METASAT

MODULAR MODEL-BASED DESIGN AND
TESTING FOR APPLICATIONS IN SATELLITES

Perspective on Cloud for Space and Related Developments in the METASAT Project

Dr. Leonidas Kosmidis

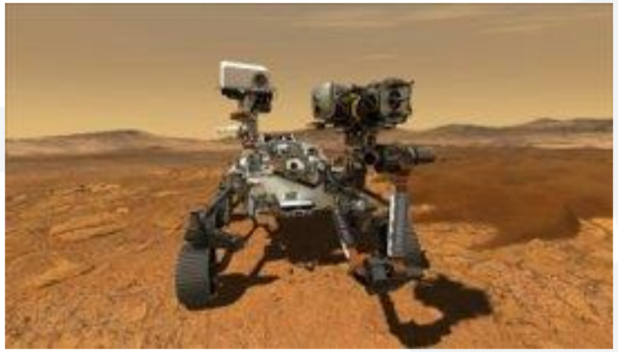
09/06/2023

Outline

- Introduction
- Historic view of Space Computing
- New Space
- The METASAT project
- Conclusion

Introduction

- Modern and upcoming space systems require increasing levels of computing power
- Traditional space processors cannot provide this performance level
- Emerging market trends and requirements similar to cloud computing
  - But with some significant differences



Historic view of Space Computing

- Traditionally the space sector has been accessible only by a few government-funded agencies
- Very high cost of missions
  - Launch cost
  - Hardware cost
  - Software cost
  - Long Term missions (10-20 years)
- Single entity ownership and use

Why Space Hardware is Expensive?

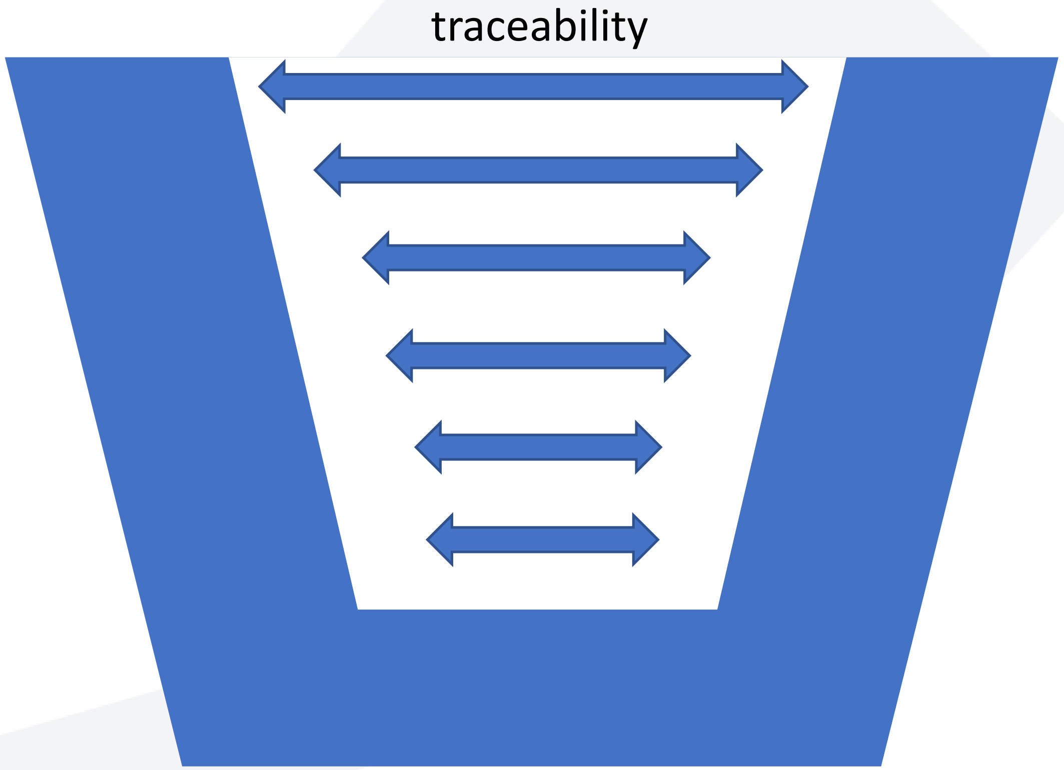
- Space is a harsh environment for electronic systems
- Radiation effects
  - Single Event Upsets (SEU)
  - Single Event Latch-ups (SEL)
  - Single Event Functional Interrupts (SEFI)
  - Total Ionising Dose (TID)
- Extreme temperatures (65°C to +125°C)
- No possibility of active cooling
- Limited available power
  - Solar panels
- Shock/vibration
- Long term availability: 10-20 years

Why Space Hardware is Expensive?

