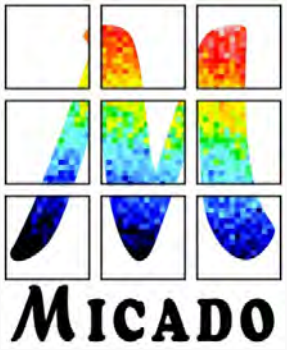


RTC4AO 6

MICADO SCAO RTC:

BUILDING THE ELT FIRST LIGHT RTC

Florian Ferreira

On behalf of the MICADO SCAO RTC Team

cnrs LESIA l'Observatoire de Paris | PSL

SUMMARY

- MICADO SCAO RTC Design overview
- Prototyping activities
  - H-RTC development status
  - RTC Tk integration
  - On-bench activities

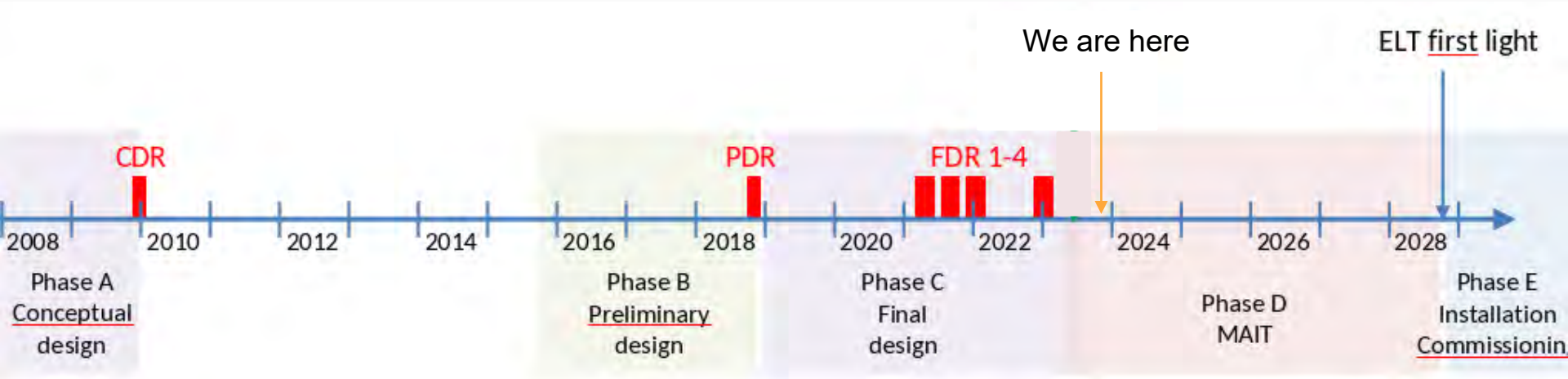
MICADO

- Status after FDR #4 (from FDR board report):

« The review of the final design can be considered complete for the majority of the MICADO sub-systems, with no showstoppers over the design but a few actions, most of them being considered “normal work” and a few being critical ones »

