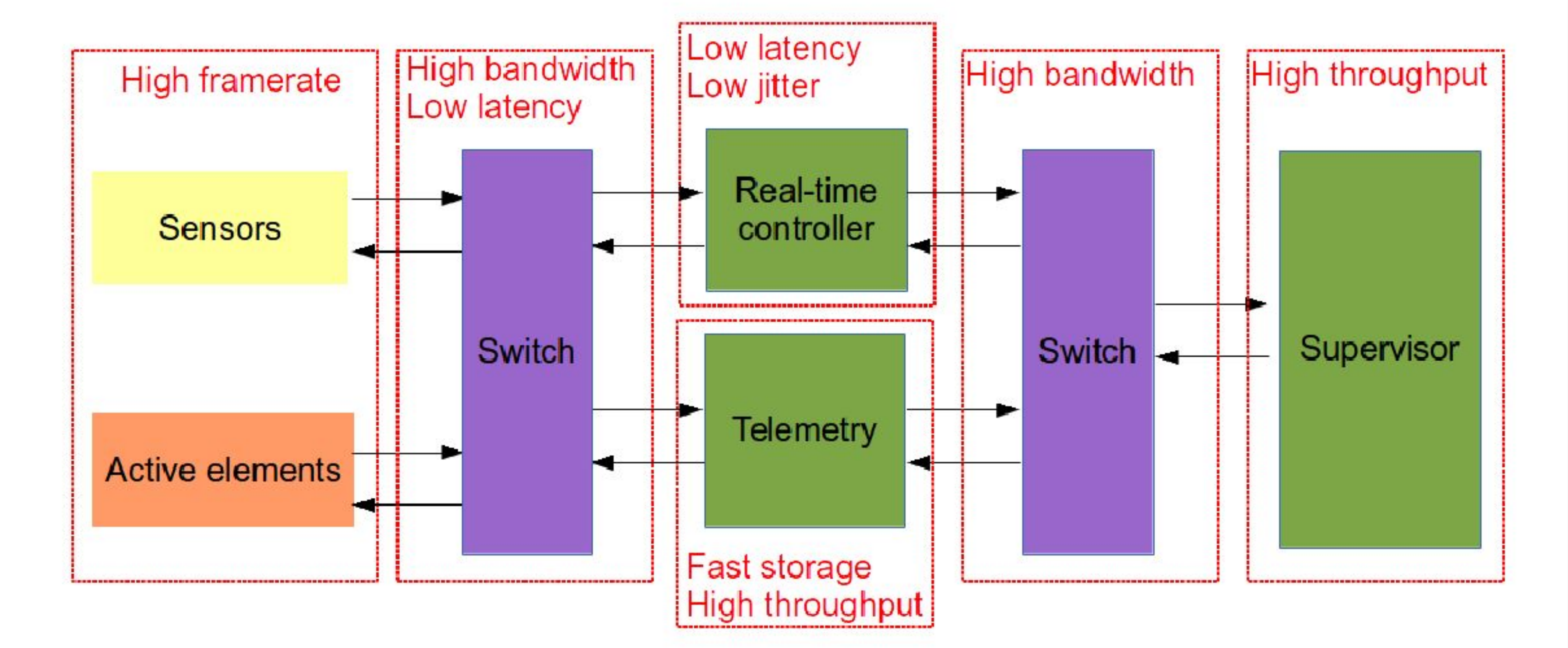
The COSMIC AO RTC platform

Damien Gratadour

AO RTC Global system architecture

Based on heterogeneous architecture to implement main functions

- Cope with various functional & non-functional requirements for the different sub-systems
- Mix high throughput Machine Learning (supervisor, a.k.a. SRTC) with low latency & low jitter computing (real-time controller, a.k.a. HRTC)

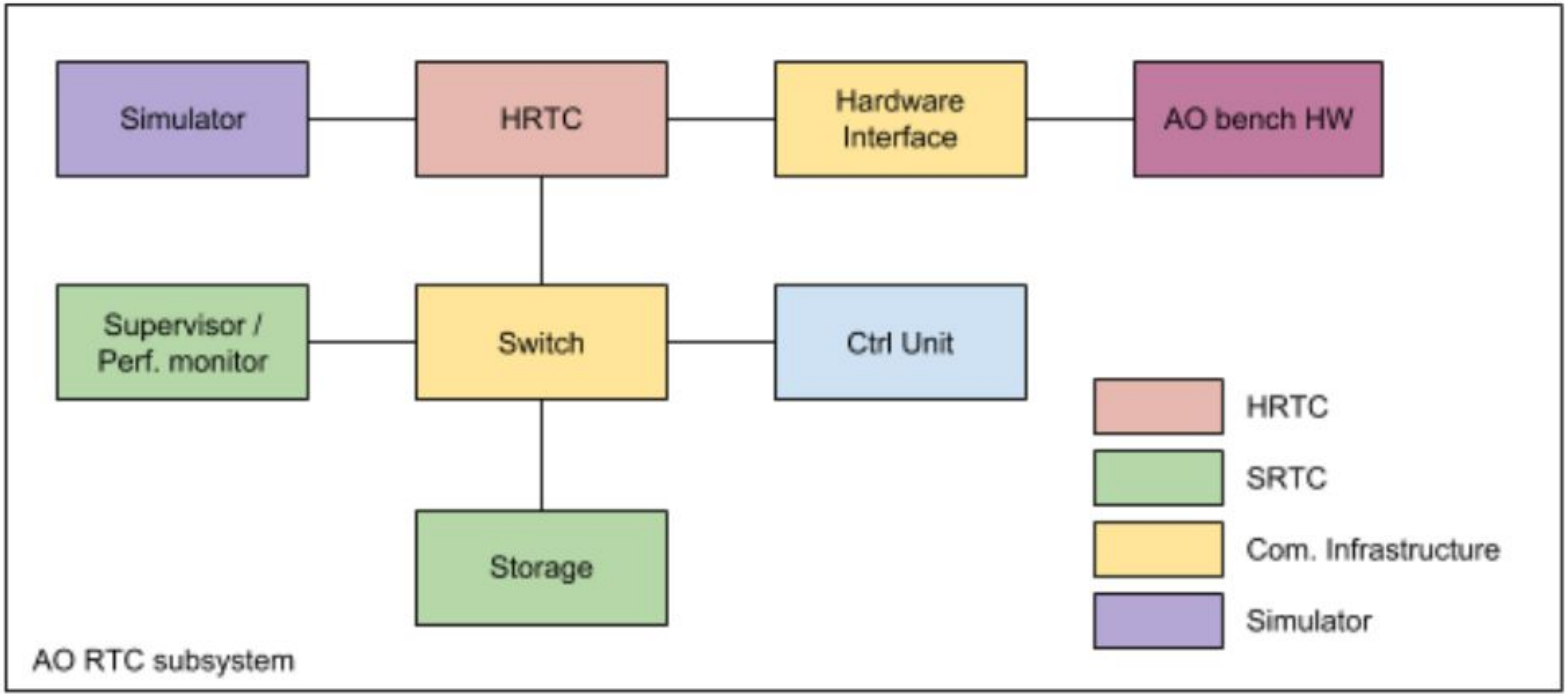


Proposed concept: SPARTA revamped

AO RTC Detailed system architecture

Inspired by SPARTA functional decomposition, inheriting from previous R&D

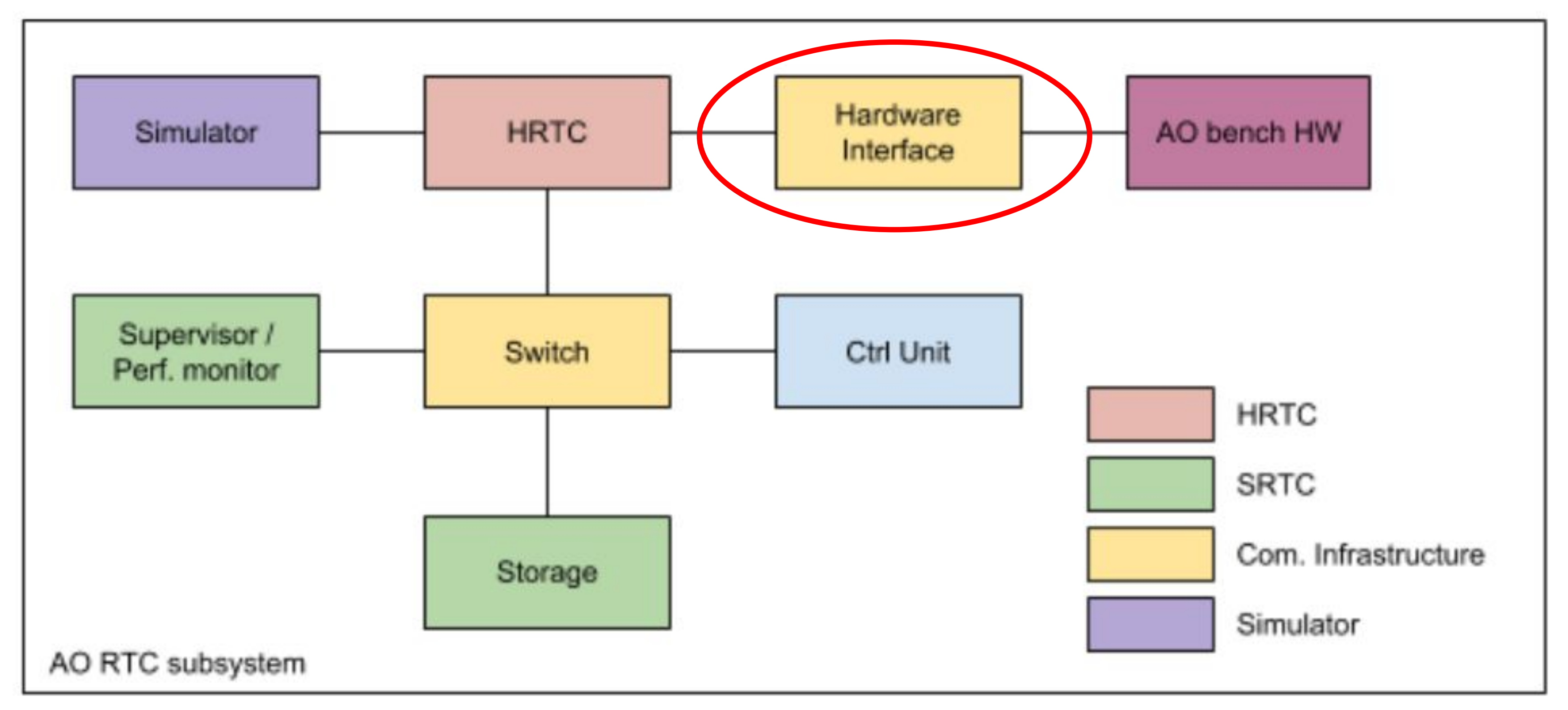
- Built to serve several AO instruments:
    - Keck Telescope: ongoing contract (first light yesterday !)
    - MICADO: ELT first light instrument (on-sky by 2025)
    - MAVIS: high complexity VLT instrument (on-sky by 2026)
    - ...



AO RTC DETAILED SYSTEM ARCHITECTURE

Inspired by SPARTA functional decomposition, inheriting from previous R&D

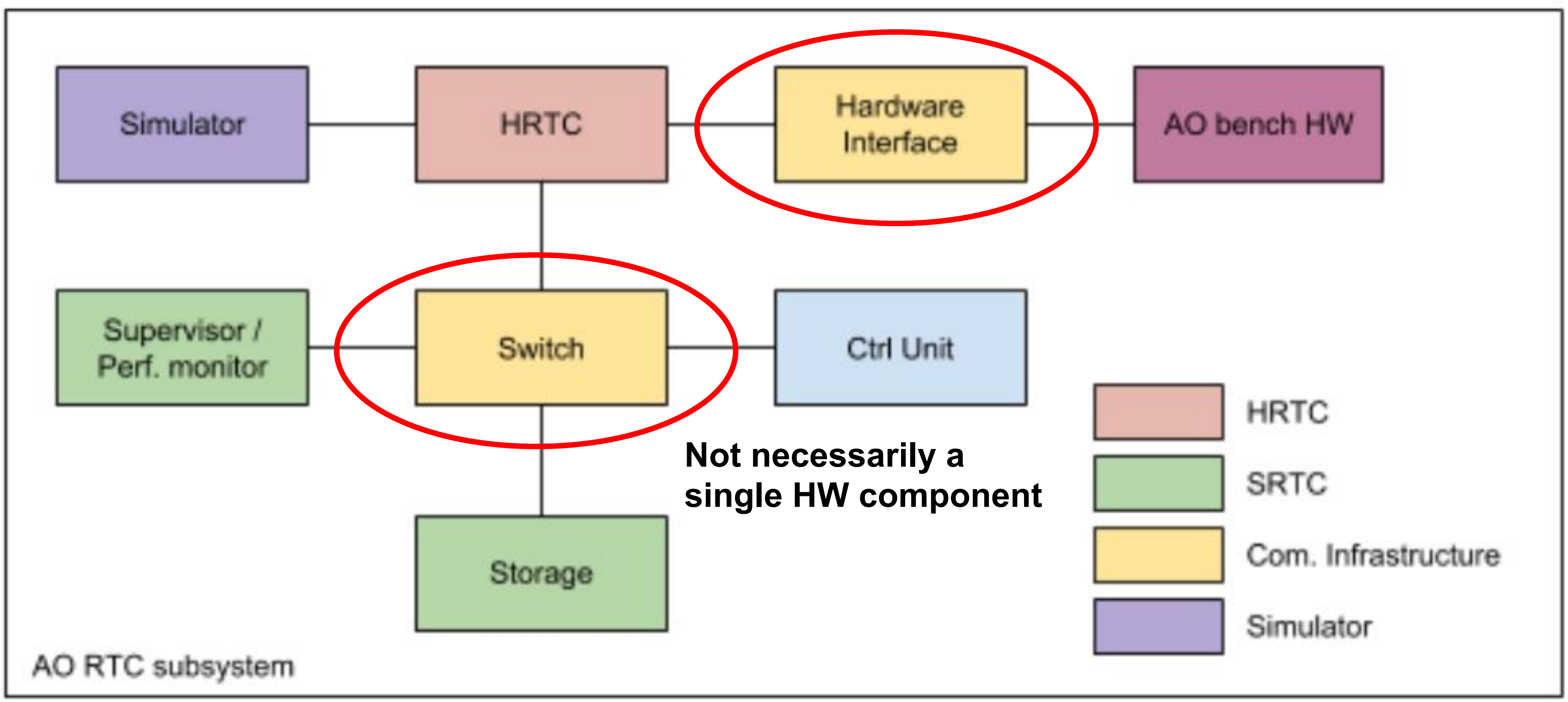
- **Flexibility : dedicated HW interface**



AO RTC DETAILED SYSTEM ARCHITECTURE

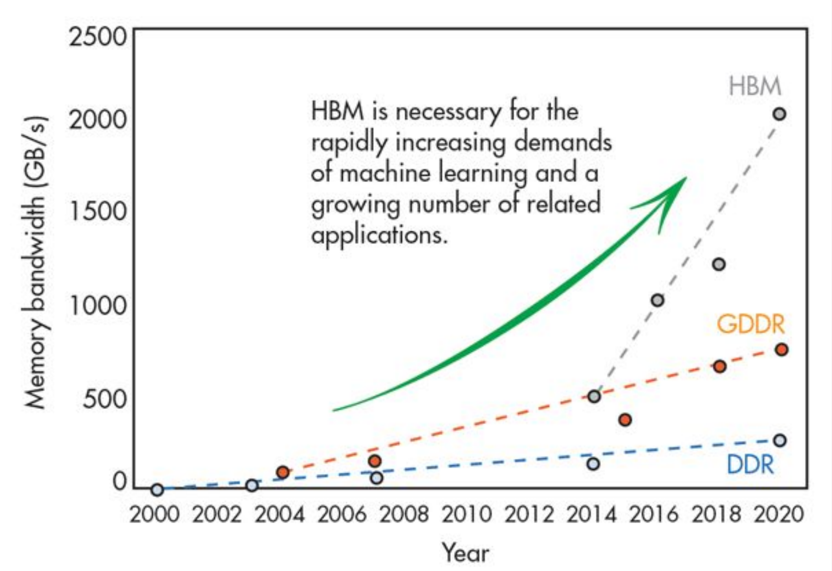
Inspired by SPARTA functional decomposition, inheriting from previous R&D

- **Flexibility : dedicated HW interface**
- **SW based on abstraction layers**

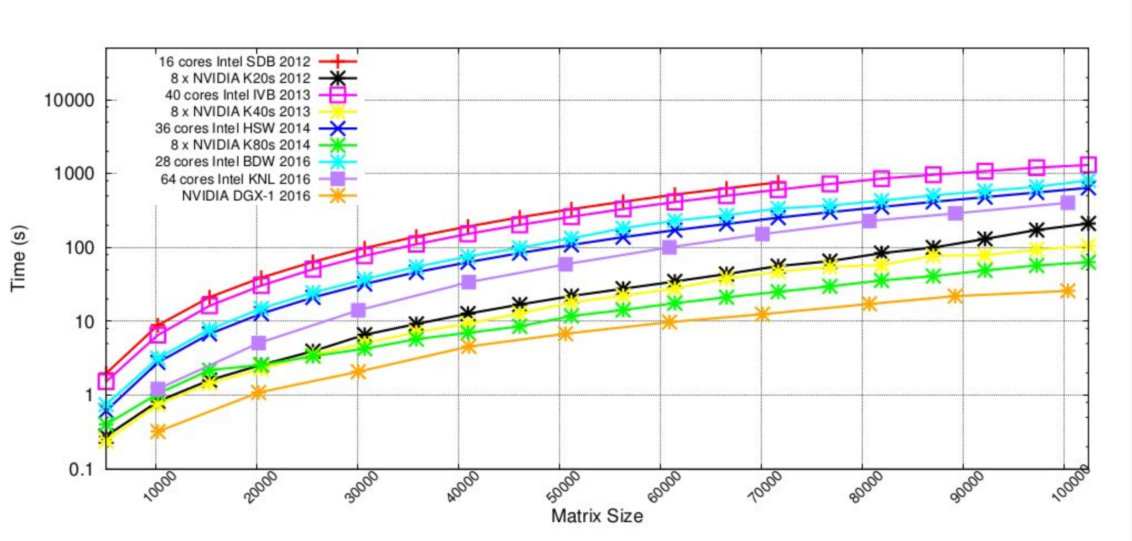


HARDWARE CHOICES

HRTC based on MVM: **memory bound**.
GPU technology, equipped with HBM memory is the silver bullet ! (as compared to CPU with classical DDR)



SRTC based on **embarrassingly parallel linear algebra**: GPUs perform 10x better than CPU for this kind of workload, consistently over the past 6 years.