- Custom designs for space
  - Radiation hardened
    - Node Level: Radiation hardened cells, older/larger manufacturing technologies
    - Circuit/Architecture Level: Triplication, Error Correction Codes, Parity
    - Less die area from contemporary consumer electronics
    - Simpler Processors: in-order processing, shallow memory hierarchies, low number of cores
  - Radiation Tolerant
    - Consumer electronics chips screened for space use
    - Undergo extreme tests
    - Need full traceability
    - Simpler Processors: the more complex, the more difficult to test and higher probability of failure
  - When high performance is needed custom FPGA designs are used
    - Usually unique per mission
  - Custom interfaces: SpaceWire, SpaceFiber etc
- Very low volume

Why Space Software is Expensive?

- Needs to be qualified according to stringent safety standards e.g. ECSS
- Different categories depending on the software criticality
- MC/DC Coverage
- V-development model
  - Requirements
  - Verification
  - Traceability
- Linux cannot be qualified for institutional missions



Recent Changes in the Space Sector

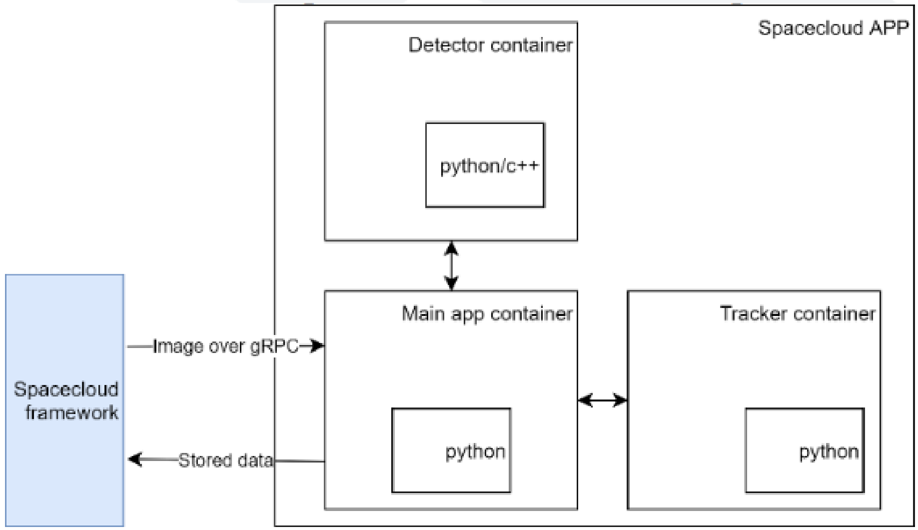
- Increased involvement of private companies in space
- Lower launch cost
  - Reusable rockets
  - Smaller satellites
    - Nanosats, cubesats, satellite constellations
- Lower hardware cost
  - Use of Commercial-off the Shelf (COTS) processors
  - Use of automotive-grade processors
- Lower software cost
  - Use of COTS software
- Smaller missions
  - Low Earth Orbit
  - Short duration

New Space

- Affordable access to space
- Software-defined Satellites
- Satellite-as-a-service
  - Single owner, multiple users
  - Satellite software is no longer known at launch time
    - The architecture needs to be generic and powerful enough
  - Multiple modes of operation
    - Periodic Time sharing
    - Slot Reservation
    - Parallel execution provided different instruments are used by each user

Example of Existing Space Cloud Service

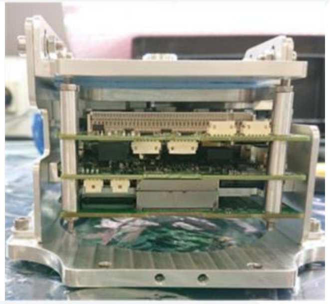
- Unibap's SpaceCloud® Framework [1]
- Container-based solution
  - Linux, Docker
- Instrument Access API
- Access to GPU, AI accelerators
  - TensorFlow, OpenVINO etc
- Redundant boot mechanism
- Reliability monitoring system
- Radiation tolerant middleware
- Future Integration with AWS



[1] O. Flordal et al, SpaceCloud Cloud Computing and In-Orbit Demonstration, European Workshop on On-Board Data Processing (OBDP), 2021, https://doi.org/10.5281/zenodo.5522872

