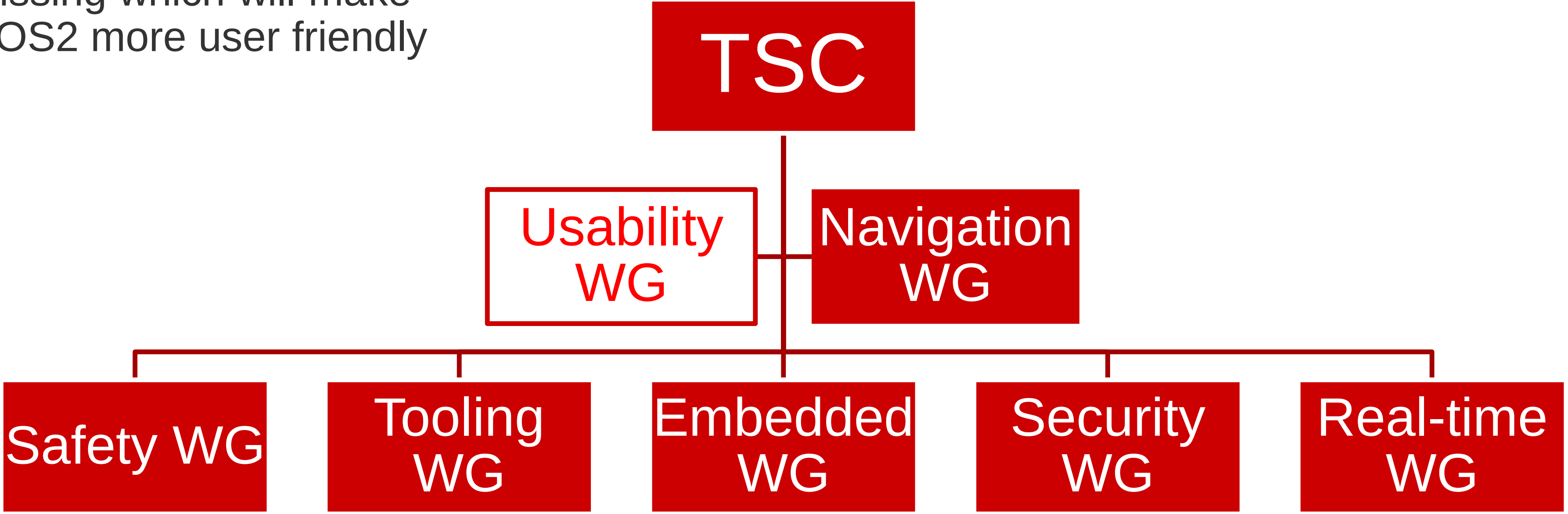


ROS2 Developer Journey
where is the magic button to optimize a run-time behaviour?