Several hardware design options:

- All-in-one HRTC+SRTC
- Separate servers in a cluster configuration

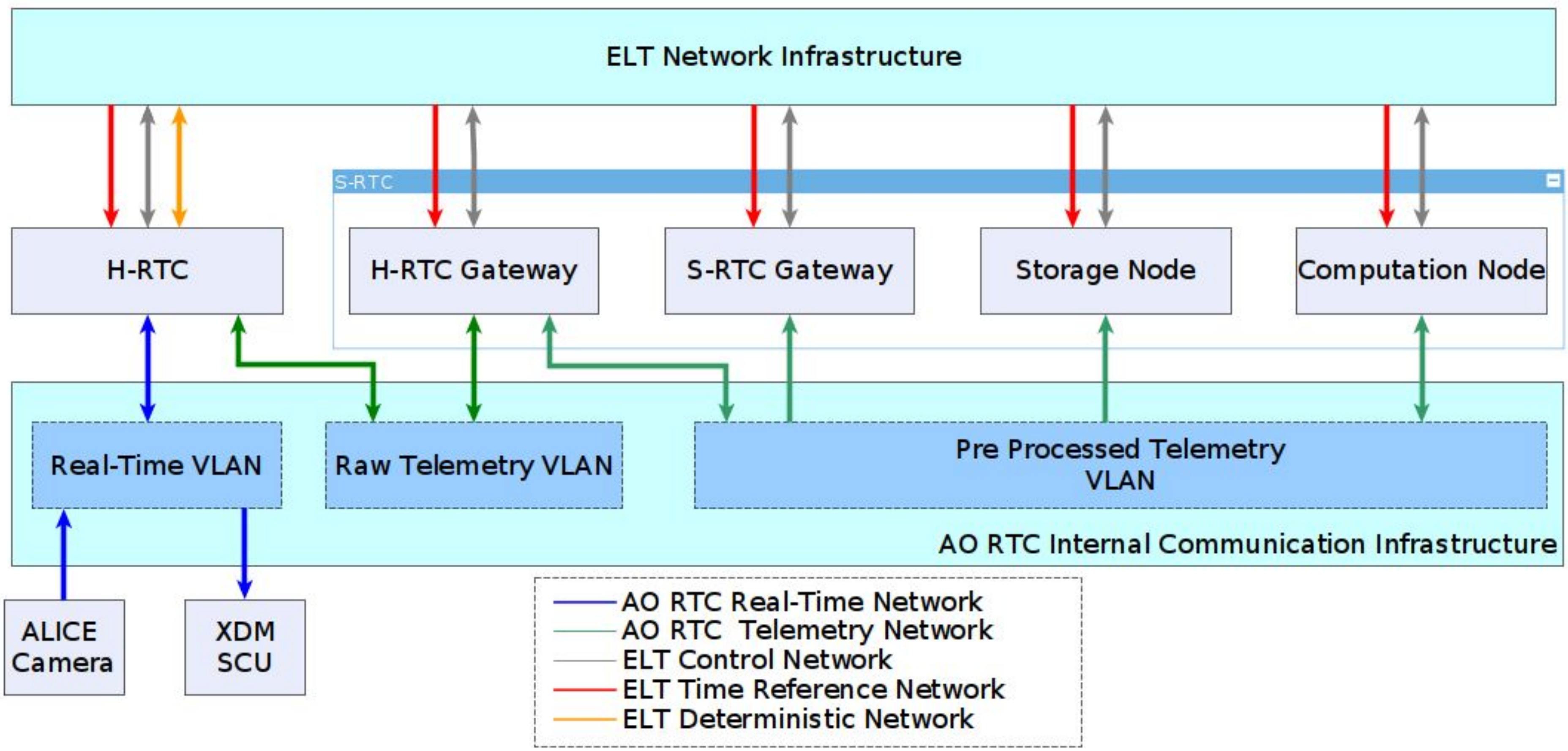
RTC Hardware design

The future is Now (and ESO compliant !)

Proposed Implementation

Based on the **ESO Standard RTC architecture for the ELT**

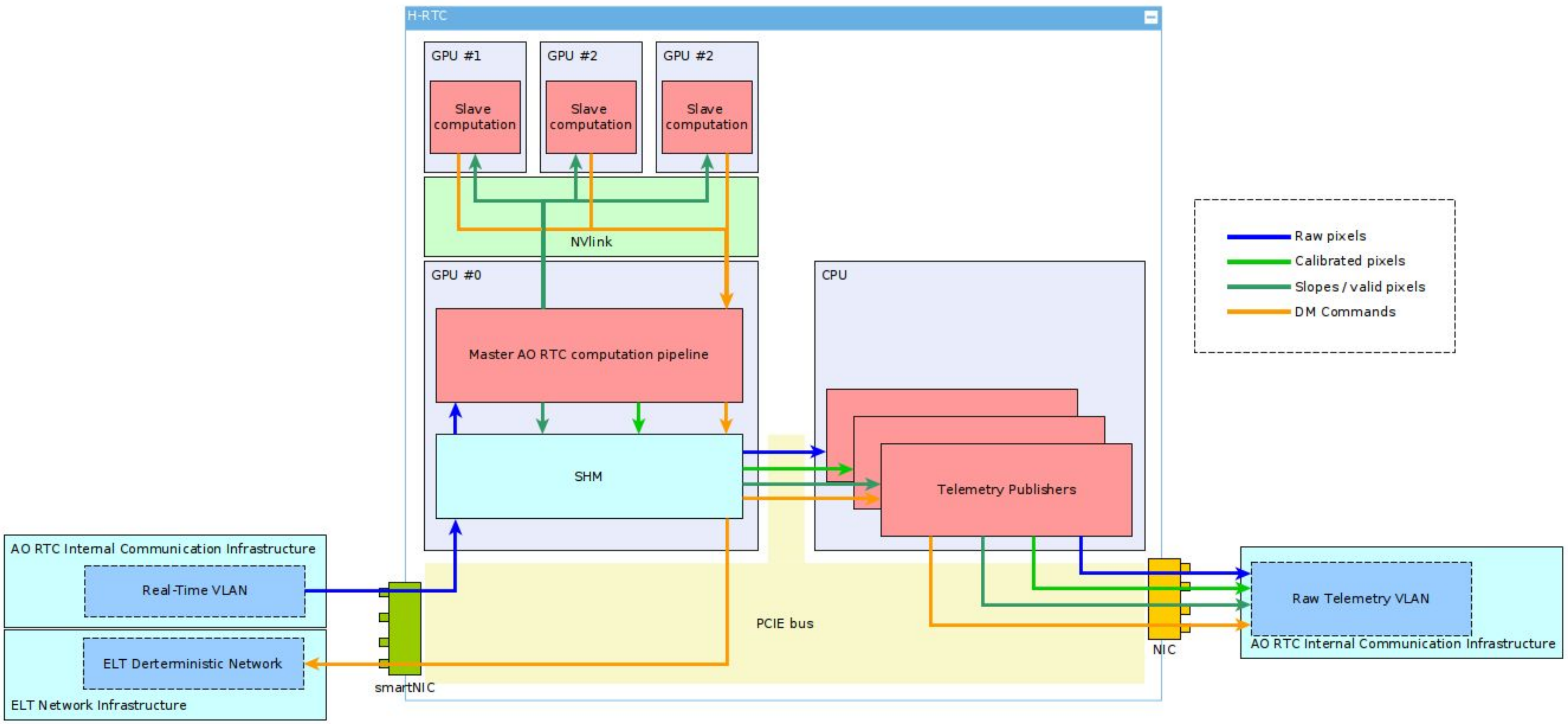
- Proposed for MICADO and MAVIS
- Can be easily adapted to SPHERE+



Proposed Implementation

Multi-GPU implementation of HRTC (dimensioning TBD for SPHERE+)

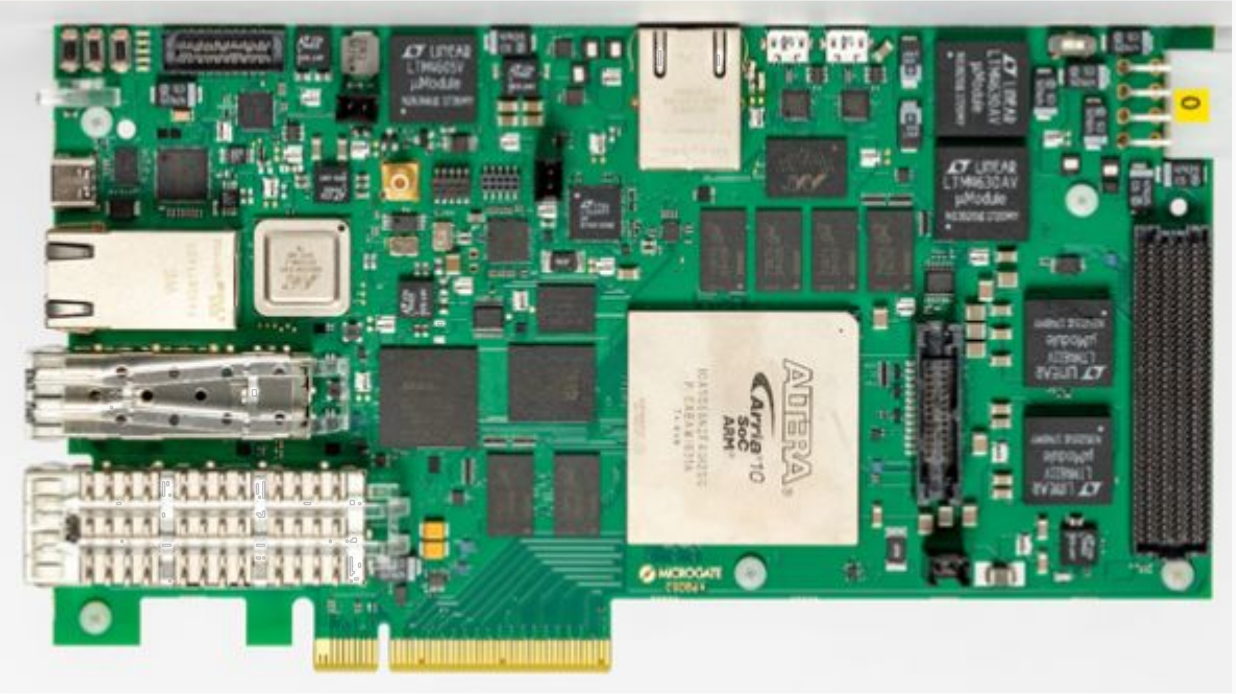
- MICADO & MAVIS design: leverage high density GPU cluster (PCIE + NVlink)
- Keck design: standard server with dual GPU (PCIe interface)



HRTC DATA STREAM INTERFACE

Based on µXlink board from Microgate

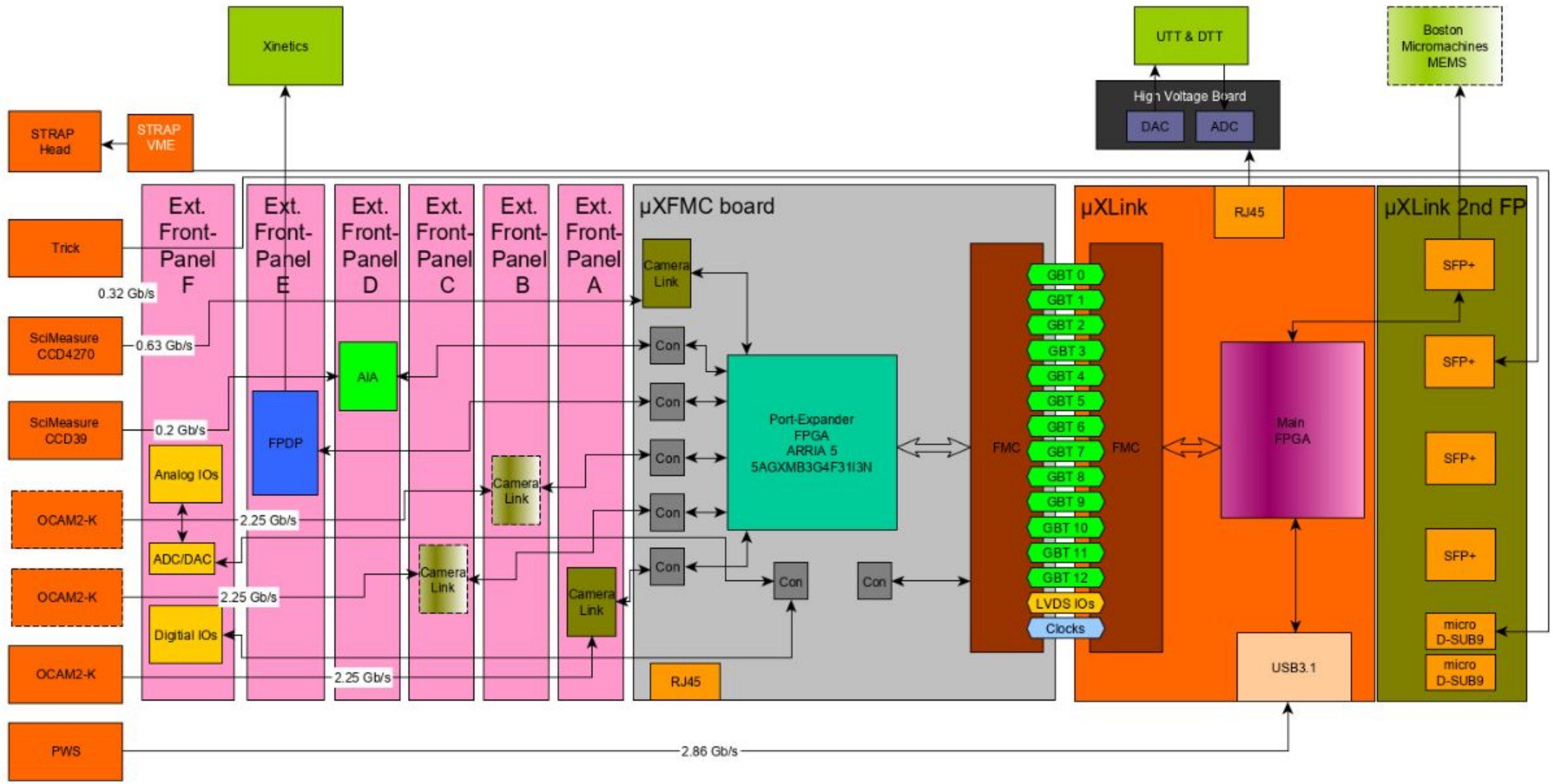
- Developed in the context of Green Flash, available as a COTS product
- Provide flexibility to any kind of hardware (WFS cameras, DMs, VLT / ELT frameworks) and efficient data transfer to/from compute units (GPUdirect)
- In production, used in ELT M4 design => long term availability to support ESO projects, already part of ESO ecosystem