Example of Existing Space Cloud Service

- Radiation Tolerant High Performance Hardware
- Unibap iX5-100
  - AMD V1605B Embedded Ryzen
  - 4 dual threaded x86 Zen CPUs, AMD Vega GPU with 8 CUs
  - 12-25W TDP
- In-orbit demonstrator in 2021



[1] O. Flordal et al, SpaceCloud Cloud Computing and In-Orbit Demonstration, European Workshop on On-Board Data Processing (OBDP), 2021, https://doi.org/10.5281/zenodo.5522872

Example of Existing Space Cloud Service

- Example applications
    - Near-real time mid-air aircraft detection
    - Image and video compression
    - Car tracking
    - Volcanic activity
    - Forest Fires
    - Precision Agriculture
    - Oil Spill Detection

[1] O. Flordal et al, SpaceCloud Cloud Computing and In-Orbit Demonstration, European Workshop on On-Board Data Processing (OBDP), 2021, https://doi.org/10.5281/zenodo.5522872

Example of Existing Space Cloud Service

- Similar in-orbit demonstrators from other companies: (non-exhaustive list)
  - FPGA SoCs with ARM CPUs
    - Xilinx Ultrascale+
    - Intel/Altera Cyclone V
  - NVIDIA based GPU platforms
    - NVIDIA TX2, NVIDIA Xavier
  - AI/Vision Accelerators
    - Intel/Movidius Myriad
- Similar software stack based on Linux and Docker

Limitations of existing space cloud solutions

- Use in institutional and mixed criticality missions
  - Dependence on non-qualifiable software stacks
    - Linux, docker, GPU drivers
  - No support for legacy space applications running on space qualified real-time operating systems
  - Security issues
    - Docker provides limited isolation
  - Hardware cannot be qualified for critical missions
    - Radiation tolerance is not enough for complex hardware
    - Enough information about the design internals is required
    - Potential design changes to enhance radiation tolerance and other features
    - Custom designs for space with existing commercial GPU IPs are not viable [1]
    - Short-term product availability for hardware and software
- ITAR and export control restrictions [1]

[1] L. Kosmidis et al, GPU4S: Major Project Outcomes, Lessons Learnt and Way Forward, DATE 2021

Open Hardware to the rescue

- RISC-V has revolutionalised the computer architecture landscape across all domains, from supercomputers to space
- Royalty-free ISA, modular and customizable
- Several open source and closed source implementations
- Unique opportunity for safety critical markets like space to get access to full implementation details and/or customise designs to their needs
- A large ecosystem is under development

