

ROSCon 2018

Integrating ROS and ROS2 on mixed-critical robotic systems based on embedded heterogeneous platforms

Fabio Federici, Giulio M. Mancuso



**United Technologies
Research Center**

Created at UTRC-ITALY

UTC PROPRIETARY - This document contains no USA or EU export controlled technical data.

Background



is a multinational company
operating in different domains
(aerospace, building systems, aircraft engines, ...)



Our
business
units



United Technologies Research Center (UTRC) ensures UTC's technological advantage in the market and solve the toughest scientific challenges for our business unit customers

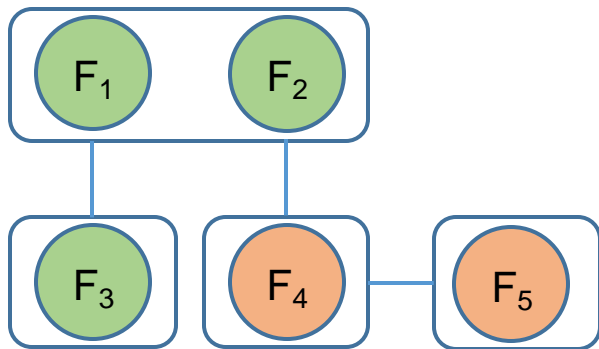
*UTC is not a robot manufacturer, but is a **user & integrator** of robotic and intelligent systems
Main applications are advanced manufacturing, assembly, manipulation, inspection, ...*

Proposal

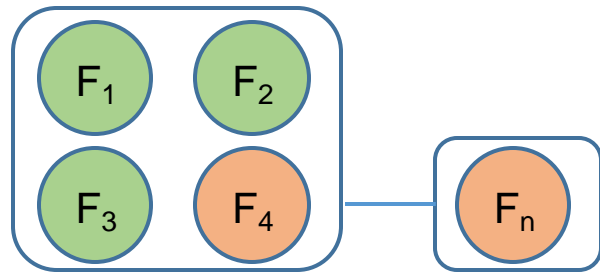
Robotic systems includes **different functions** with a different level of **criticality**. Functions with different criticality are usually allocated on **separate processing units**.