ROS2 Working Groups

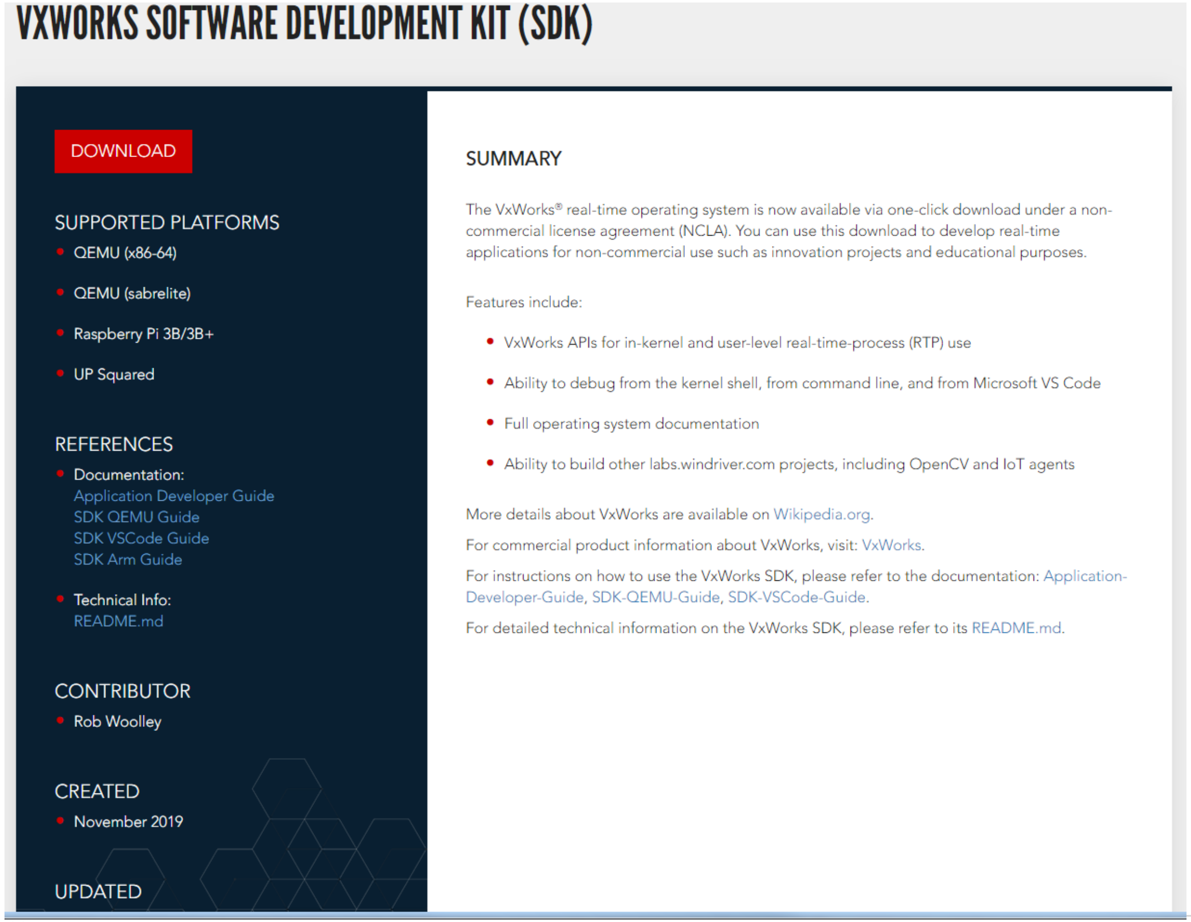
In my view a usability WG is missing which will make ROS2 more user friendly



WHEN IT MATTERS, IT RUNS ON WIND RIVER.

https://labs.windriver.com
Platform for Innovation

https://labs.windriver.com/vxworks-sdk



VXWORKS SOFTWARE DEVELOPMENT KIT (SDK)

DOWNLOAD

SUPPORTED PLATFORMS
- QEMU (x86-64)
- QEMU (sabrelite)
- Raspberry Pi 3B/3B+
- UP Squared

REFERENCES
- Documentation:
  Application Developer Guide
  SDK QEMU Guide
  SDK VSCode Guide
  SDK Arm Guide
- Technical Info:
  README.md

CONTRIBUTOR
- Rob Woolley

CREATED
- November 2019

UPDATED

SUMMARY

The VxWorks® real-time operating system is now available via one-click download under a non-commercial license agreement (NCLA). You can use this download to develop real-time applications for non-commercial use such as innovation projects and educational purposes.

Features include:

- VxWorks APIs for in-kernel and user-level real-time-process (RTP) use
- Ability to debug from the kernel shell, from command line, and from Microsoft VS Code
- Full operating system documentation
- Ability to build other labs.windriver.com projects, including OpenCV and IoT agents

More details about VxWorks are available on Wikipedia.org.