HRTC DATA STREAM INTERFACE

Currently under final integration phase for Keck

- Multiple interfaces: CameraLink, 10G Ethernet, USB, sFPDP



HRTC DATA STREAM INTERFACE

Currently under final integration phase for Keck

- Multiple interfaces: CameraLink, 10G Ethernet, USB, sFPDP

HRTC DATA STREAM INTERFACE PERFORMANCE

Testing the data acquisition interface to the GPU:

- direct memory access (relying on standard development tools and factory versions of drivers / API)
- Here compared to “classical” CPU memory access

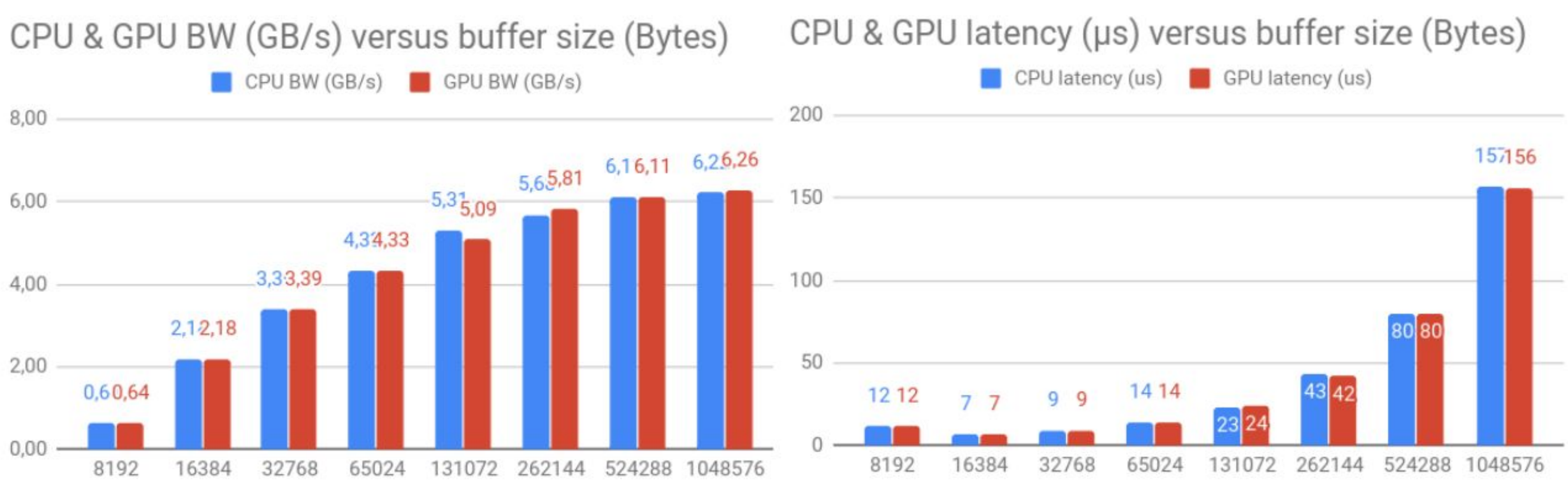


Figure 4-56 - Bandwidth and latency measurement of DMA operation between the µXcomp board and CPU (respectively GPU) with respect to buffer size

RTC Software design: Modular, comprehensive and powerful

AO RTC Software components

4 components:

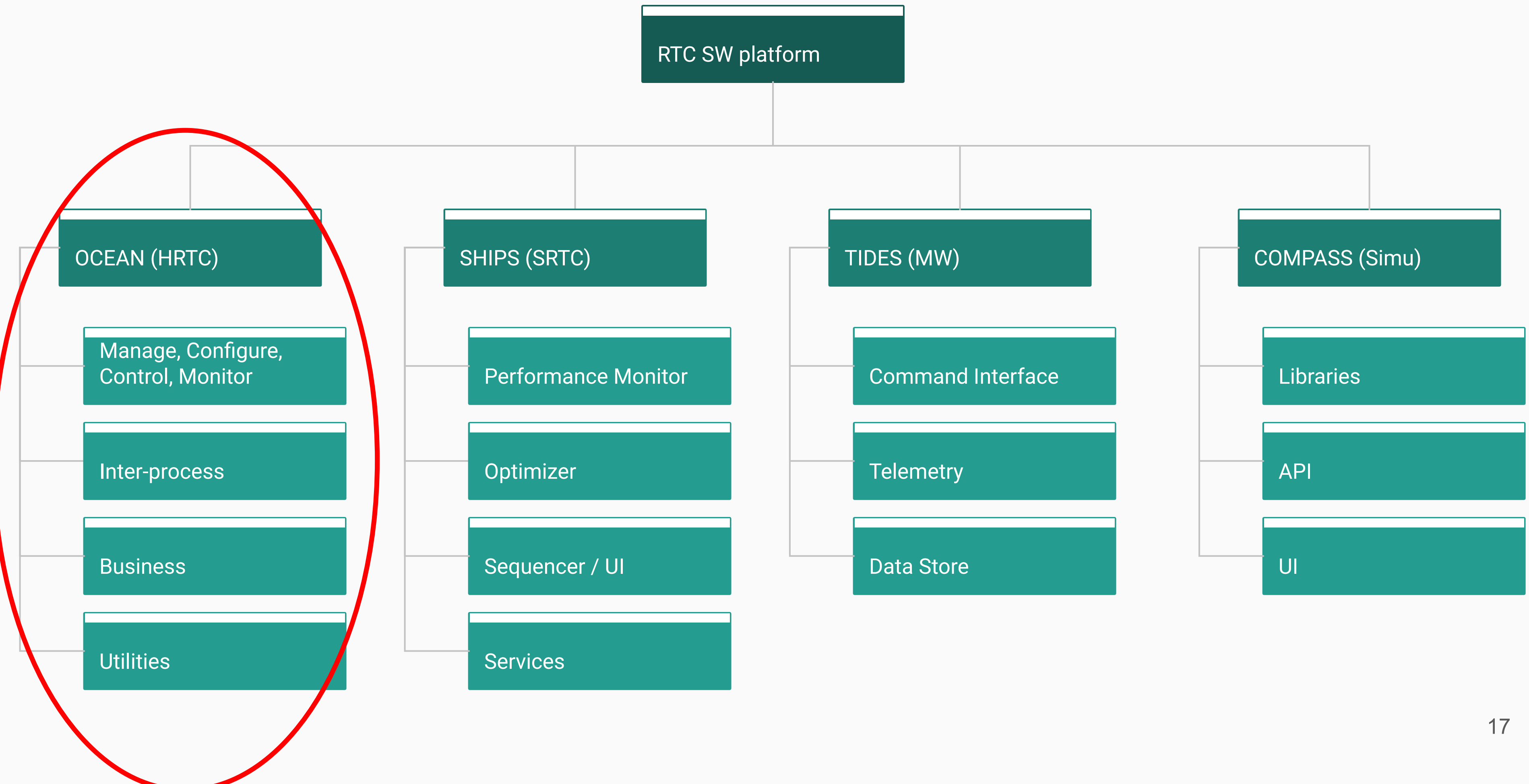
- OCEAN, SHIPS, TIDES and COMPASS

- RTC SW platform
  - OCEAN (HRTC)
    - Manage, Configure, Control, Monitor
    - Inter-process
    - Business
    - Utilities
  - SHIPS (SRTC)
    - Performance Monitor
    - Optimizer
    - Sequencer / UI
    - Services
  - TIDES (MW)
    - Command Interface
    - Telemetry
    - Data Store
  - COMPASS (Simu)
    - Libraries
    - API
    - UI

AO RTC Software components

4 components:

- OCEAN, SHIPS, TIDES and COMPASS



HRTC SW DESIGN

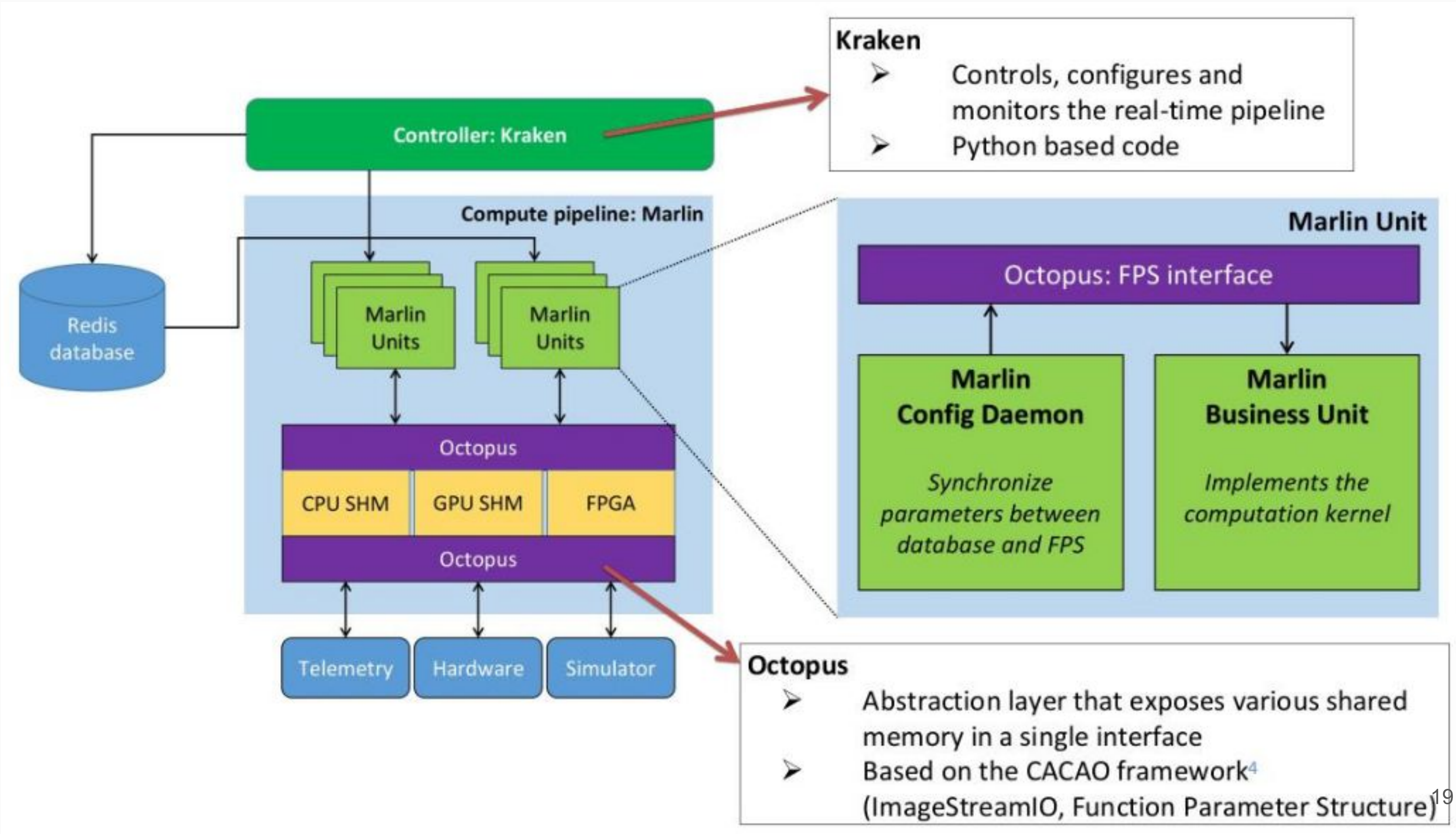
Flexible implementation in heterogeneous environment through **abstraction layers**

OCEAN (**O**ptimized **C**ore **E**ngine for **A**daptive optics pipeli**N**es)

- **Kraken**: Python based control, management and monitoring:
- **Marlin**: C++ library for real-time kernels execution
- **Octopus**: C++ library for data interface:
- Other components include: **Seahorse, Wyrm, Fish** (utilities)

HRTC SW DESIGN

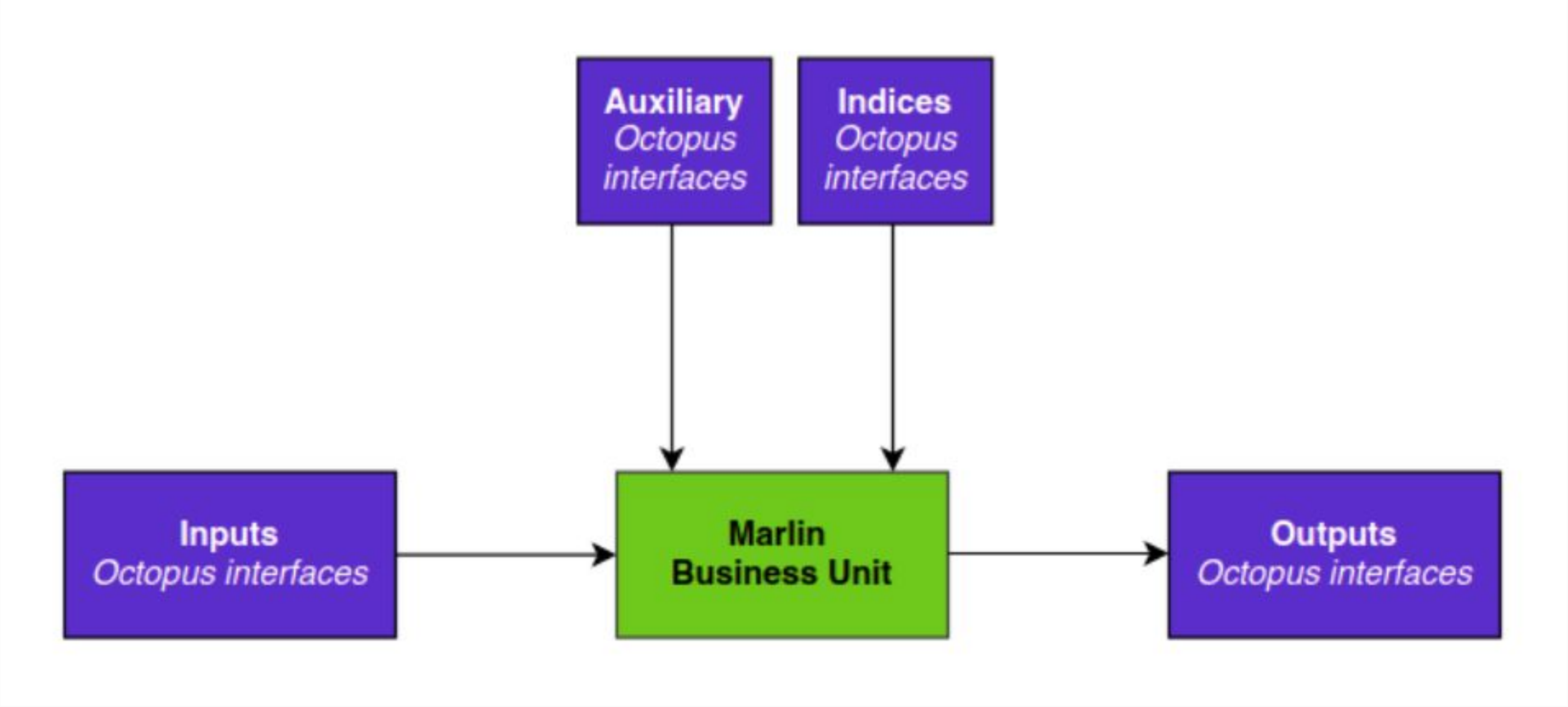
Core pipeline, made highly modular through the Marlin library