Goal: integration of multiple functions over single, advanced processing units

Pros: reduced size, weight, power, cost
Cons: interference

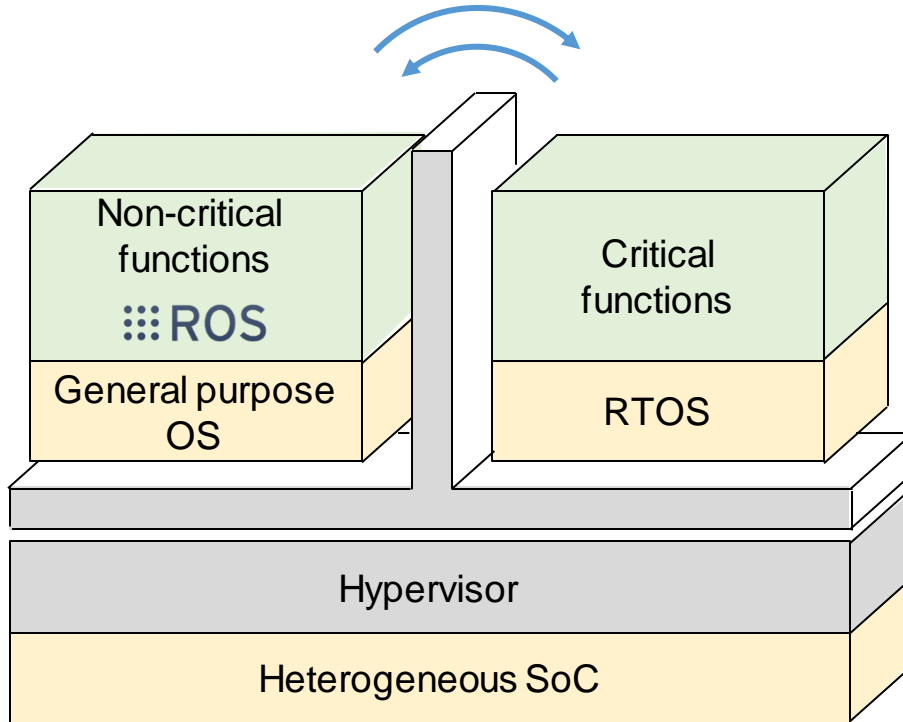


Non mixed-critical scenario



Mixed-critical system

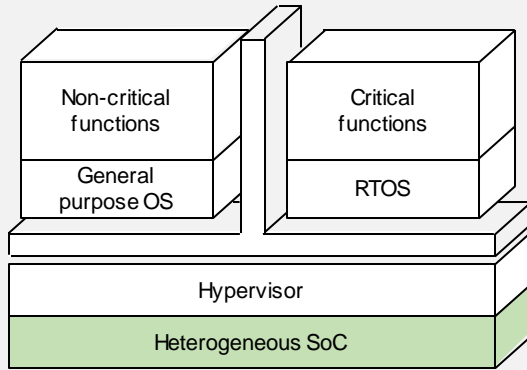
Solution



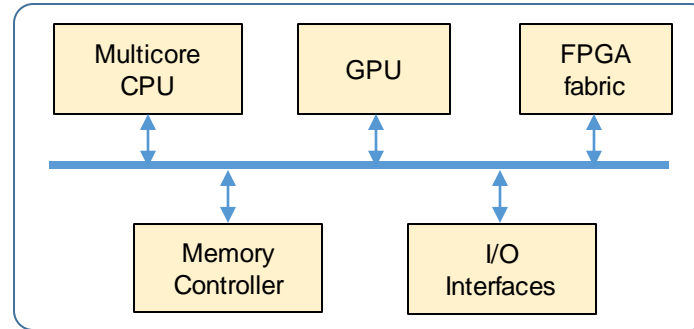
Integration of ROS-based and non-ROS-based application on the same hardware platform

- **Isolation** between different functions (time and space partitioning)
- **Communication** between the different isolated application domains.

Hardware Platforms



COTS heterogeneous devices



Example/Candidates



Nvidia
Jetson TX2



Xilinx Zynq
Ultrascale+

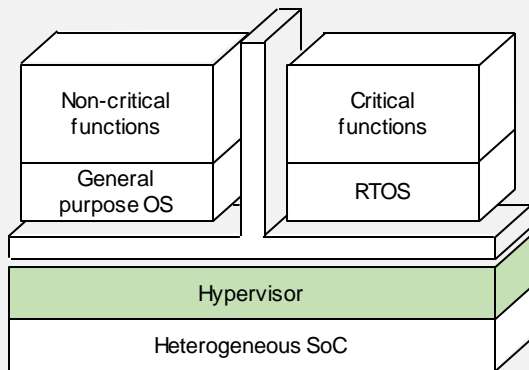
PROS

- Better **SWaP**, lower-cost
- Short time to market

CONS

- Sophisticated (obfuscated) components
- Greater complexity
- Resource sharing potentially jeopardizing **safety**

Hypervisor



Jailhouse:

- Partitioning Hypervisor based on Linux.
 - Able to run bare-metal applications or (adapted) operating systems.
- Originally developed by Siemens
- Released as Free Software (GPLv2) since November 2013



<https://github.com/siemens/jailhouse>

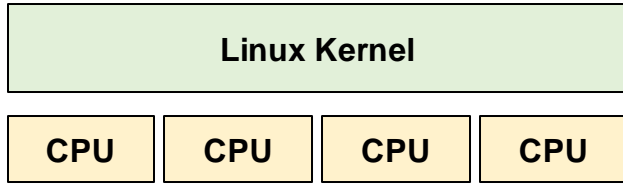
PROS

- Native support for the **Linux** kernel
- Low latencies, good performance
- Open Source (GPL v2)
- Ported on several embedded platforms (Xilinx Zynq, Nvidia Jetson TX1/TX2)

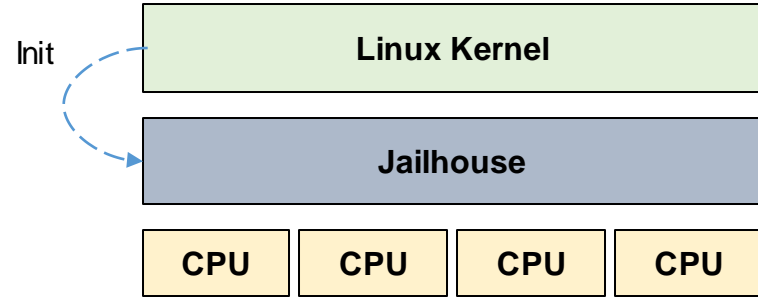
CONS

- System boot depends on the Linux Kernel
- No partition scheduling, only static resource assignment
- Limited **maturity**