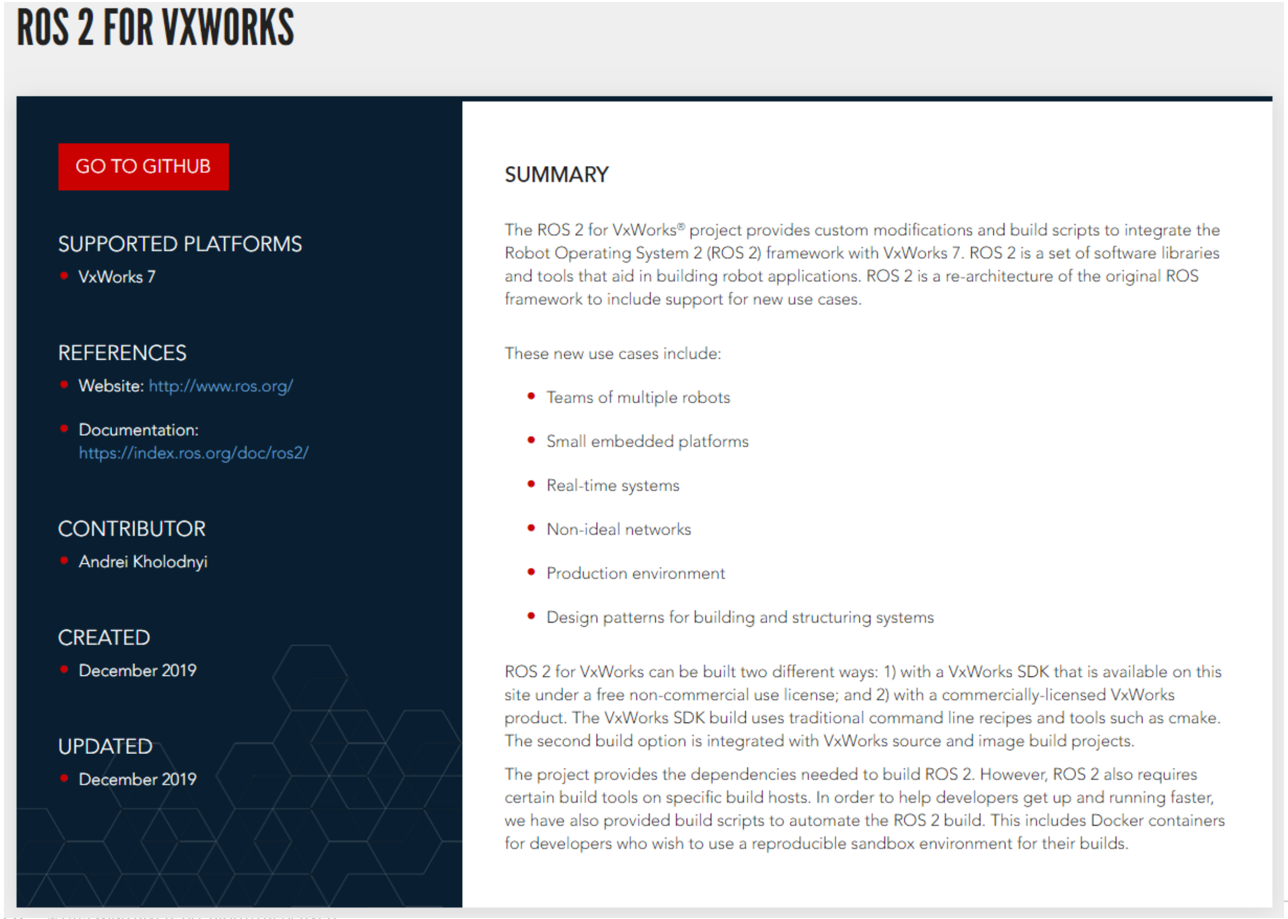
For commercial product information about VxWorks, visit: VxWorks.

For instructions on how to use the VxWorks SDK, please refer to the documentation: Application-Developer-Guide, SDK-QEMU-Guide, SDK-VSCode-Guide.

For detailed technical information on the VxWorks SDK, please refer to its README.md.

For the first time Wind River provide a downloadable SDK for the non-commercial usage

https://labs.windriver.com/ros2-for-vxworks



ROS 2 FOR VXWORKS

GO TO GITHUB

SUPPORTED PLATFORMS

- VxWorks 7

REFERENCES

- Website: http://www.ros.org/
- Documentation: https://index.ros.org/doc/ros2/

CONTRIBUTOR

- Andrei Kholodnyi

CREATED

- December 2019

UPDATED

- December 2019

SUMMARY

The ROS 2 for VxWorks® project provides custom modifications and build scripts to integrate the Robot Operating System 2 (ROS 2) framework with VxWorks 7. ROS 2 is a set of software libraries and tools that aid in building robot applications. ROS 2 is a re-architecture of the original ROS framework to include support for new use cases.