The METASAT Horizon Europe Project

- 2-year project: January 2023-December 2024
- TRL 3-4

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

fentISS

Collins Aerospace

ikerlan
MEMBER OF BASQUE RESEARCH
& TECHNOLOGY ALLIANCE

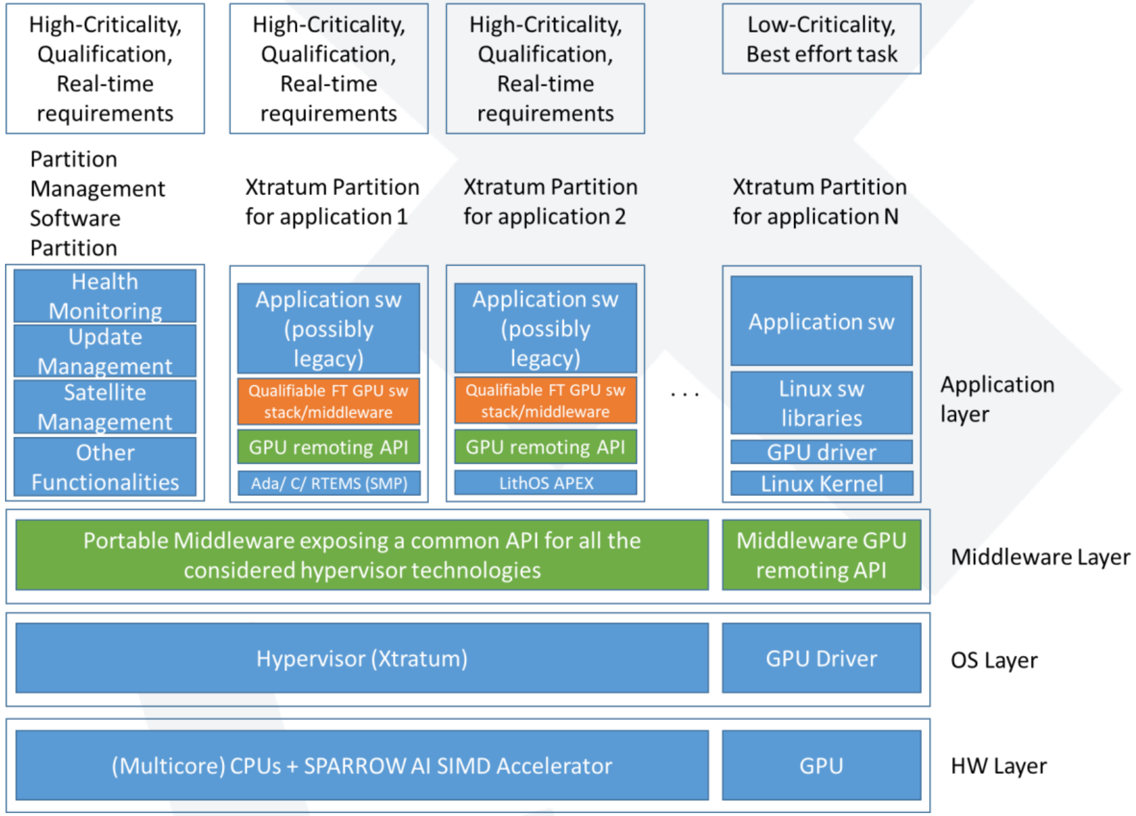
OHB

METASAT Overview

- Manage the complexity of future hardware and software for space
- A prototype High Performance Platform is required
  - Multicore CPUs, AI accelerators, GPU
- Support for Virtualisation
  - Time and Space isolation provide benefits for faster and easier integration
  - Components can be developed and tested in isolation
  - Fault Detection, Isolation and Recovery (FDIR)
  - Qualified Hypervisor for space systems
    - Xtratum
    - Can provide the features required for the implementation of a high criticality space cloud for institutional missions

The METASAT RISC-V Platform

- Mixed Criticality Platform
- FPGA Prototype on a Xilinx VCU118
- Multicore CPU Based on NOEL-V + SPARROW AI SIMD Accelerator
  - Qualifiable software stack for high criticality software with moderate AI acceleration needs



High-Criticality, Qualification, Real-time requirements

High-Criticality, Qualification, Real-time requirements

High-Criticality, Qualification, Real-time requirements

Low-Criticality, Best effort task

Partition Management Software Partition

Health Monitoring

Update Management

Satellite Management

Other Functionalities

Xtratum Partition for application 1

Application sw (possibly legacy)

Qualifiable FT GPU sw stack/middleware

GPU remoting API

Ada/ C/ RTEMS (SMP)

Xtratum Partition for application 2

Application sw (possibly legacy)

Qualifiable FT GPU sw stack/middleware

GPU remoting API

LithOS APEX

. . .

Xtratum Partition for application N

Application sw

Linux sw libraries

GPU driver

Linux Kernel

Application layer

Portable Middleware exposing a common API for all the considered hypervisor technologies

Middleware GPU remoting API

Middleware Layer

Hypervisor (Xtratum)