HRTC SW DESIGN

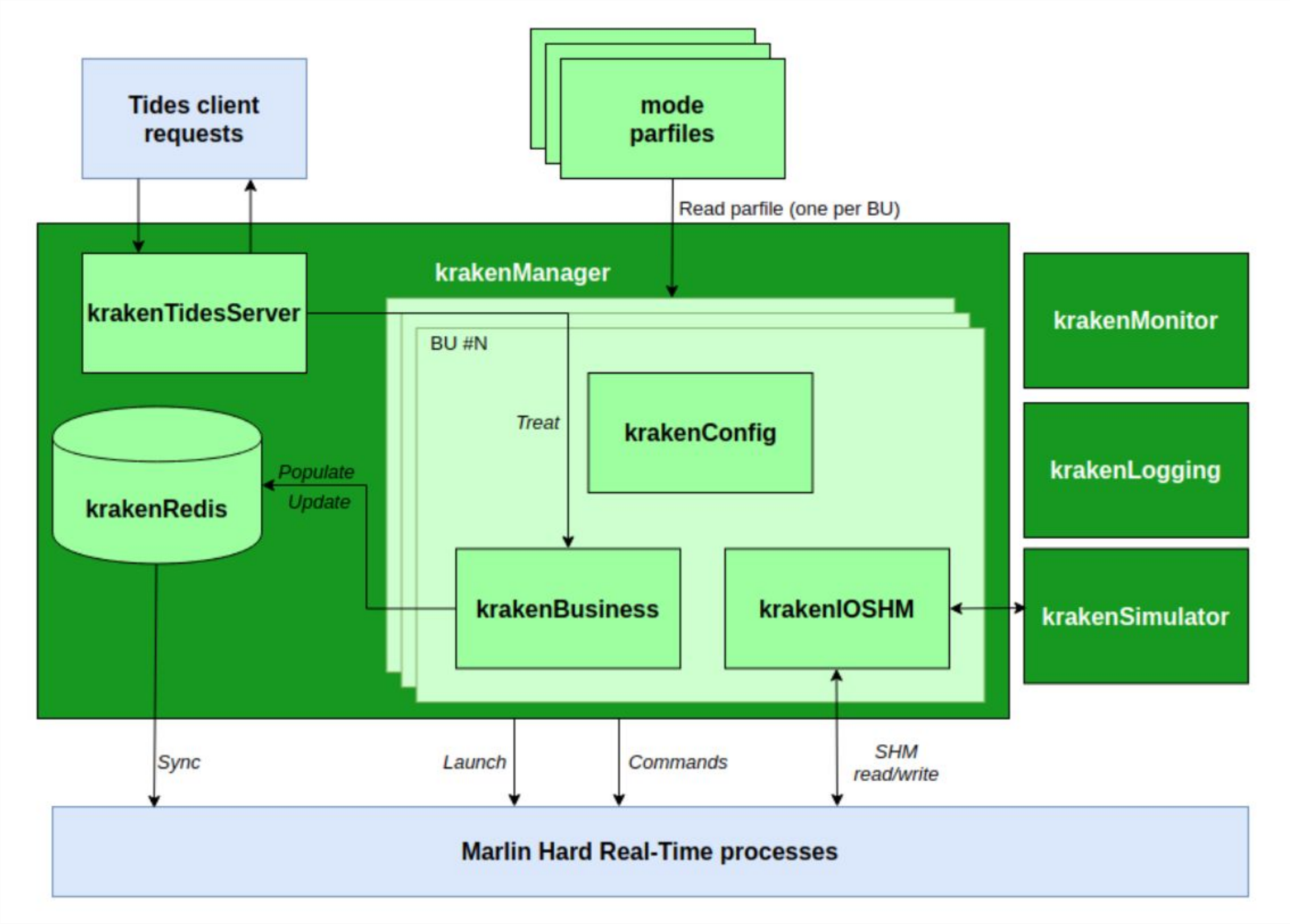
Main features of Marlin library:

- Moved away from persistent kernel solutions:
  - Better flexibility
  - Want to rely as much as possible on standard libraries
- Retained one concept: “active wait” for kernels sync
- Added support for semaphore based sync



HRTC SW DESIGN

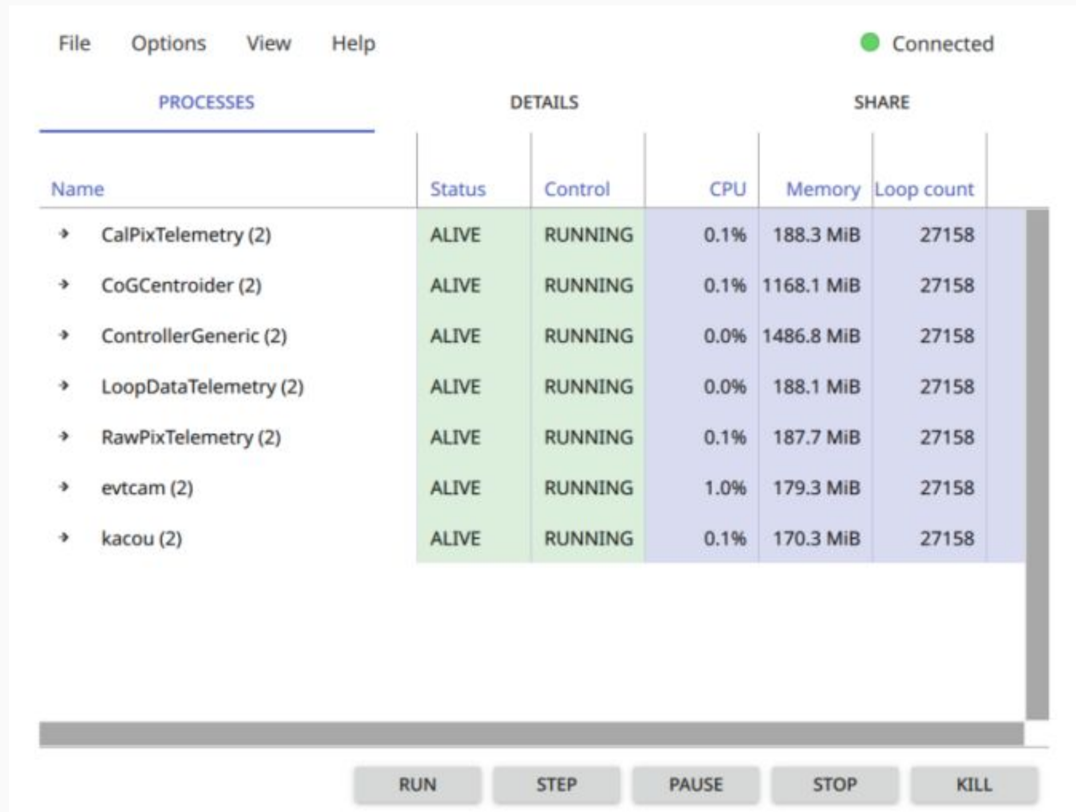
Kraken Software architecture



HRTC SW CONTROL / MONITORING INTERFACE

Python + Qt graphical interface

- Based on backend (running on RTC) publishing on a dedicated socket + frontend (running anywhere)
- Start / Stop / Pause / Step / Kill Processes
- Access to logs

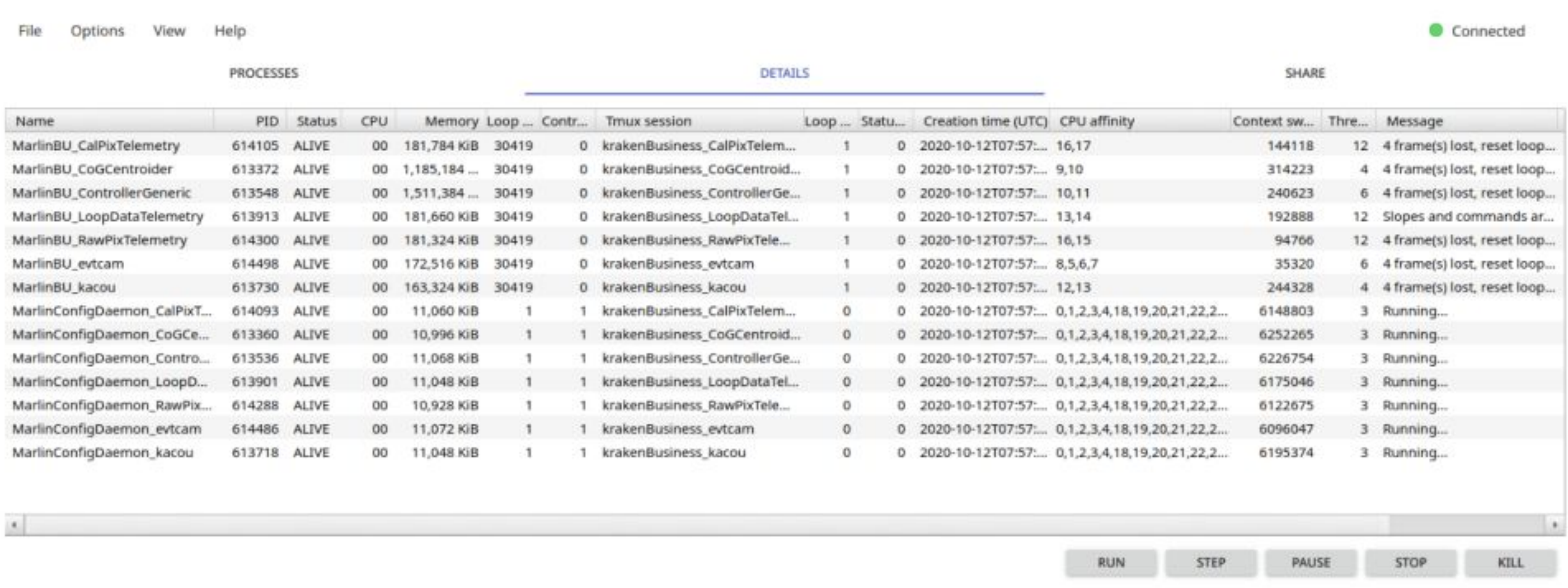


HRTC SW CONTROL / MONITORING INTERFACE

Python + Qt graphical interface

- Based on backend (running on RTC) publishing on a dedicated socket + frontend (running anywhere)

- Detailed view



Name	PID	Status	CPU	Memory	Loop ...	Contr...	Tmux session	Loop ...	Statu...	Creation time (UTC)	CPU affinity	Context sw...	Thre...	Message
MarlinBU_CalPixTelemetry	614105	ALIVE	00	181,784 KiB	30419	0	krakenBusiness_CalPixTelem...	1	0	2020-10-12T07:57:...	16,17	144118	12	4 frame(s) lost, reset loop...
MarlinBU_CoGCentroider	613372	ALIVE	00	1,185,184 ...	30419	0	krakenBusiness_CoGCentroid...	1	0	2020-10-12T07:57:...	9,10	314223	4	4 frame(s) lost, reset loop...
MarlinBU_ControllerGeneric	613548	ALIVE	00	1,511,384 ...	30419	0	krakenBusiness_ControllerGe...	1	0	2020-10-12T07:57:...	10,11	240623	6	4 frame(s) lost, reset loop...
MarlinBU_LoopDataTelemetry	613913	ALIVE	00	181,660 KiB	30419	0	krakenBusiness_LoopDataTel...	1	0	2020-10-12T07:57:...	13,14	192888	12	Slopes and commands ar...
MarlinBU_RawPixTelemetry	614300	ALIVE	00	181,324 KiB	30419	0	krakenBusiness_RawPixTele...	1	0	2020-10-12T07:57:...	16,15	94766	12	4 frame(s) lost, reset loop...
MarlinBU_evtcam	614498	ALIVE	00	172,516 KiB	30419	0	krakenBusiness_evtcam	1	0	2020-10-12T07:57:...	8,5,6,7	35320	6	4 frame(s) lost, reset loop...
MarlinBU_kacou	613730	ALIVE	00	163,324 KiB	30419	0	krakenBusiness_kacou	1	0	2020-10-12T07:57:...	12,13	244328	4	4 frame(s) lost, reset loop...
MarlinConfigDaemon_CalPixT...	614093	ALIVE	00	11,060 KiB	1	1	krakenBusiness_CalPixTelem...	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6148803	3	Running...
MarlinConfigDaemon_CoGCe...	613360	ALIVE	00	10,996 KiB	1	1	krakenBusiness_CoGCentroid...	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6252265	3	Running...
MarlinConfigDaemon_Contro...	613536	ALIVE	00	11,068 KiB	1	1	krakenBusiness_ControllerGe...	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6226754	3	Running...
MarlinConfigDaemon_LoopD...	613901	ALIVE	00	11,048 KiB	1	1	krakenBusiness_LoopDataTel...	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6175046	3	Running...
MarlinConfigDaemon_RawPix...	614288	ALIVE	00	10,928 KiB	1	1	krakenBusiness_RawPixTele...	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6122675	3	Running...
MarlinConfigDaemon_evtcam	614486	ALIVE	00	11,072 KiB	1	1	krakenBusiness_evtcam	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6096047	3	Running...
MarlinConfigDaemon_kacou	613718	ALIVE	00	11,048 KiB	1	1	krakenBusiness_kacou	0	0	2020-10-12T07:57:...	0,1,2,3,4,18,19,20,21,22,2...	6195374	3	Running...

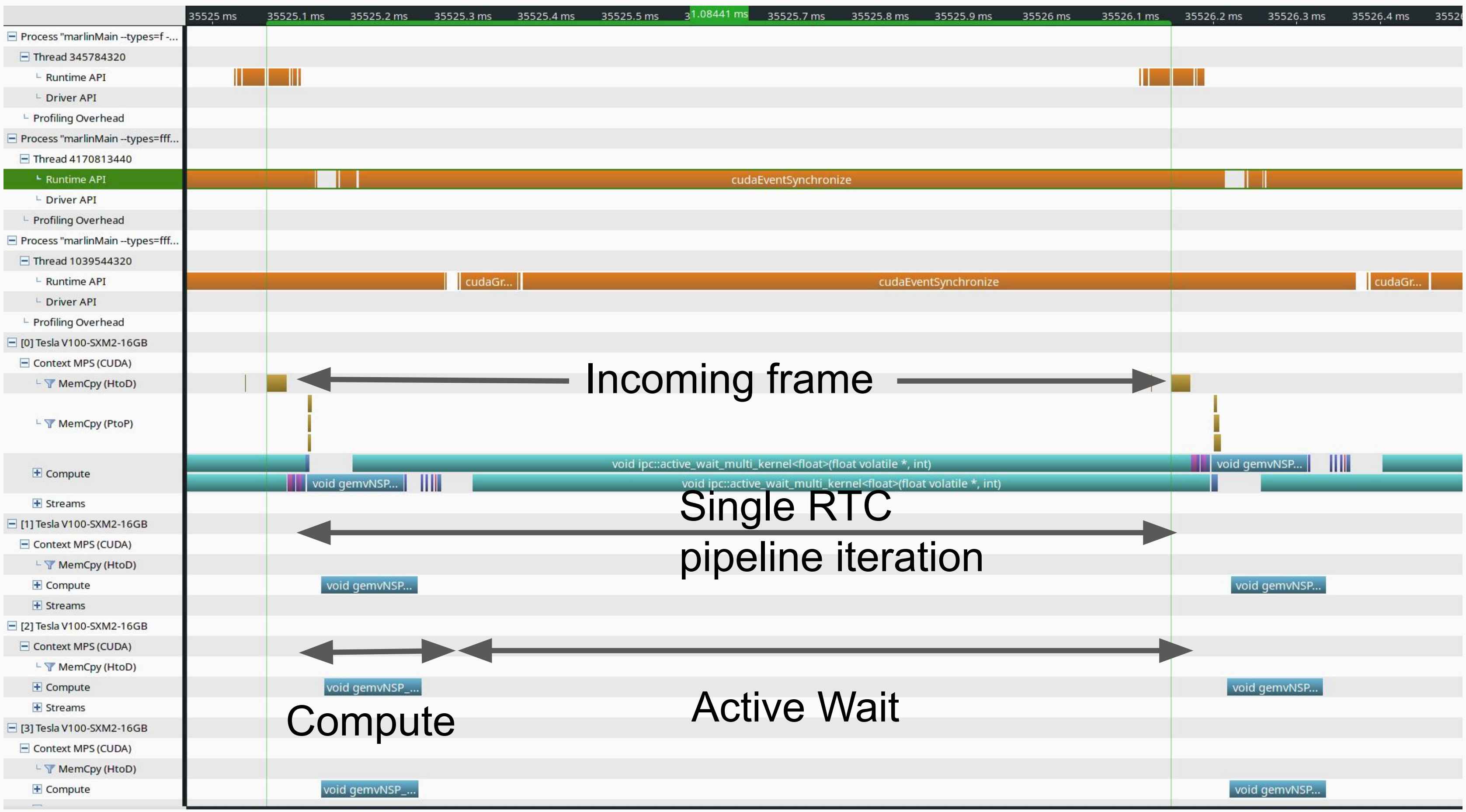
HRTC Performance

COSMIC is already there.