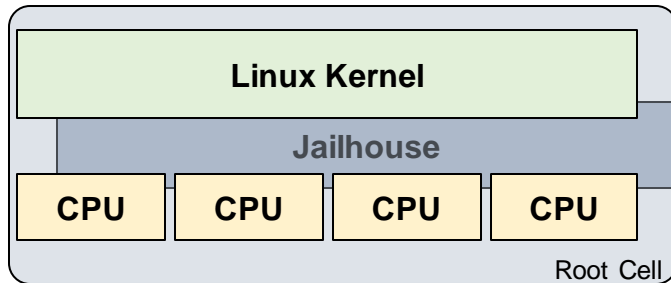
Jailhouse concepts



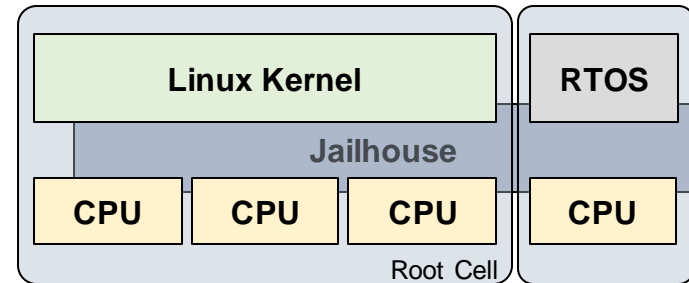
1) Fully booted Linux system



2) Linux loading Jailhouse

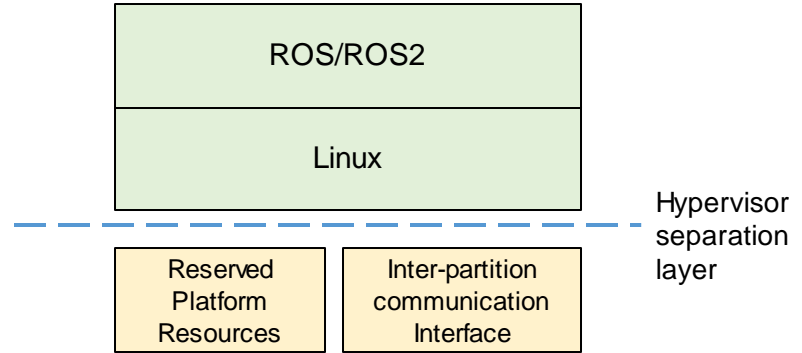
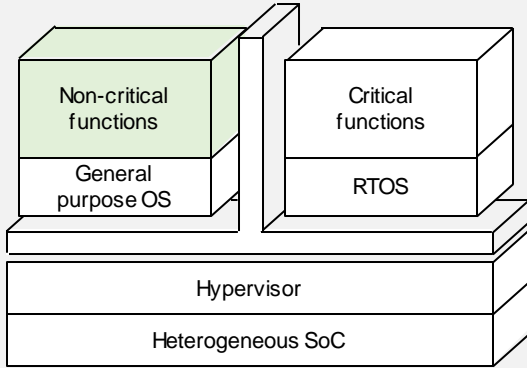


3) Starting the root cell



4) Loading an additional cell

Integrating ROS/ROS 2



	ROS	2
PROS	<ul style="list-style-type: none"> Widely adopted Large community Algorithms, Libraries, Drivers 	<ul style="list-style-type: none"> Determinism Data Distribution Service Security
CONS	<ul style="list-style-type: none"> Lack of determinism Not well fit for safety critical systems 	<ul style="list-style-type: none"> Maturity level Adoption

Ongoing activity & future work

Ongoing: ROS+Jailhouse benchmarking

- Inter-partition interference
- Hypervisor overhead on performance
- Inter-partition communication

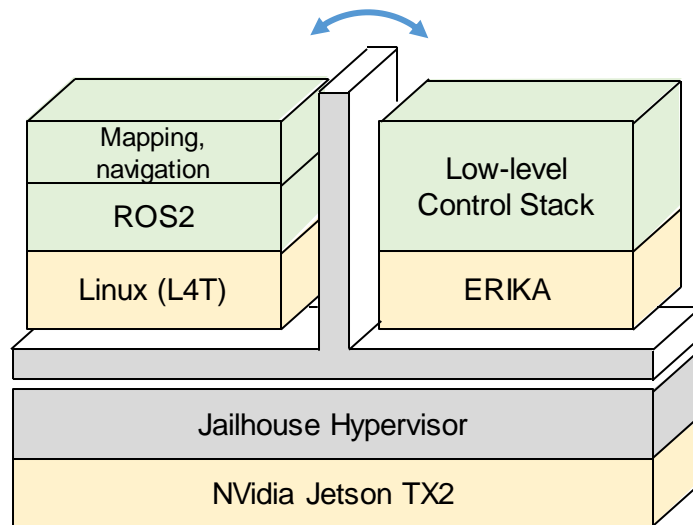
Testing on NVidia Jetson TX2

Communication latency between two nodes in the same partition

	Average Overhead
ROS 1 (Kinetic)	~ +3 %
ROS 2 (Ardent)	~ +5 %

ROS+Linux Vs. ROS+Linux+Jailhouse

Next step: full-stack demonstrator for autonomous UAV



Current collaborations



<https://hipert.unimore.it>

Questions?

fabio.federici@utrc.utc.com
giuliomose.mancuso@utrc.utc.com

<http://www.utrc.utc.com>