GPU Driver

OS Layer

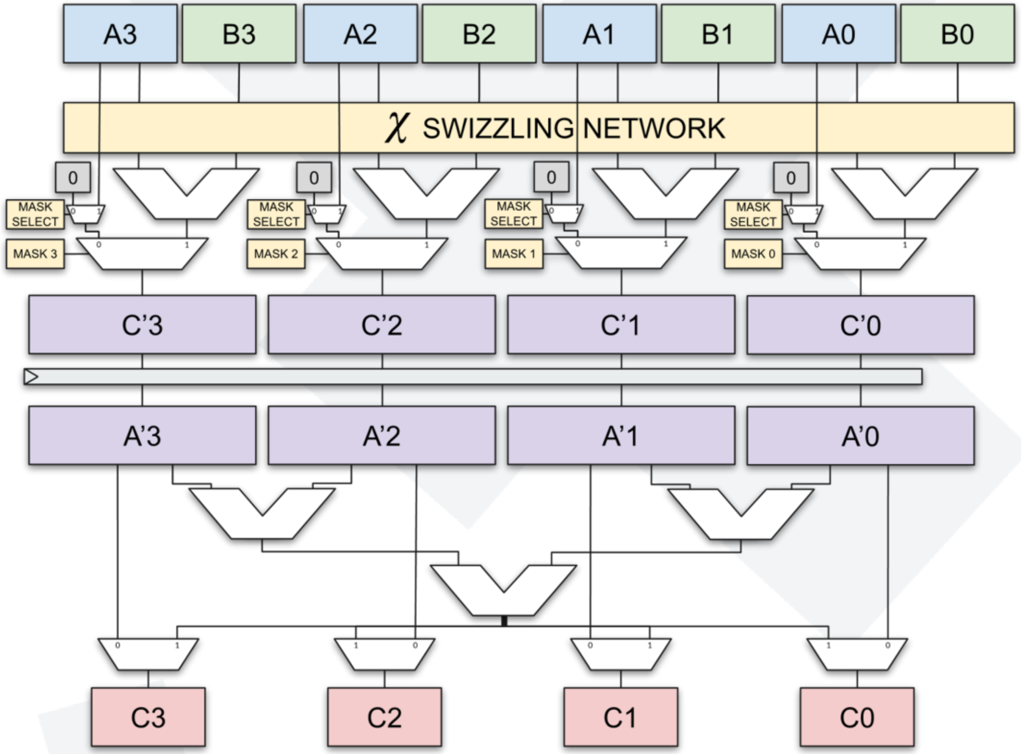
(Multicore) CPUs + SPARROW AI SIMD Accelerator

GPU

HW Layer

The METASAT RISC-V Platform

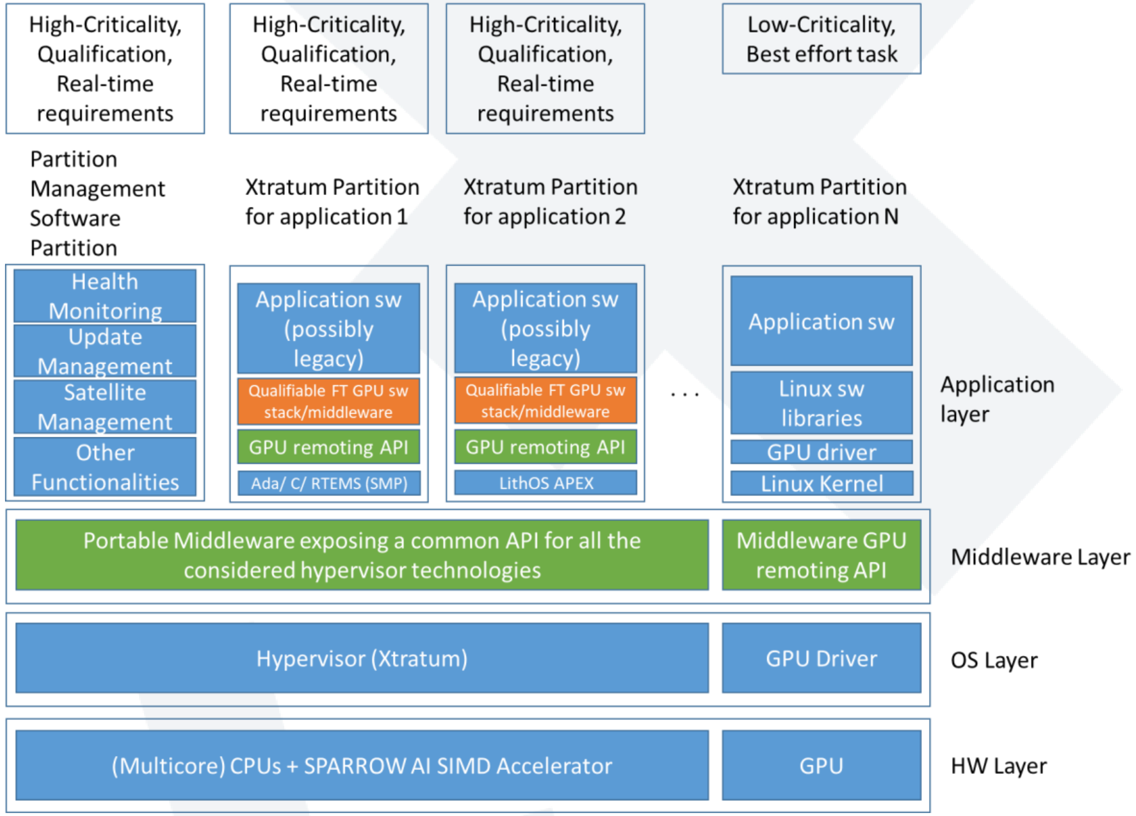
- SPARROW AI SIMD Accelerator
- High-performance, Low-cost at least 30% smaller than conventional vector processors with similar performance
- Minimal core modifications
  - incremental qualification
- Key features: reuse of integer register file, short SIMD unit (8-bit), swizzling, reductions
- Intrinsics-like software support similar to ARM’s NEON