HRTC SW PERFORMANCE

Typical profile: pipeline includes frame transfer + centroiding + MVM

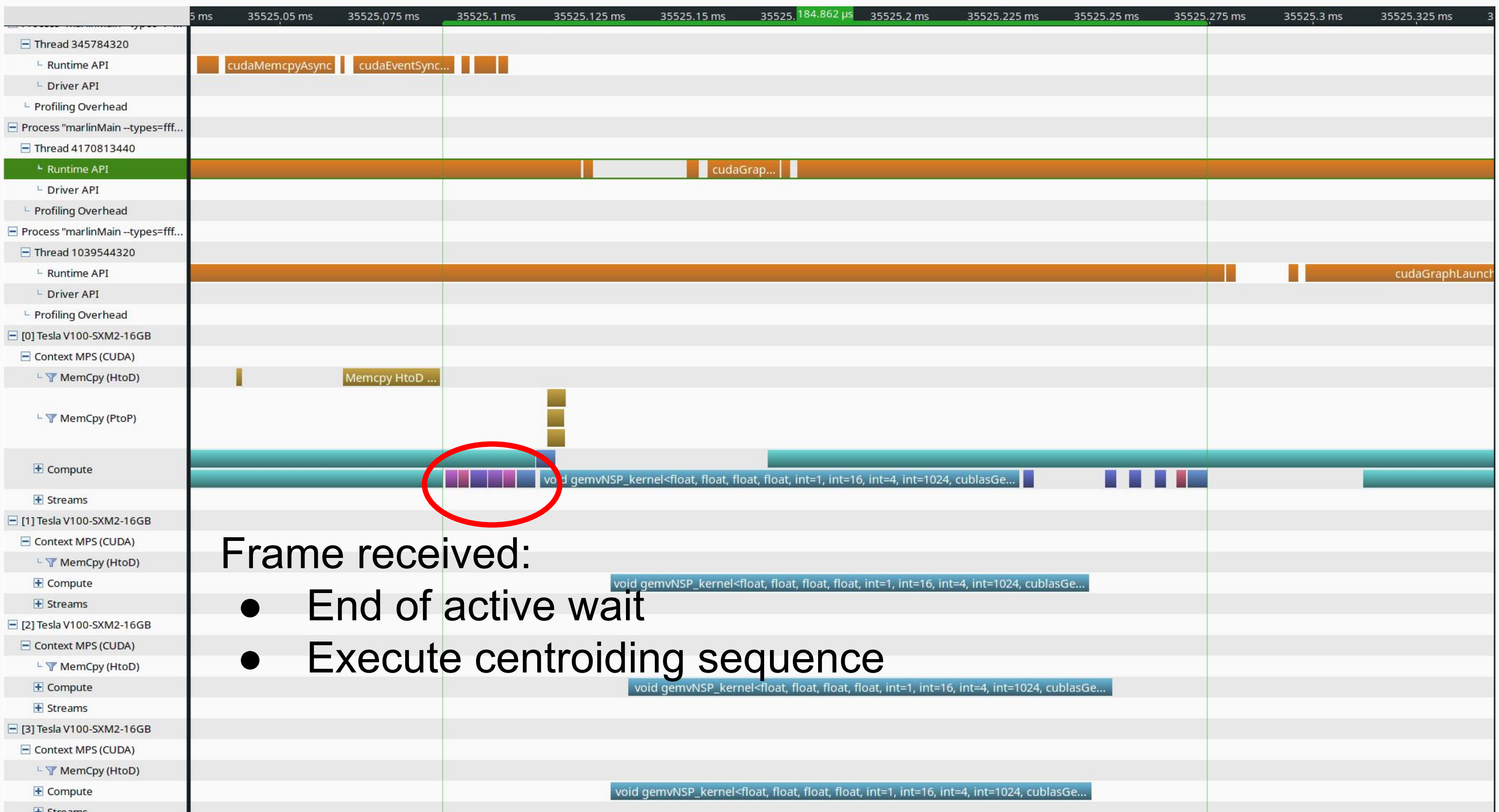
- WFS frames are sent by a hardware emulator at a regular rate (1 kHz)
- GPU is mostly “active waiting”



HRTC SW PERFORMANCE

Typical profile: pipeline includes frame transfer + centroiding + MVM

- WFS frames are sent by a hardware emulator at a regular rate (1 kHz)
- GPU is mostly "active waiting"



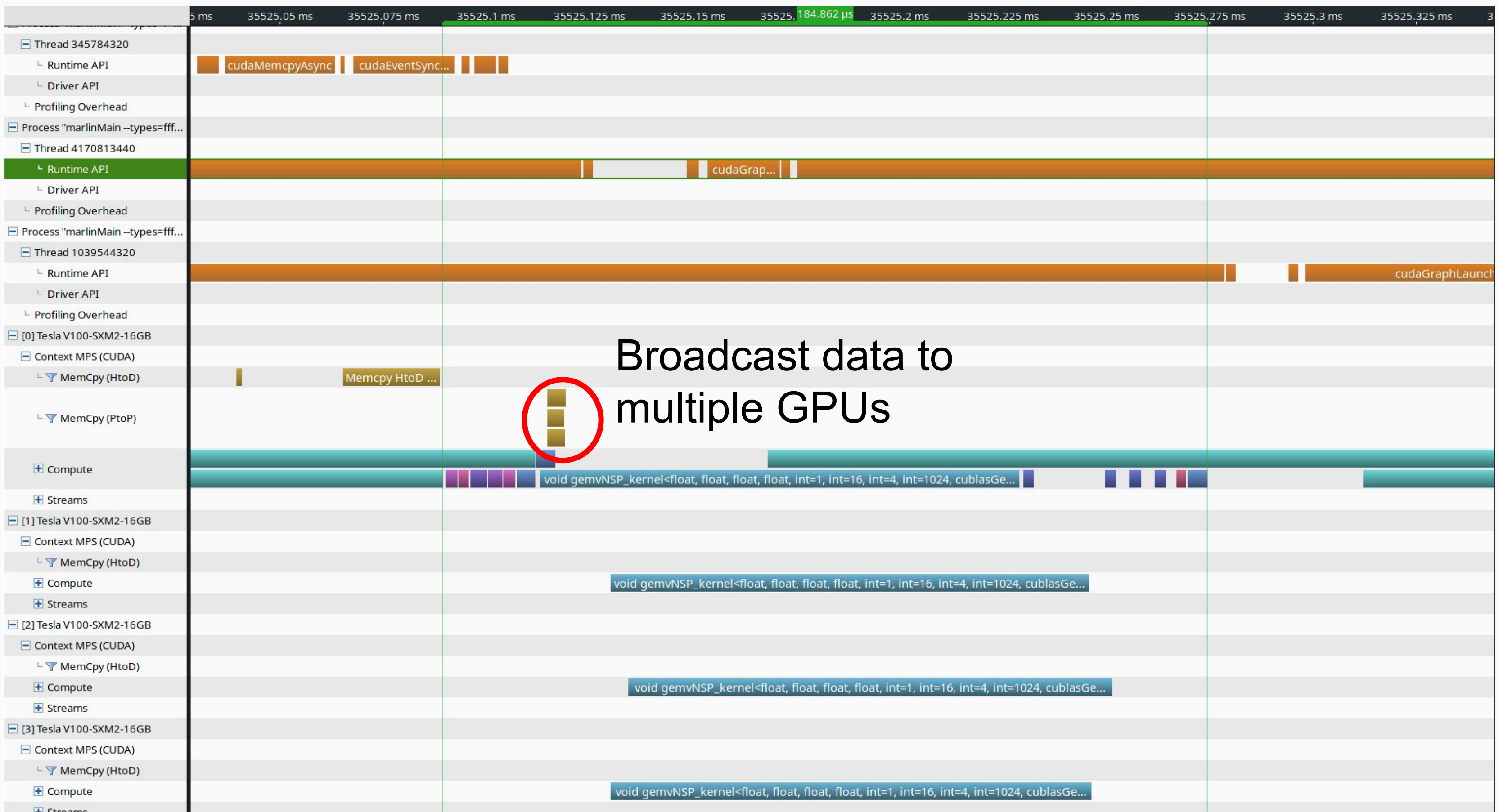
Frame received:

- End of active wait
- Execute centroiding sequence

HRTC SW PERFORMANCE

Typical profile: pipeline includes frame transfer + centroiding + MVM

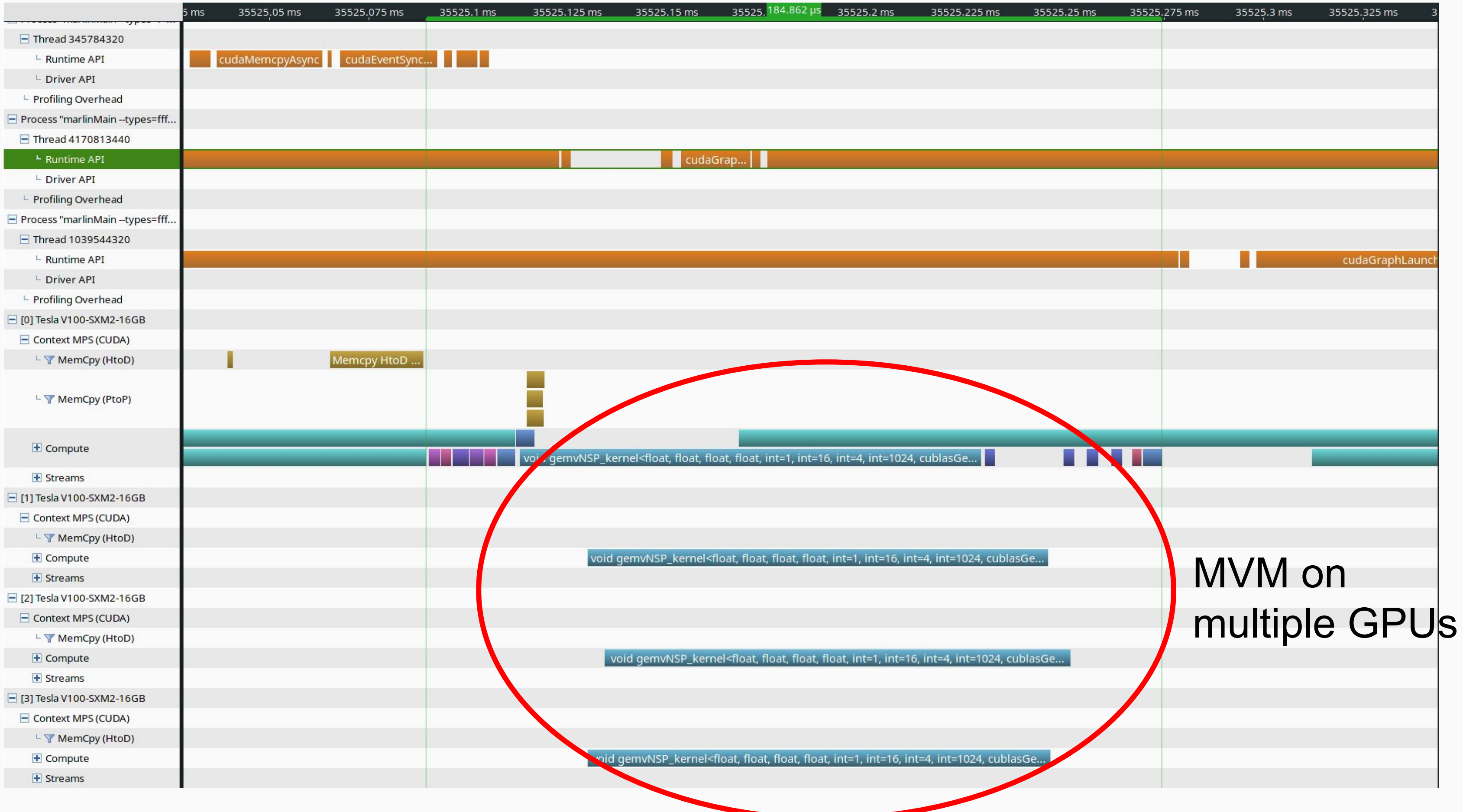
- WFS frames are sent by a hardware emulator at a regular rate (1 kHz)
- GPU is mostly “active waiting”



HRTC SW PERFORMANCE

Typical profile: pipeline includes frame transfer + centroiding + MVM

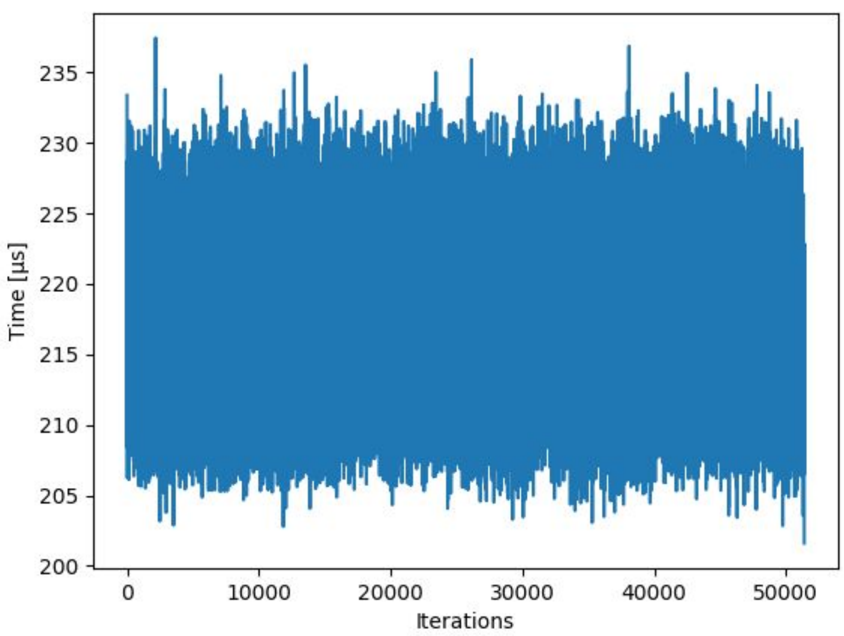
- WFS frames are sent by a hardware emulator at a regular rate (1 kHz)
- GPU is mostly “active waiting”

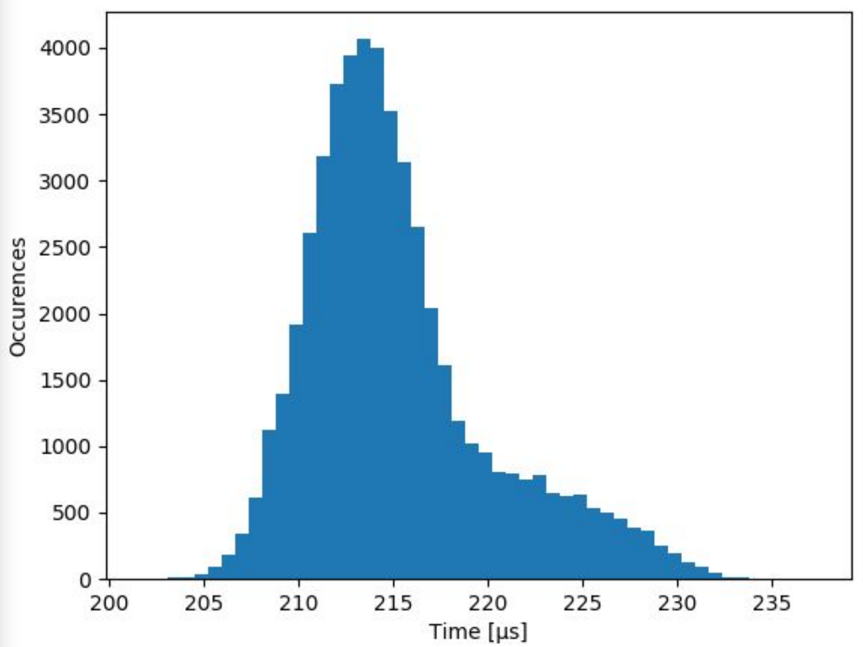


HRTC SW PERFORMANCE

Initial performance analysis : MAVIS pipeline (6+3 WFS, 5k x 20k command matrix) on 4 GPUs (NVIDIA V100)

- **Mean time-to-solution** (pure RTC latency) : **~215µs**
- P2V jitter: ~40µs
- **RMS jitter: ~5µs**