These new use cases include:

- Teams of multiple robots
- Small embedded platforms
- Real-time systems
- Non-ideal networks
- Production environment
- Design patterns for building and structuring systems

ROS 2 for VxWorks can be built two different ways: 1) with a VxWorks SDK that is available on this site under a free non-commercial use license; and 2) with a commercially-licensed VxWorks product. The VxWorks SDK build uses traditional command line recipes and tools such as cmake. The second build option is integrated with VxWorks source and image build projects.

The project provides the dependencies needed to build ROS 2. However, ROS 2 also requires certain build tools on specific build hosts. In order to help developers get up and running faster, we have also provided build scripts to automate the ROS 2 build. This includes Docker containers for developers who wish to use a reproducible sandbox environment for their builds.

ROS2 is built on top of the VxWorks SDK

Developers can deploy and run VxWorks on ARM and Intel

ROS2 Dashing Release VxWorks Port

ROS 2 Apps

ROS 2 VxWorks SDK

ROS 2 dependencies: ASIO, tinyxml2, OpenCV

Python 3.8 | POSIX

Cmake / autotools build primitives

LLVM C++11/C++14

VxWorks SR620

Intel 64-bit / Arm / QEMU

- Complete ROS 2 Dashing release has been ported to VxWorks
- Build using colcon, the same look and feel as a native ROS 2 build (command line)
- OpenCV integration
- Python (ported, not tested)
- Only graphical packages (like RViz) are not ported and stay on Ubuntu

based on the ROS 2 dashing release

approx. 200 ROS 2 packages

OSS_BUILD layer
UNIX_EXTRA layer

https://raw.githubusercontent.com/ros2/ros2/release-latest/ros2.repos

VXWORKS7-ROS2-BUILD (A Helper Repo)

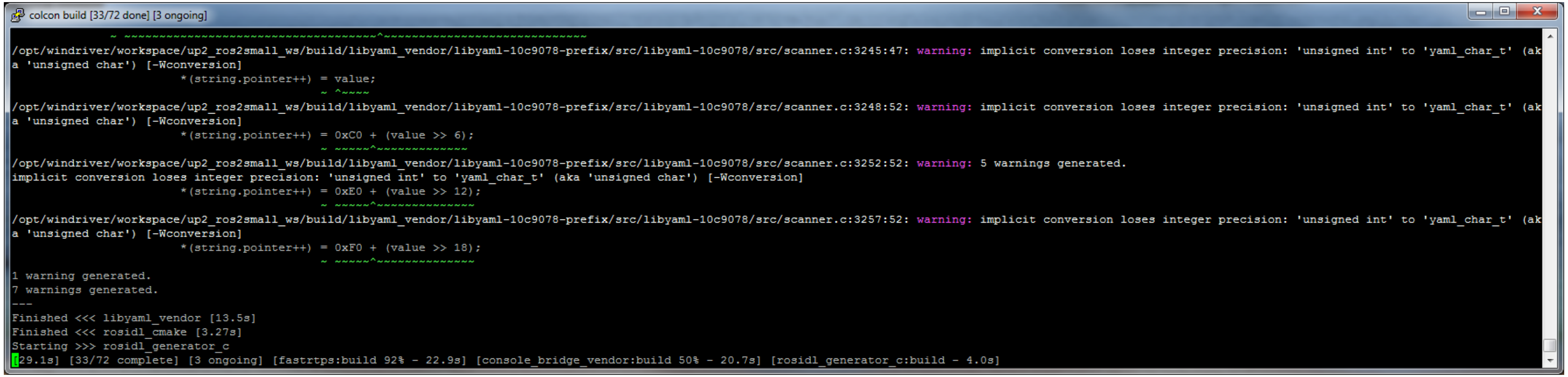
- https://github.com/Wind-River/vxworks7-ros2-build
- Based on the VxWorks SDK
- Download the SDK
- Setup the development environment and do: make
- ROS2 middleware and dependencies will be built
- Board support:
  - RPI3, UP2 and QEMU
  - RPI4 and others to come
- Docker build:
  - VxWorks SDK

VXWORKS7-LAYER-FOR-ROS2 (VxWorks patches)

- https://github.com/Wind-River/vxworks7-layer-for-ros2
- ROS 2 dependencies patches:
  - ASIO, tinyxml2
- ROS 2 patches:
  - fastcdr, fastrtps, rcl, rclutils, etc.