[1] M. Solé, SPARROW: A Low-Cost Hardware/Software Co-designed SIMD Microarchitecture for AI Operations in Space Processors, DATE 2021

The METASAT RISC-V Platform

- Mixed Criticality Platform
- FPGA Prototype on a Xilinx VCU 118
- Configurable Vortex RISC-V GPU [1]
  - Enhancements for real-time execution and reliability
  - Qualifiable software stack for tasks requiring very high performance
  - Enable the use of GPUs from bare metal, or RTOS
  - Share the GPU among partitions
- The hardware platform will be open sourced as well as much of its software



High-Criticality, Qualification, Real-time requirements

High-Criticality, Qualification, Real-time requirements

High-Criticality, Qualification, Real-time requirements

Low-Criticality, Best effort task

Partition Management Software Partition

Xtratum Partition for application 1

Xtratum Partition for application 2

Xtratum Partition for application N

Health Monitoring

Update Management

Satellite Management

Other Functionalities

Application sw (possibly legacy)

Qualifiable FT GPU sw stack/middleware

GPU remoting API

Ada/ C/ RTEMS (SMP)

Application sw (possibly legacy)

Qualifiable FT GPU sw stack/middleware

GPU remoting API

LithOS APEX

. . .

Application sw

Linux sw libraries

GPU driver

Linux Kernel

Application layer

Portable Middleware exposing a common API for all the considered hypervisor technologies

Middleware GPU remoting API

Middleware Layer

Hypervisor (Xtratum)

GPU Driver

OS Layer

(Multicore) CPUs + SPARROW AI SIMD Accelerator

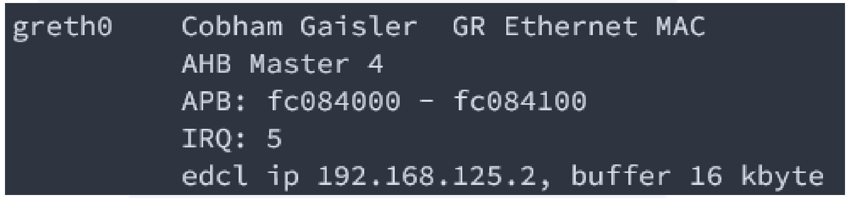
GPU

HW Layer

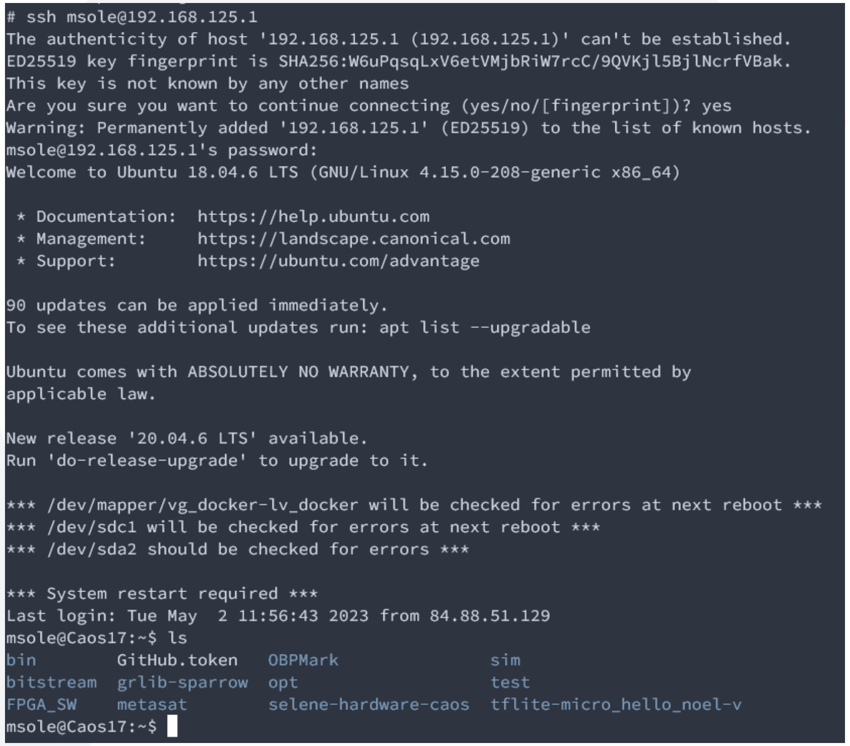
[1] B. Tine et al, Vortex: Extending the RISC-V ISA for GPGPU and 3D-Graphics, MICRO 2021