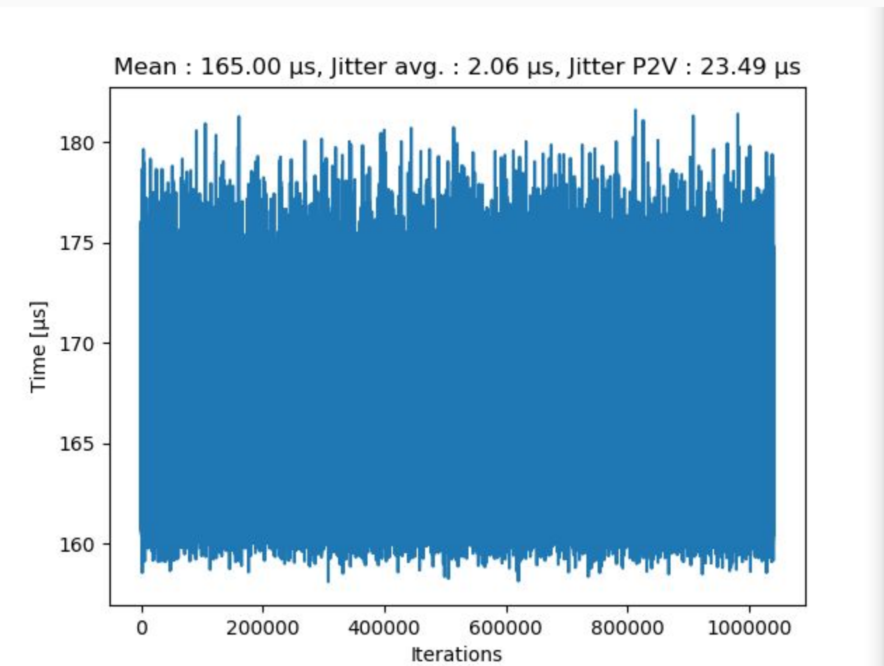


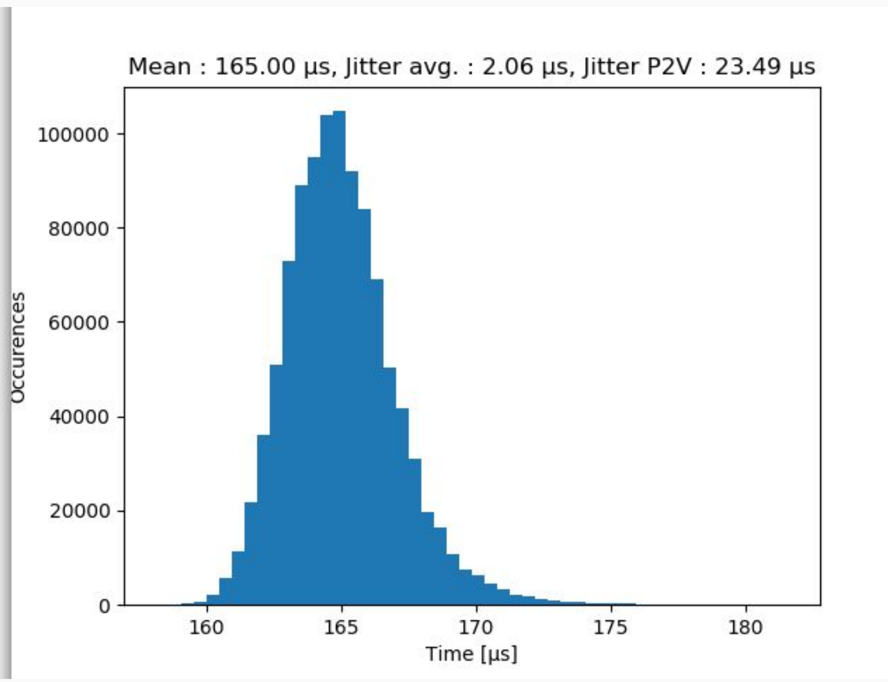


HRTC SW PERFORMANCE

Current performance figures : MICADO pipeline (1 Pyr WFS, 5k x 25k command matrix) on 4 GPUs (NVIDIA V100)

- **Mean time-to-solution** (pure RTC latency) : **~165µs**
- P2V jitter: ~23µs
- **RMS jitter: ~2µs**

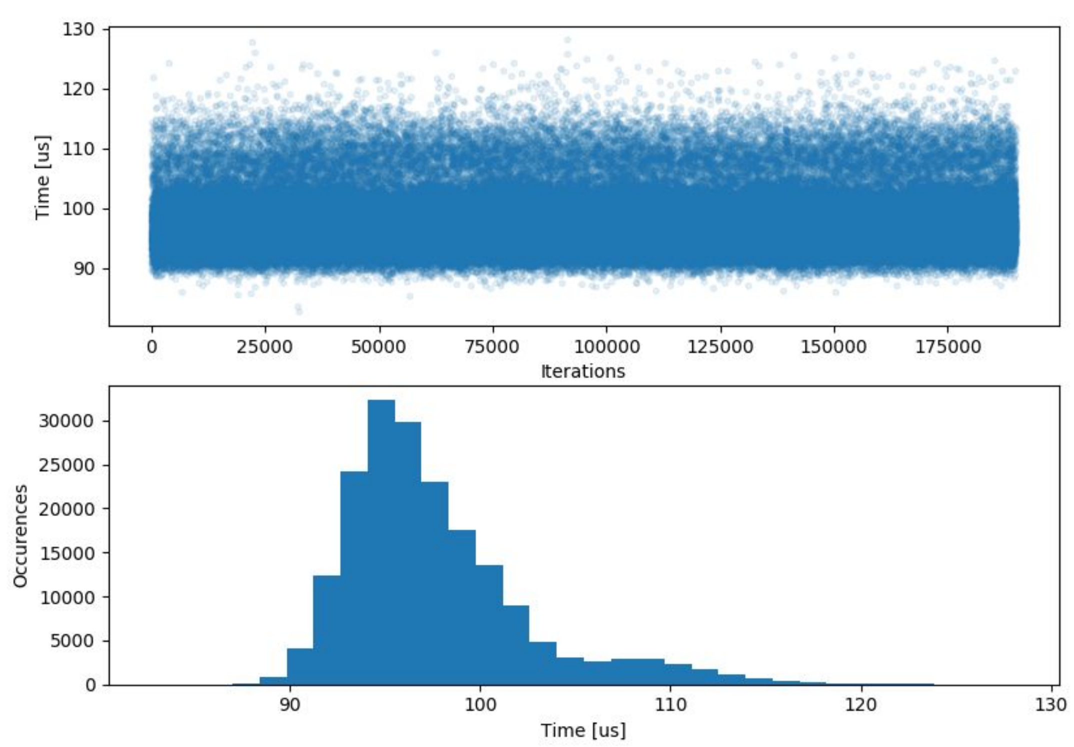




HRTC SW PERFORMANCE

End-to-end performance (i.e. with hardware interface) : Keck pipeline (1 SH WFS, ~350 x 600 command matrix, **11 BUs sequence**) on 1 GPU (NVIDIA V100)

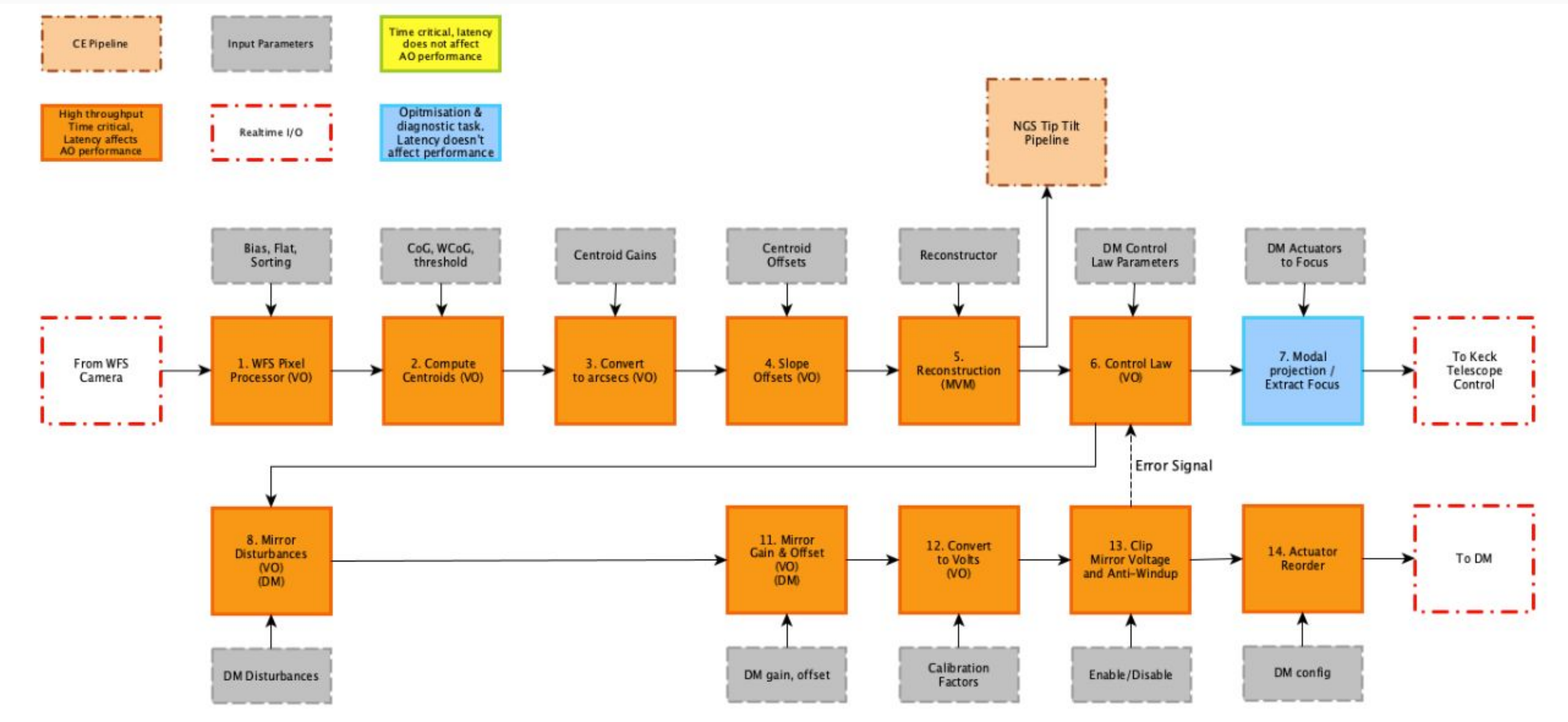
- **Mean time-to-solution** (pure RTC latency) : **~105µs**



HRTC SW PERFORMANCE

End-to-end performance (i.e. with hardware interface) : Keck pipeline (1 SH WFS, ~350 x 600 command matrix, **11 BUs sequence**) on 1 GPU (NVIDIA V100)

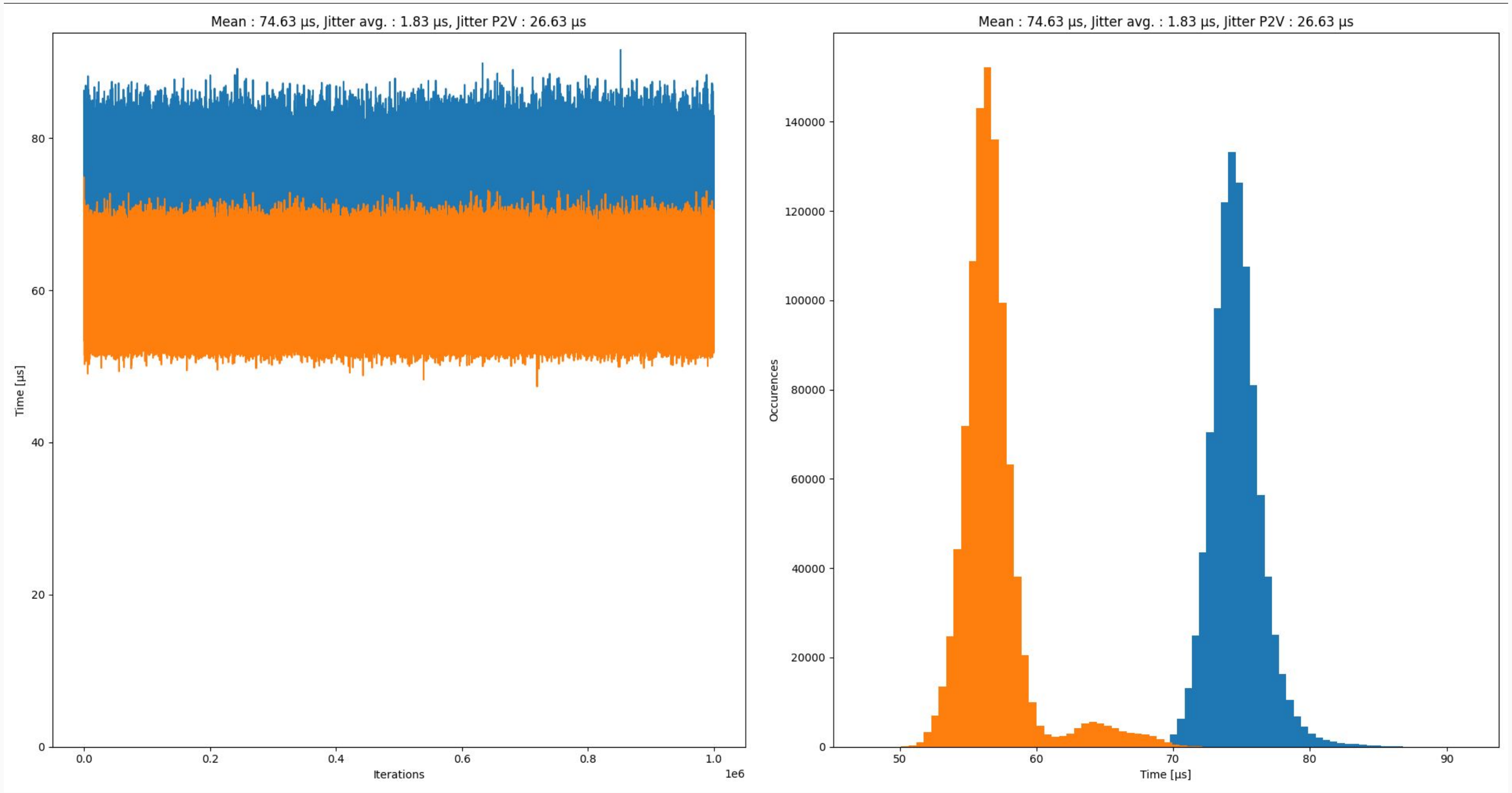
- **Mean time-to-solution** (pure RTC latency) : **~105µs**



HRTC SW PERFORMANCE

End-to-end performance (i.e. with hardware interface) : Keck pipeline (1 SH WFS, ~350 x 600 command matrix, **11 BUs sequence**) on 1 GPU (NVIDIA V100)

- **Already supporting two stages** at different framerates (orange: TT stage, blue HO stage)



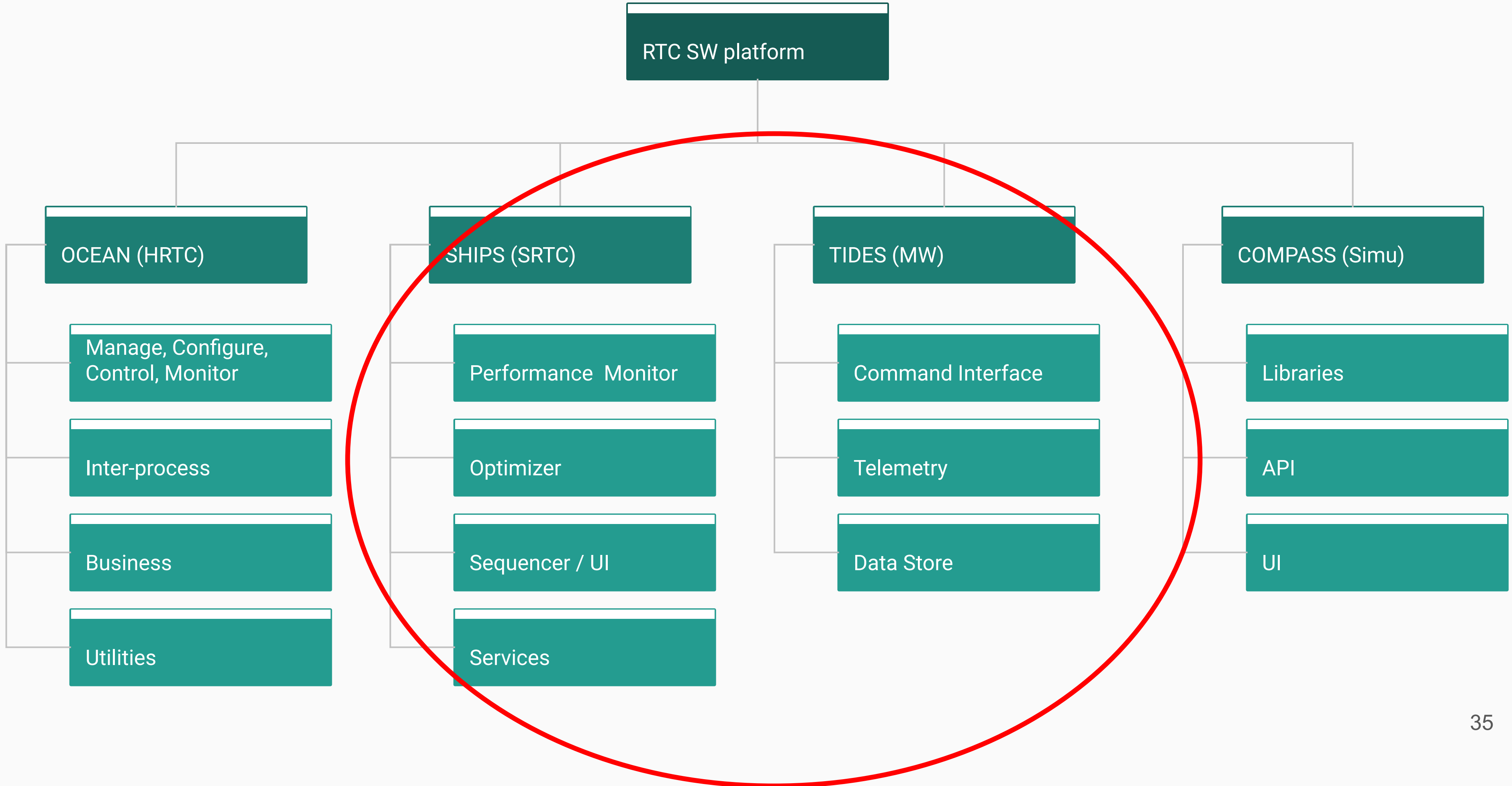
SRTC software

A feature-Rich stack fitting all AO flavors needs

AO RTC Software components

4 components:

