Supporting AI-Powered Cyber-Physical Systems on Heterogeneous Platforms

Edoardo Cittadini, Giorgiomaria Cicero, Mauro Marinoni,
Alessandro Biondi and Giorgio Buttazzo

Scuola Superiore Sant'Anna, Pisa, Italy

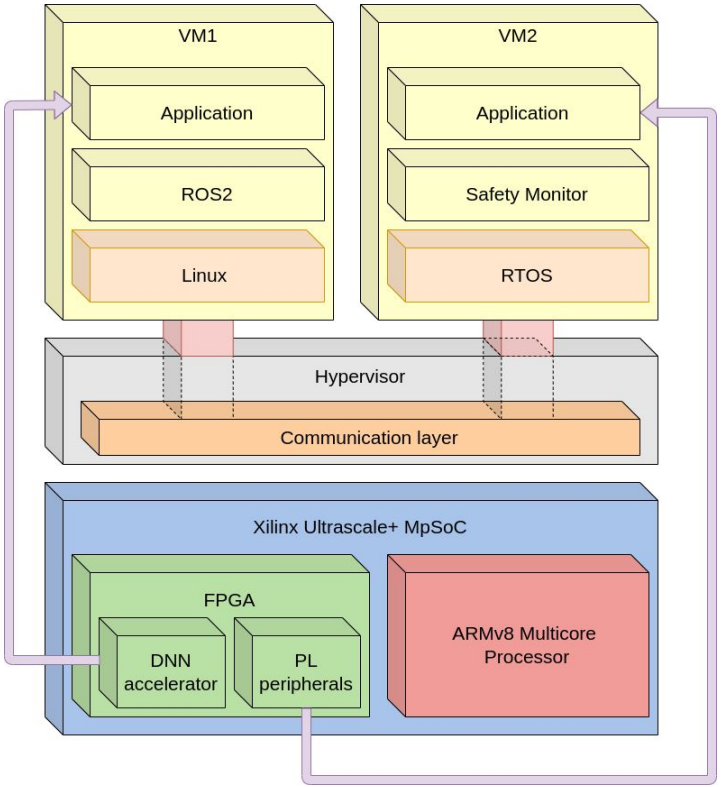
Sant'Anna
Scuola Universitaria Superiore Pisa

Objectives

Supporting modern embedded systems to make them safer and more secure and predictable by a multi-domain architecture

This is achieved by

- Separating Linux-based and real-time subsystems is different domains;
- Using a Type-1 hypervisor to separate them;
- Allowing communication using high speed and reliable channels.



Why multi-domain ?

Linux

- Modern embedded systems have to interface with complex devices such lidars and cameras.
- Most of the libraries with advanced and optimized functionalities to process this kind of data require Linux services support to be executed (i.e. openCV).
- Robotics applications nowadays are mostly based on ROS which is strictly dependent on OS services and version.