The METASAT RISC-V Platform

- Mixed Criticality Platform
- FPGA Prototype on a Xilinx VCU 118
- Ethernet connectivity through Gaisler's GRLIB Ethernet controller (greth)
- Currently the driver is getting ported from Linux to Xtratum



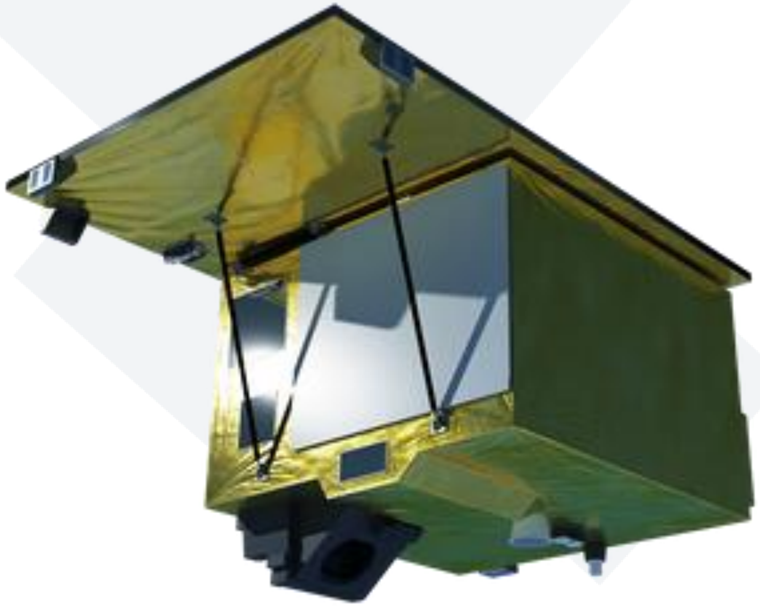
```
greth0      Cobham Gaisler  GR Ethernet MAC
            AHB Master 4
            APB: fc084000 - fc084100
            IRQ: 5
            edcl ip 192.168.125.2, buffer 16 kbyte
```



```
# ssh msole@192.168.125.1
The authenticity of host '192.168.125.1 (192.168.125.1)' can't be established.
ED25519 key fingerprint is SHA256:W6uPqsqLxV6etVMjbRiW7rcC/9QVKjl5BjlNcrfVBak.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.125.1' (ED25519) to the list of known hosts.
msole@192.168.125.1's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-208-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

90 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** /dev/mapper/vg_docker-lv_docker will be checked for errors at next reboot ***
*** /dev/sdc1 will be checked for errors at next reboot ***
*** /dev/sda2 should be checked for errors ***

*** System restart required ***
Last login: Tue May  2 11:56:43 2023 from 84.88.51.129
msole@Caos17:~$ ls
bin        GitHub.token   OBPMark               sim
bitstream  grlib-sparrow  opt                   test
FPGA_SW    metasat        selene-hardware-caos  tflite-micro_hello_noel-v
msole@Caos17:~$ 
```

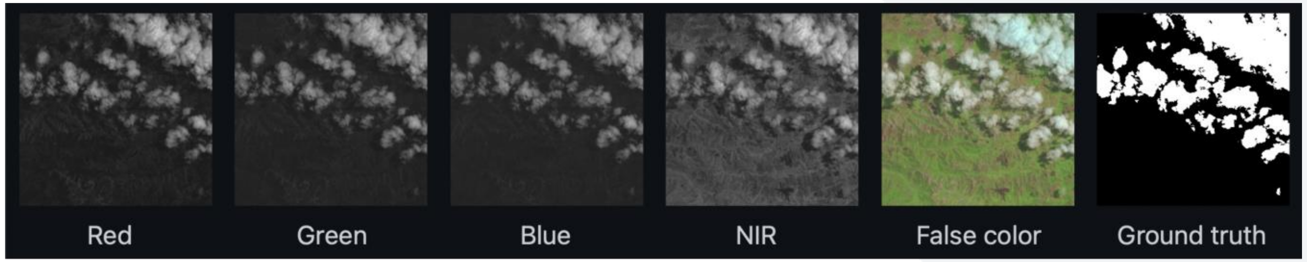
Project Use Cases

- 3 Project Use cases will be implemented
- OHB/DLR Use Case
  - Hardware interlocking
    - Protect against wrong software behaviour
  - Implement interlocks at software level instead of hardware
    - Reduce cost
  - Implement AI Based FDIR
    - To be accelerated on the CPU using the SPARROW AI accelerator
  - Housekeeping data from the ENMAP Satellite