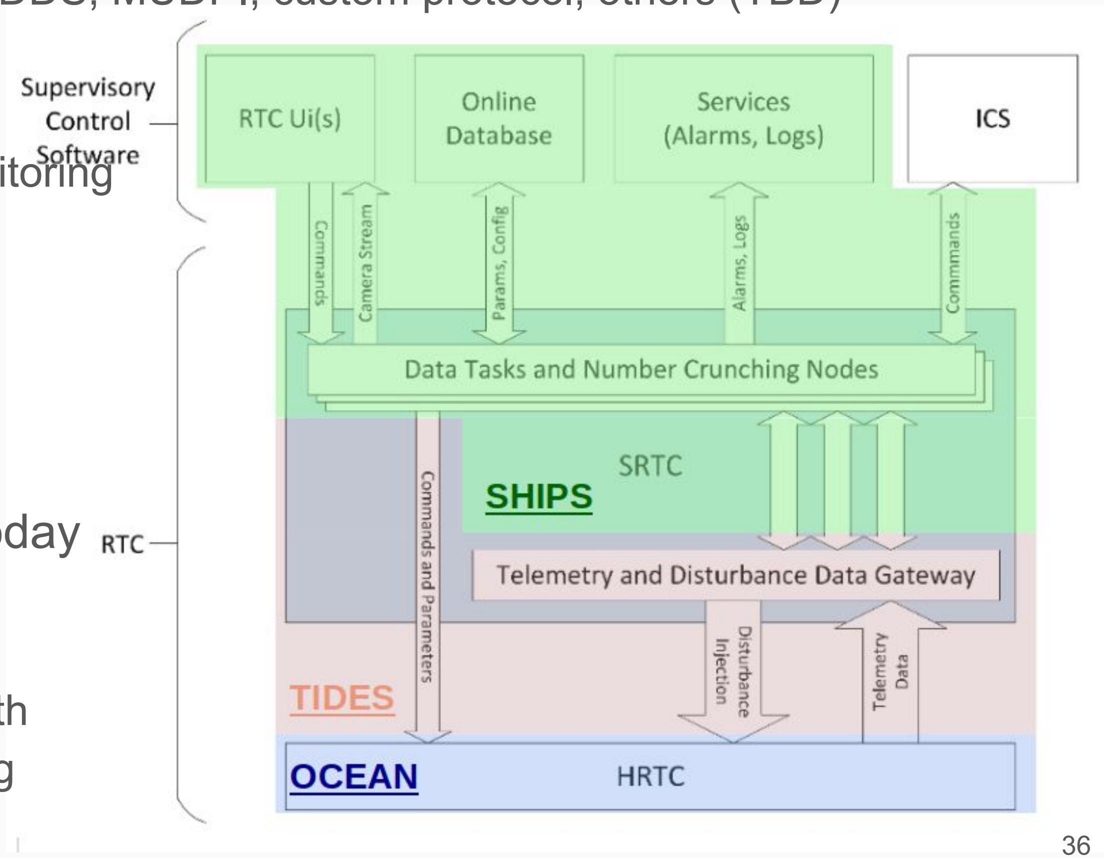
- OCEAN, SHIPS, TIDES and COMPASS



SRTC compatibility with RTC toolkit

Two components in which RTC Tk will fit:

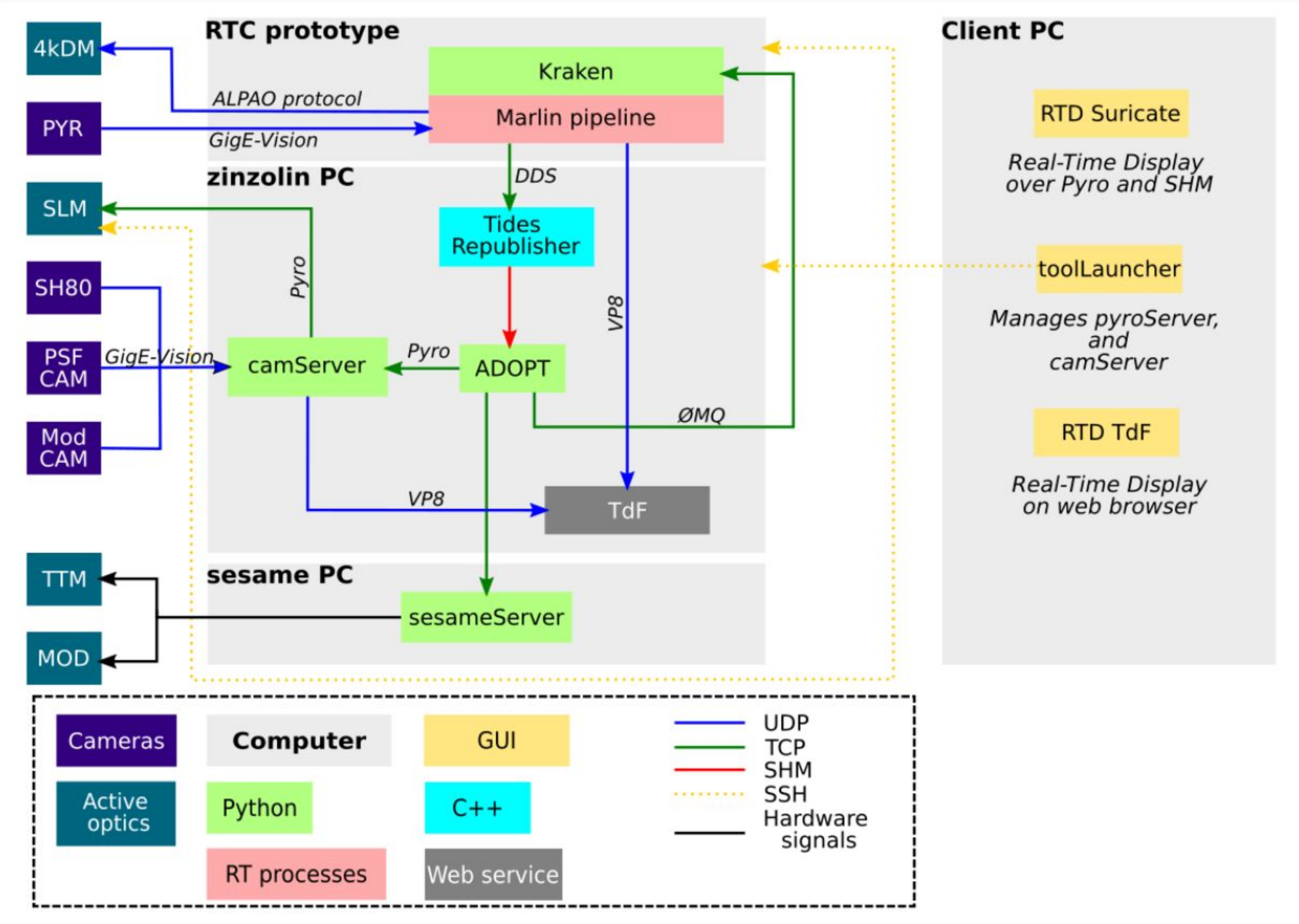
- TIDES:
  - **control / command**: support for ZeroMQ, Pyro, others (TBD)
  - **data distribution**: support DDS, MUDPI, custom protocol, others (TBD)
- SHIPS:
  - Supervisor
  - AO loop optimisation / monitoring
  - Sequencing
  - UI
  - Services
  - …



- Under heavy development today
  - MICADO: pyramid support
  - MAVIS: multi-SH WFS
  - Originally designed to fit with ESO's RTC toolkit but being adapted to Keck needs

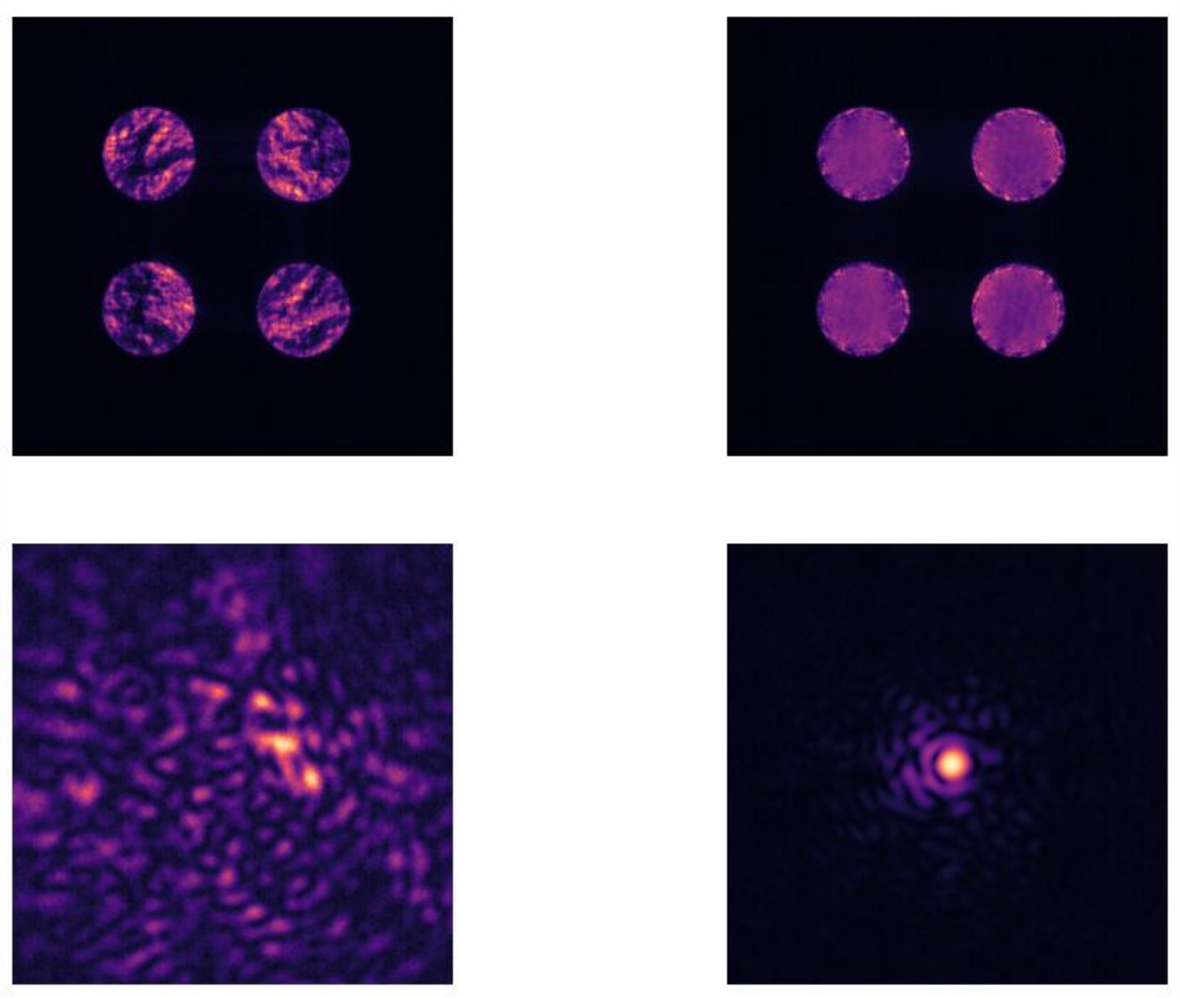
COSMIC IN ACTION

AO BENCH EXPERIMENT @ LESIA



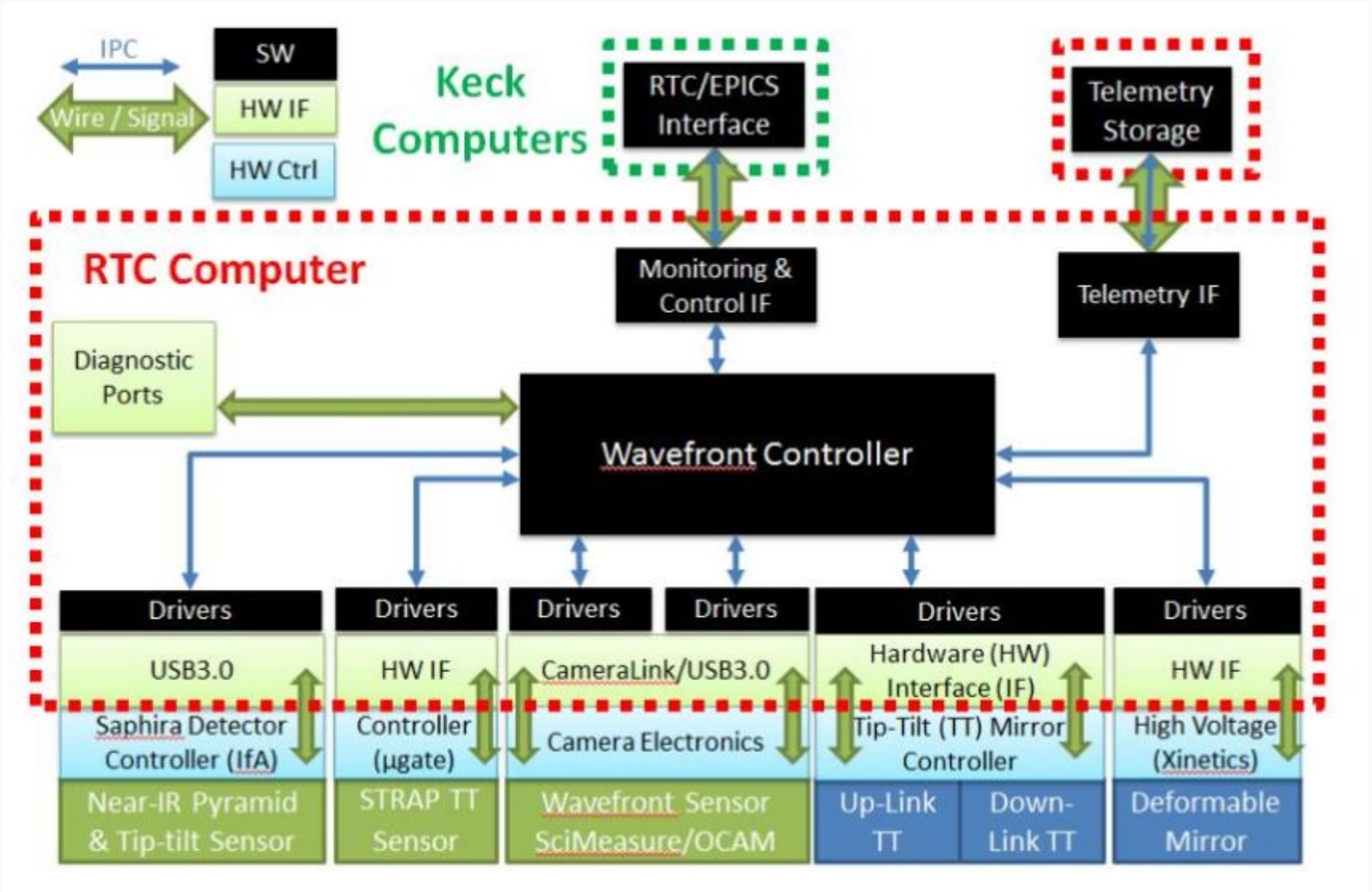
AO BENCH EXPERIMENT @ LESIA

Open / Closed loop operation with pyramid WFS (resp. Left and Right)