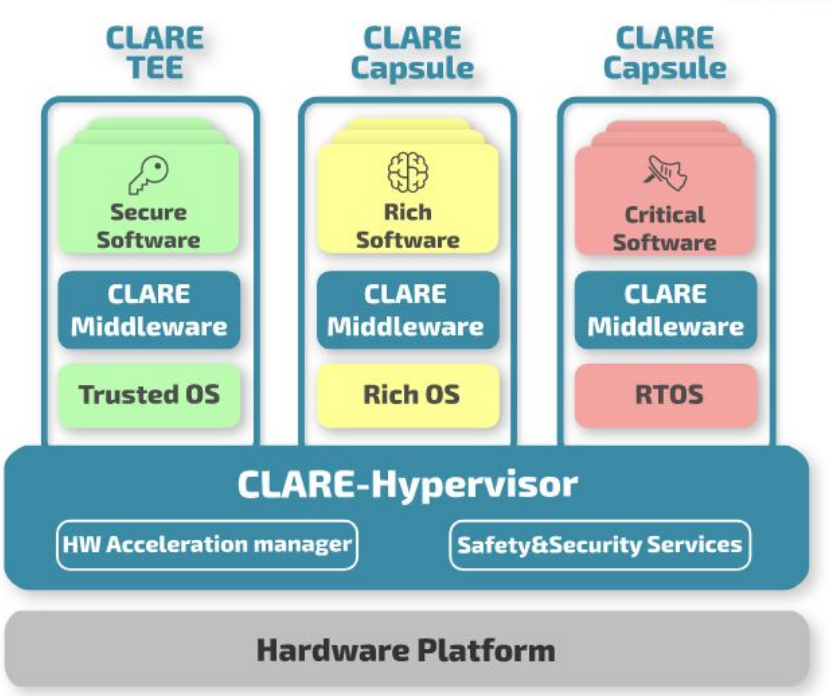
RTOS

- It guarantees time constraints and precise periodic execution of tasks.
- Suitable for wave generation and HW low level interfaces.
- Harder to be remotely attacked.

Hypervisor

CLARE

- Type-1 hypervisor integrating mechanisms to host safe, secure, and time-predictable virtual machines (VMs) that execute in isolation upon the same hardware platform.
- Designed to support modern heterogeneous platforms as **GPGPU** and **FPGA-based SoCs**.
- It virtualizes the FPGA area allowing multiple domains using hardware accelerators in isolation.
- Reference: https://accelerat.eu/clare



FPGA acceleration

Neural network optimization: Accelerate the runtime of complex neural models to reach a higher system throughput.

Custom peripherals: Implement custom hardware in order to meet system requirements or increase system predictability.

Advantages

1) FPGA accelerated models are very power efficient.
2) It reduces size, weight, power, and costs (SWaP-C).
3) It allows increasing system time predictability.

Disadvantages

1) Poor integration with development toolchains (e.g., AI frameworks).
2) Harder implementation and enforced neural network quantization.

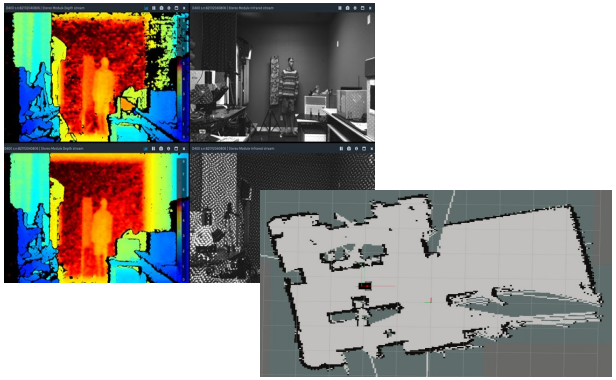
Application 1 - Rover

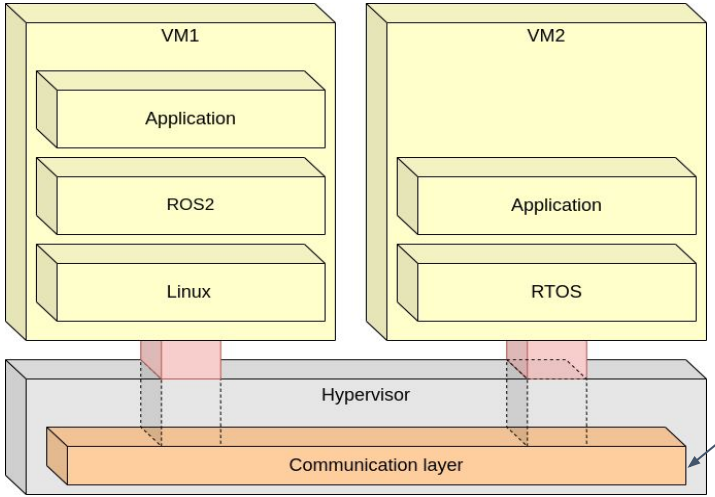
Linux domain

- Ubuntu 20.04
- ROS2 Foxy
- Process camera/lidar
- Generate commands for the RTOS

RTOS domain

- FreeRTOS 10
- Motor actuation
- Safety features
- Linux runtime fault handler





Safety feature

- Temporal information on data sent from Linux.
- Linux crash detection and safe system isolation.