ROS2 Build Under VxWorks

- From the command line (ROS 2 native build)
  - colcon build --symlink-install --cmake-force-configure --cmake-args -DBUILD_TESTING=OFF
- The same look and feel as a ROS 2 native build
  - source $SDK_PATH/toolkit/wind_sdk_env.linux
  - colcon build --symlink-install --cmake-force-configure --cmake-args -DCMAKE_TOOLCHAIN_FILE=$(CMAKE_MODULE_DIR)/buildspecs/cmake/rtp.cmake -DCMAKE_PREFIX_PATH=$PRJ_WS/install;$EXPORT_DIR/usr/root -DBUILD_TESTING=OFF

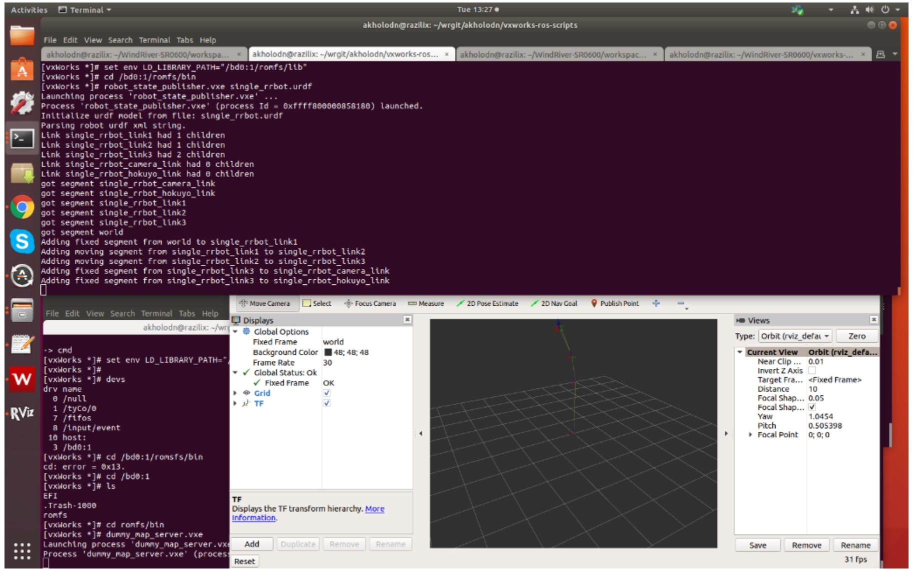


colcon build [33/72 done] [3 ongoing]

```
               ~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3245:47: warning: implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (ak
a 'unsigned char') [-Wconversion]
                         *(string.pointer++) = value;
                                             ~ ^~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3248:52: warning: implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (ak
a 'unsigned char') [-Wconversion]
                         *(string.pointer++) = 0xC0 + (value >> 6);
                                             ~ ~~~~~^~~~~~~~~~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3252:52: warning: 5 warnings generated.
implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (aka 'unsigned char') [-Wconversion]
                         *(string.pointer++) = 0xE0 + (value >> 12);
                                             ~ ~~~~~^~~~~~~~~~~~~~
/opt/windriver/workspace/up2_ros2small_ws/build/libyaml_vendor/libyaml-10c9078-prefix/src/libyaml-10c9078/src/scanner.c:3257:52: warning: implicit conversion loses integer precision: 'unsigned int' to 'yaml_char_t' (ak
a 'unsigned char') [-Wconversion]
                         *(string.pointer++) = 0xF0 + (value >> 18);
                                             ~ ~~~~~^~~~~~~~~~~~~~

1 warning generated.
7 warnings generated.
---
Finished <<< libyaml_vendor [13.5s]
Finished <<< rosidl_cmake [3.27s]
Starting >>> rosidl_generator_c
[29.1s] [33/72 complete] [3 ongoing] [fastrtps:build 92% - 22.9s] [console_bridge_vendor:build 50% - 20.7s] [rosidl_generator_c:build - 4.0s]
```