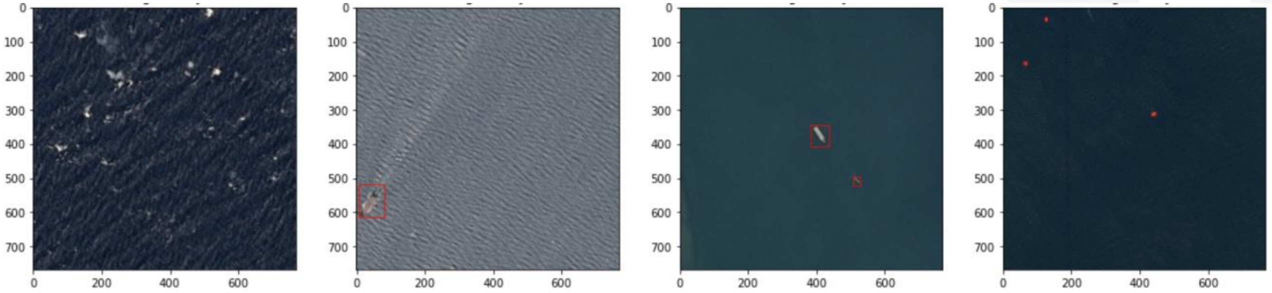
Project Use Cases

- 2 BSC-provided use cases based on OBPMark-ML [1][2]
- Cloud screening



4 Channels RGB/NIR mapped to binary mask (cloud/no cloud)

- Ship Detection



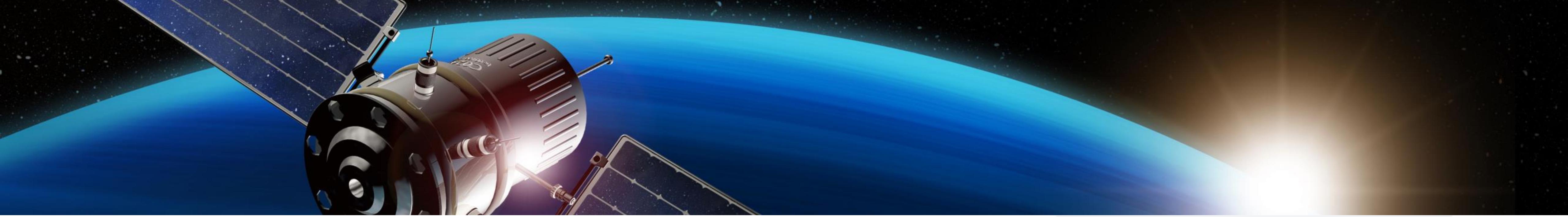
- To be executed on the GPU

[1] D. Steenari et al, OBPMark (On-Board Processing Benchmarks) – Open Source Computational Performance Benchmarks for Space Applications, European Workshop on On-Board Data Processing (OBDP2021). https://doi.org/10.5281/zenodo.5638577

[2] http://obpmark.org

Conclusion

- Cloud space computing is becoming a reality
- Many similarities and some differences with classic cloud computing
  - Software stack
  - Smaller target systems i.e. high performance embedded systems
  - Low power consumption and thermal limits
  - Increased reliability and radiation tolerance
  - Interface with instruments
- New Space approaches are based on Linux and Docker
- Institutional and high criticality missions require qualifiable stacks and high degree of isolation
- The METASAT project with its RISC-V based hardware and software implementations provides some underlying features which contribute to the implementation of a cloud system complying with these requirements



METASAT

https://metasat-project.eu/
info@metasat-project.eu

https://twitter.com/MetasatProject

https://www.linkedin.com/company/metasat-project

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

fentISS
FENT Innovative Software Solutions

Collins Aerospace

ikerlan
MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

OHB

METASAT has received funding from the European Union's Horizon Europe programme under grant agreement number 101082622.