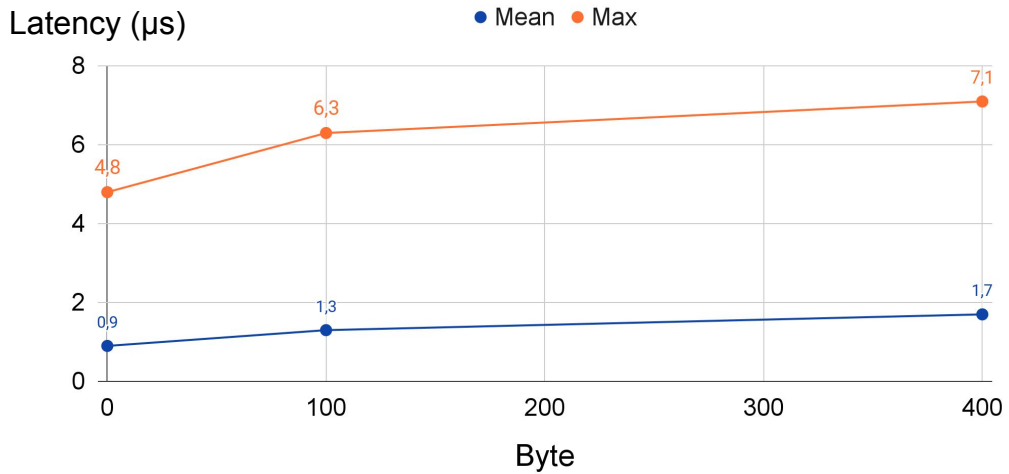
Communication channel

- Shared port provided by the hypervisor.
- Accessed as file descriptor using a standard POSIX interface.

Hypercall latency

Shared port

- The inter domain communication ensures spatial isolation among domains and is hence not zero-copy.
- Latency depends on the amount of transferred data
- Data transfer is lock-less so latency increases almost linearly with data size



Application 2 - Agricolture drone

Linux domain

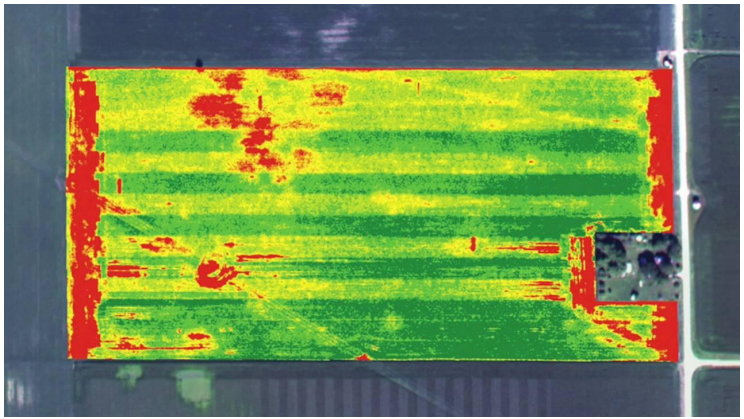
- Ubuntu 20.04
- ROS2 Foxy
- Camera acquisition
- Execute RT NN computations
- Generate commands for RTOS

RTOS domain

- FreeRTOS 10
- Flight controller
- Safety certified features
- Linux runtime fault handler

Communication channel

- Shared memory area
- Very fast and reliable, more appropriate when moving a lot of data



Safety feature

- Temporal information from writes in memory.
- Watchdog timer interrupt and fail system activation.

Conclusions

- We proposed a backbone multi-domain architecture for safe, secure, and predictable heterogeneous embedded systems.
- We implemented two different use cases to show how the same architecture can be applied in very different scenarios.
- We showed that complex high-level software can be integrated without modifying safety properties and reliability of the real-time critical portion of the system.

Future work

- Provide a precise timing analysis to guarantee safety-critical tasks.
- Enhance the support for executing deep neural networks to make them more trustworthy when used in safety-critical applications.

Thank you!

Edoardo Cittadini
edoardo.cittadini@santannapisa.it