TURTLEBOT3 Support will come soon


Activities
Terminal
Tue 13:27
akholodn@razilix: ~/wrgit/akholodn/vxworks-ros-scripts
File Edit View Search Terminal Tabs Help
akholodn@razilix: ~/WindRiver-SR0600/workspa...
akholodn@razilix: ~/wrgit/akholodn/vxworks-ros...
akholodn@razilix: ~/WindRiver-SR0600/workspac...
akholodn@razilix: ~/WindRiver-SR0600/vxworks-...

```
[vxWorks *]# set env LD_LIBRARY_PATH="/bd0:1/romfs/lib"
[vxWorks *]# cd /bd0:1/romfs/bin
[vxWorks *]# robot_state_publisher.vxe single_rrbot.urdf
Launching process 'robot_state_publisher.vxe' ...
Process 'robot_state_publisher.vxe' (process Id = 0xffff800000858180) launched.
Initialize urdf model from file: single_rrbot.urdf
Parsing robot urdf xml string.
Link single_rrbot_link1 had 1 children
Link single_rrbot_link2 had 1 children
Link single_rrbot_link3 had 2 children
Link single_rrbot_camera_link had 0 children
Link single_rrbot_hokuyo_link had 0 children
got segment single_rrbot_camera_link
got segment single_rrbot_hokuyo_link
got segment single_rrbot_link1
got segment single_rrbot_link2
got segment single_rrbot_link3
got segment world
Adding fixed segment from world to single_rrbot_link1
Adding moving segment from single_rrbot_link1 to single_rrbot_link2
Adding moving segment from single_rrbot_link2 to single_rrbot_link3
Adding fixed segment from single_rrbot_link3 to single_rrbot_camera_link
Adding fixed segment from single_rrbot_link3 to single_rrbot_hokuyo_link
```

```
-> cmd
[vxWorks *]# set env LD_LIBRARY_PATH="
[vxWorks *]#
[vxWorks *]# devs
drv name
  0 /null
  1 /tyCo/0
  7 /fifos
  8 /input/event
 10 host:
  3 /bd0:1
[vxWorks *]# cd /bd0:1/romsfs/bin
cd: error = 0x13.
[vxWorks *]# cd /bd0:1
[vxWorks *]# ls
EFI
.Trash-1000
romfs
[vxWorks *]# cd romfs/bin
[vxWorks *]# dummy_map_server.vxe
Launching process 'dummy_map_server.vxe
Process 'dummy_map_server.vxe' (process
```

Global Options	
Fixed Frame	world
Background Color	■ 48; 48; 48
Frame Rate	30
✓ Global Status: Ok	
✓ Fixed Frame	OK
Grid	✓
TF	✓

TF
Displays the TF transform hierarchy. More Information.