COSMIC @ KECK

Ongoing commissioning phase @ Keck



COSMIC @ KECK

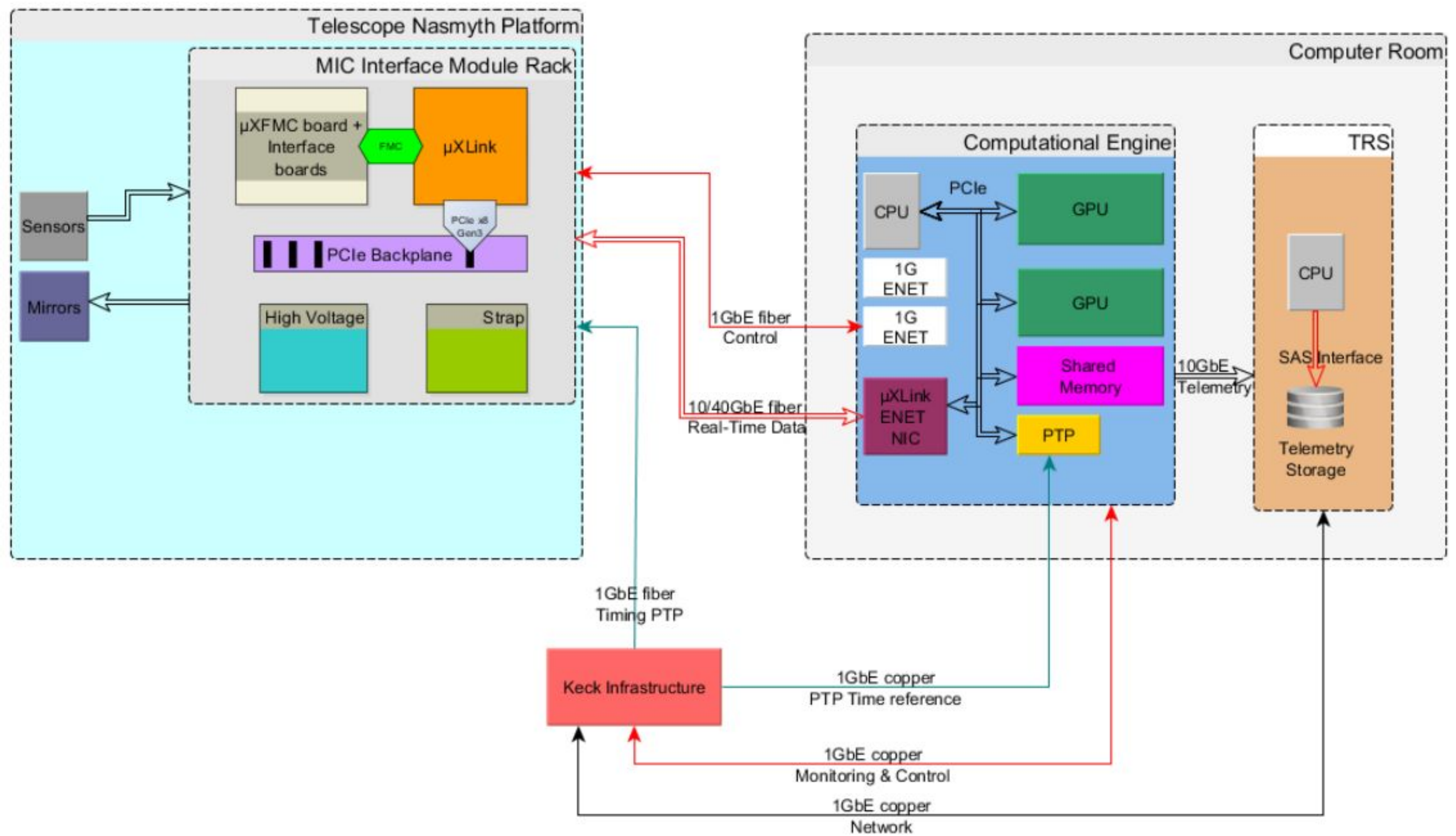
Ongoing commissioning phase @ Keck

- 2 instances: Keck I and Keck II
- Multiple modes supported: SH / pyr / LGS / LTAO
- **First-light last week. Should be offered to the community by end of June**

Mode #	Name	SH WFS		PWS	Tip-tilt (+ focus)			DM		Tip-tilt	
		SciMeasure	OCAM2K		STRAP	TRICK	PWS	Xinetics	MEMS	TTM	UTT
1	NGS w/ SciMeasure	1						1		1	
2	NGS w/ OCAM2K		1					1		1	
3	NGS w/ PWS			1				1		1	
4	NGS w/ PWS & MEMS			1					1	1	
5	LGS w/ SciMeasure & STRAP+TRICK	1			1	1		1		1	1
6	LGS w/ OCAM2K & STRAP+TRICK		1		1	1		1		1	1
7	LGS w/ OCAM2K & PWS		1	1				1		1	1
8	LGS w/ OCAM2K & PWS TT		1				1	1		1	1
9	LTAO w/ STRAP+TRICK		1		1	1		1		1	3
10	LTAO w/ PWS TT		1				1	1		1	3
11	LTAO w/ STRAP+TRICK & new DM		3		1	1		new		1	3

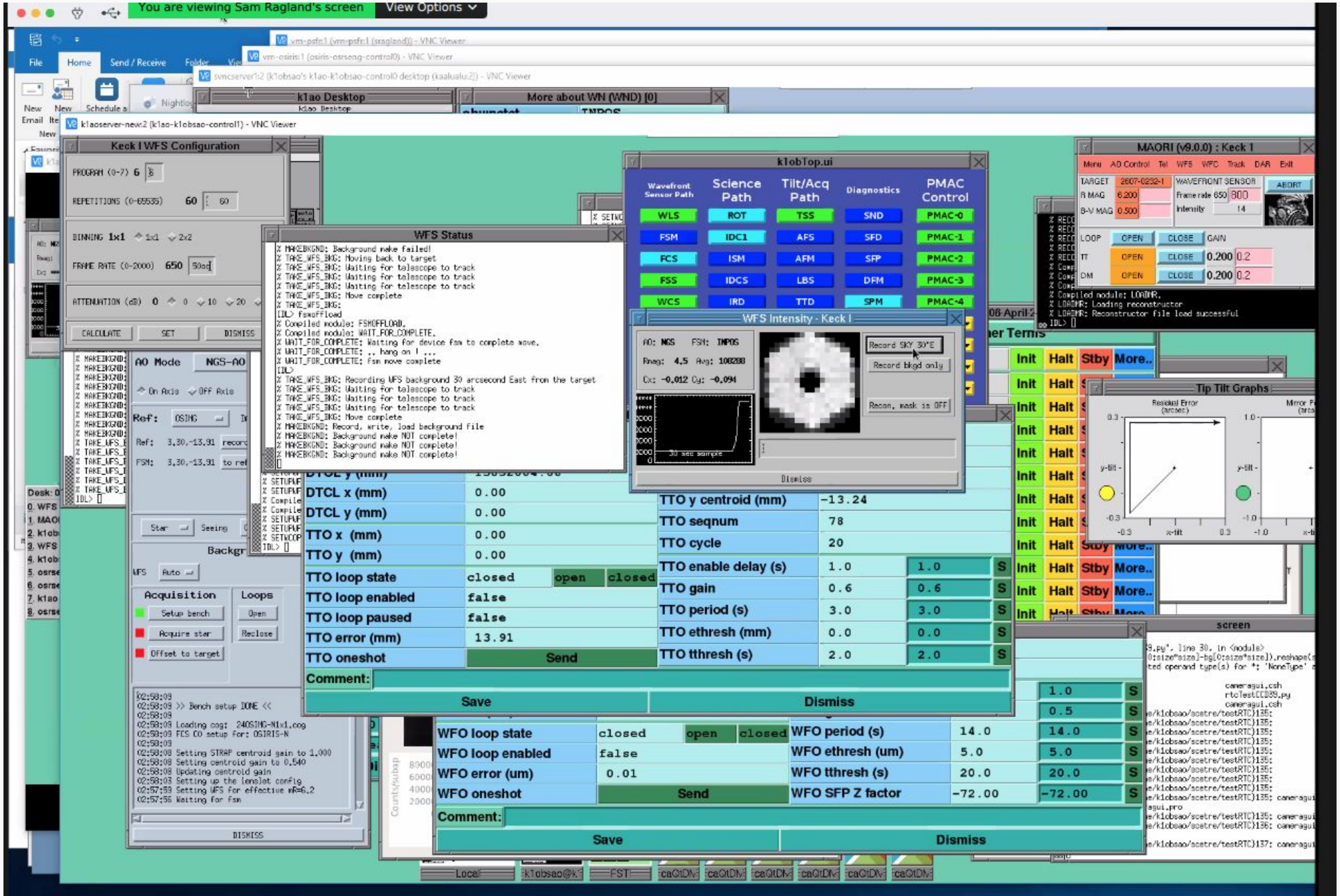
COSMIC @ KECK

Ongoing commissioning phase @ Keck



COSMIC @ KECK

HO and TT loops closed @ > 1 kHz



Non disruptive integration

Current experience with Keck:

- AO system must remain operational throughout the integration of new RTC
- RTC system being integrated over 3 continents: Italy, Australia, Hawaii
- Incremental integration
  - Validate hardware interfaces (with hardware simulators / emulators): Italy
  - Validate compute engine (with a single simulator): Australia
  - Integrate both: 2 instances in both Italy and Australia
  - Ship a third instance to Hawaii
  - Fully integrate 3rd instance in Hawaii during daytime
  - Validate interfaces with hardware during daytime
  - Close loop with hardware during daytime
  - First light
- Total amount of nighttime lost so far: 0.25 night (issues with pipeline)
- Total amount of engineering nights scheduled: 1.5 nights (3 half nights)

Similar strategy with other instruments upgrade

- Need to secure hardware simulators / emulators for the first stage

COSMIC: READY FOR NEXT LEVEL

WHAT'S NEXT ?

GPUs have enabled many progresses in deep learning in recent years

To date the best hardware platform to run DL workloads

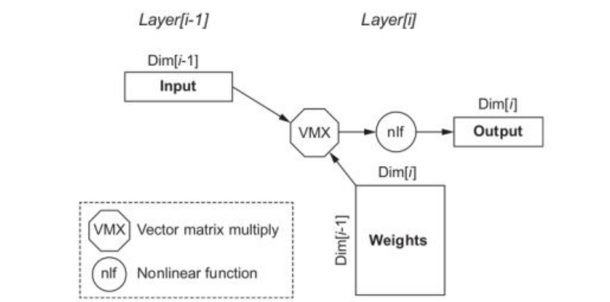


Figure 7.7 MLP showing the input Layer[$i-1$] on the left and the output Layer[i] on the right. ReLU is a popular nonlinear function for MLPs. The dimensions of the input

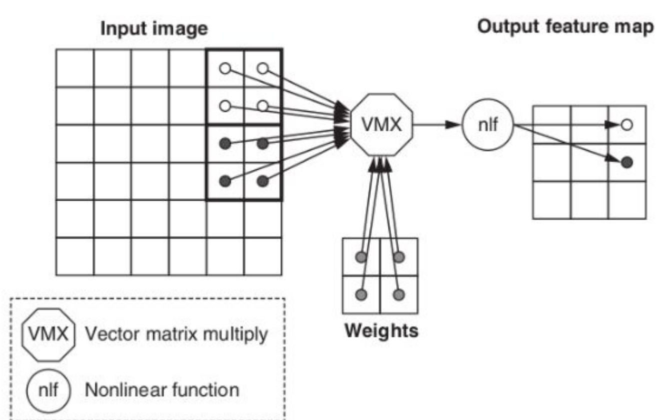


Figure 7.8 Simplified first step of a CNN. In this example, every group of four pixels of

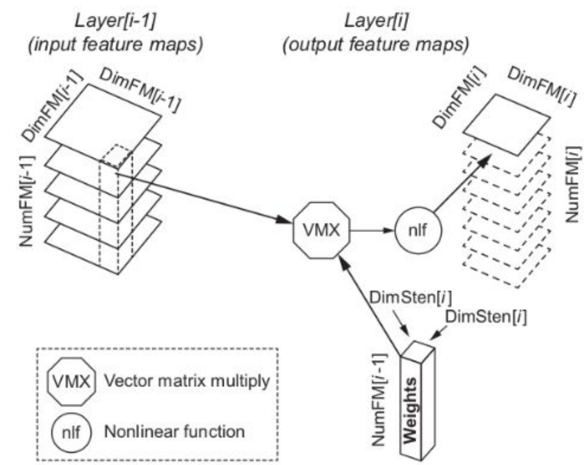
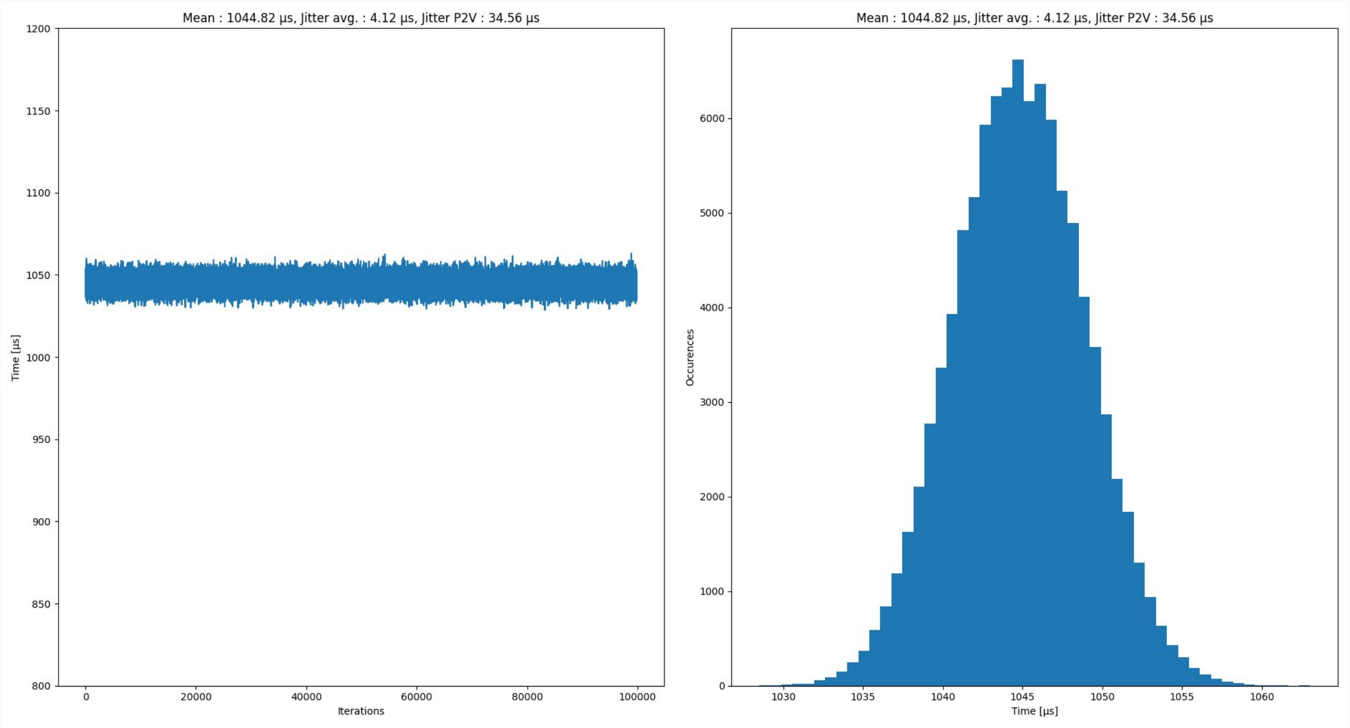


Figure 7.9 CNN general step showing input feature maps of Layer[$i-1$] on the left, the output feature maps of Layer[i] on the right, and a three-dimensional stencil over input feature maps to produce a single output feature map. Each output feature map

WHAT'S NEXT ?

COSMIC RTC platform:

- Delivers performance for model-driven ML approaches for AO: example of MAORY HRTC below (time-to-solution: 1ms on 4 V100 GPUs)
- **Is ready (by design) to run DL workloads**



- **Collab. with Stephen Jones, NVIDIA** on new programming models

Summary

COSMIC is a proven RTC solution for facility instruments

- Already well integrated into ESO ecosystem
- Already at scale for the most challenging dimensioning
- Multiple loops / multiple framerates by design
- Already supporting various AO flavors: SH / Pyr / Laser / tomography

COSMIC is a way to mitigate risk:

- SPARTA obsolescence and permanent failure risks
- Consortium development: addressing schedule risk
- Adaptable to various workloads / approaches (multiple control strategies, multiple instrument configurations)

COSMIC is ready for the next level:

- Scalable
- Future proof: AI friendly

That’s it for today !