Type: Orbit (rviz_defau

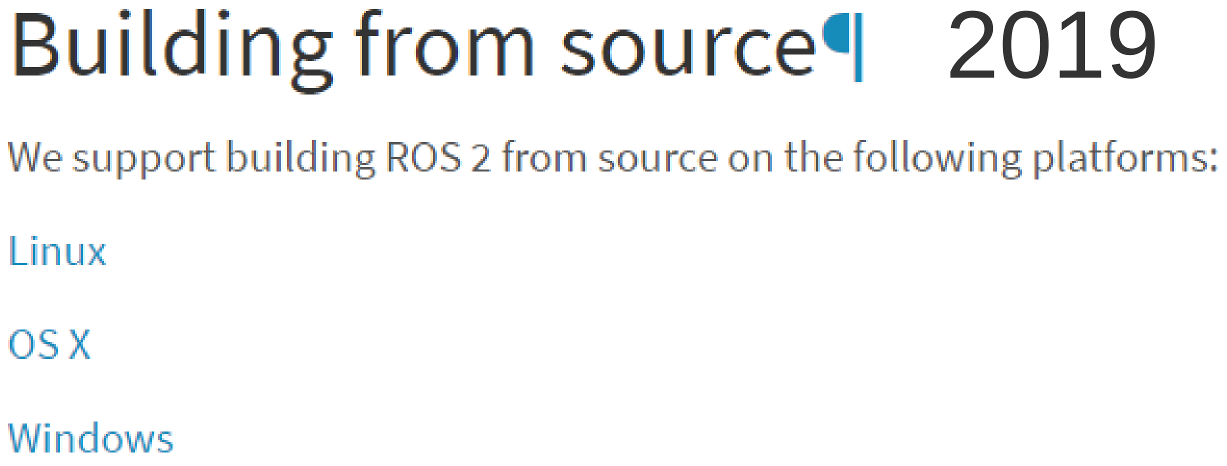
Current View	Orbit (rviz_defa...
Near Clip ...	0.01
Invert Z Axis	☐
Target Fra...	<Fixed Frame>
Distance	10
Focal Shap...	0.05
Focal Shap...	✓
Yaw	1.0454
Pitch	0.505398
Focal Point	0; 0; 0

31 fps

Wind River contribution

- VxWorks, WR Linux Yocto, StarlingX, Hypervisor, TSN, Zypher
- VxWorks SDK published on WR Labs
- ROS2 SDK on VxWorks and WRLinux
  - ARM (RPI4), Intel, QEMU
  - DevOps (containerized)
- VxWorks SDK preintegrated with Eclipse IDE and VSCode
- Participation in Real-time WG, Safety WG and Embedded WG

- Find us on github - https://github.com/Wind-River/vxworks7-ros2-build



Building from source¶

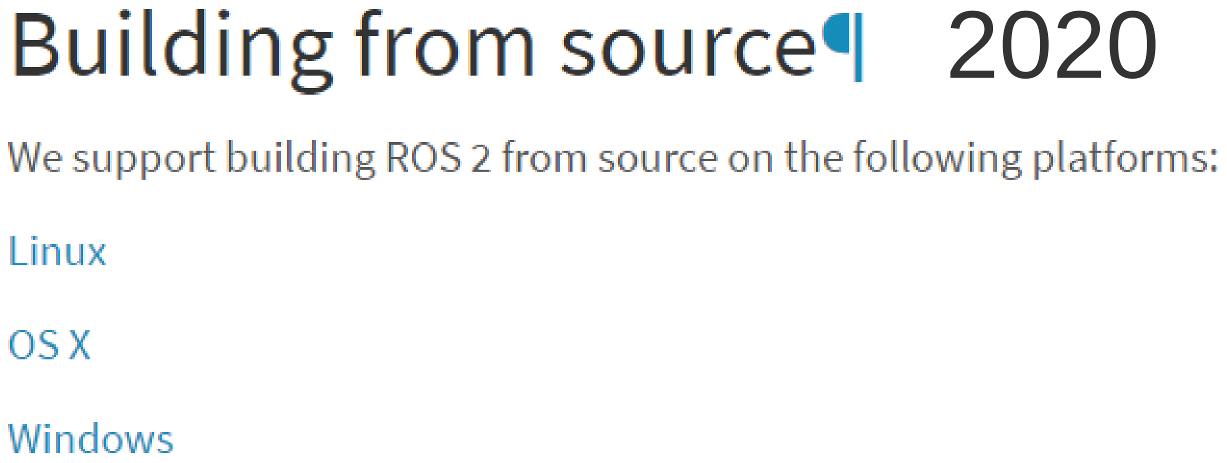
2019

We support building ROS 2 from source on the following platforms:

Linux

OS X

Windows



Building from source¶ 2020

We support building ROS 2 from source on the following platforms:

Linux

OS X

Windows

VxWorks

WRLinux

We plan to add both to the list of officially supported OSes

WIND™

Innovate with us on labs.windriver.com

WHEN IT MATTERS, IT RUNS ON WIND RIVER.