- Agreement from FDR board and ESO to start procurement and manufacturing

MICADO SCHEDULE

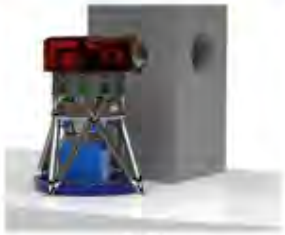


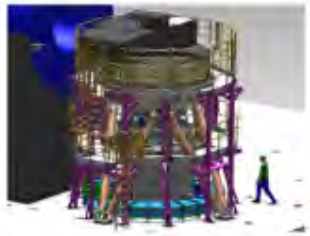

2010


2015


2016

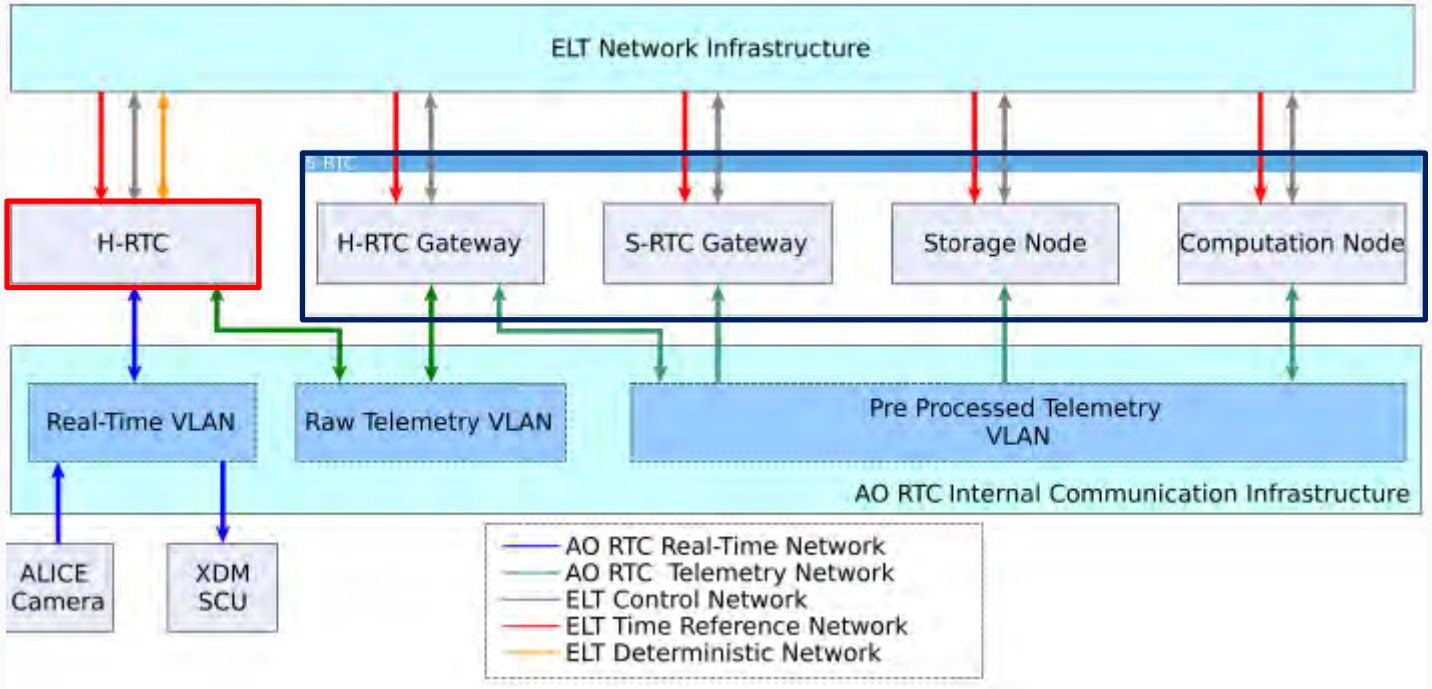

2018


2018


2022

RTC DESIGN OVERVIEW

COSMIC

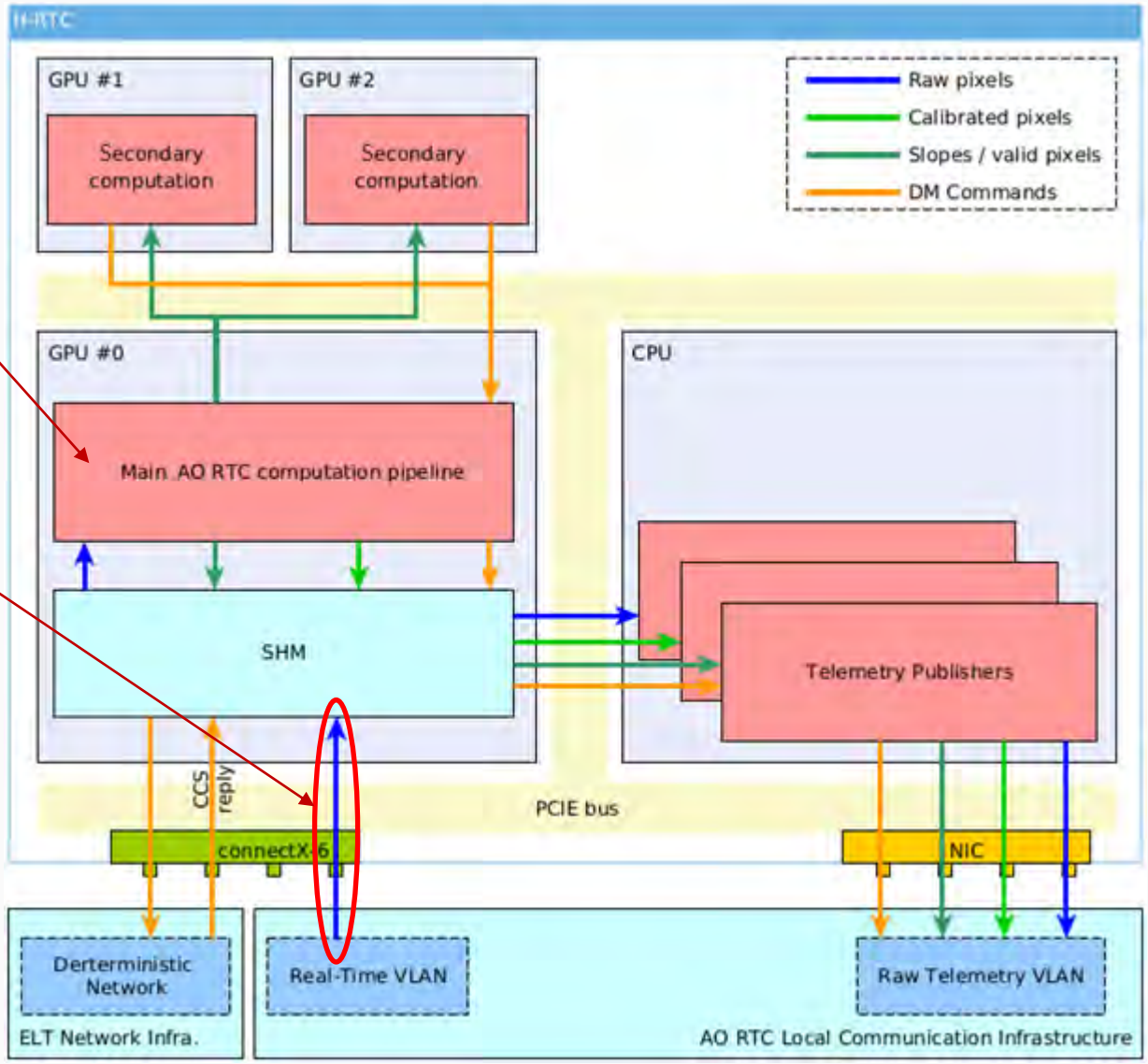


RTC Tk

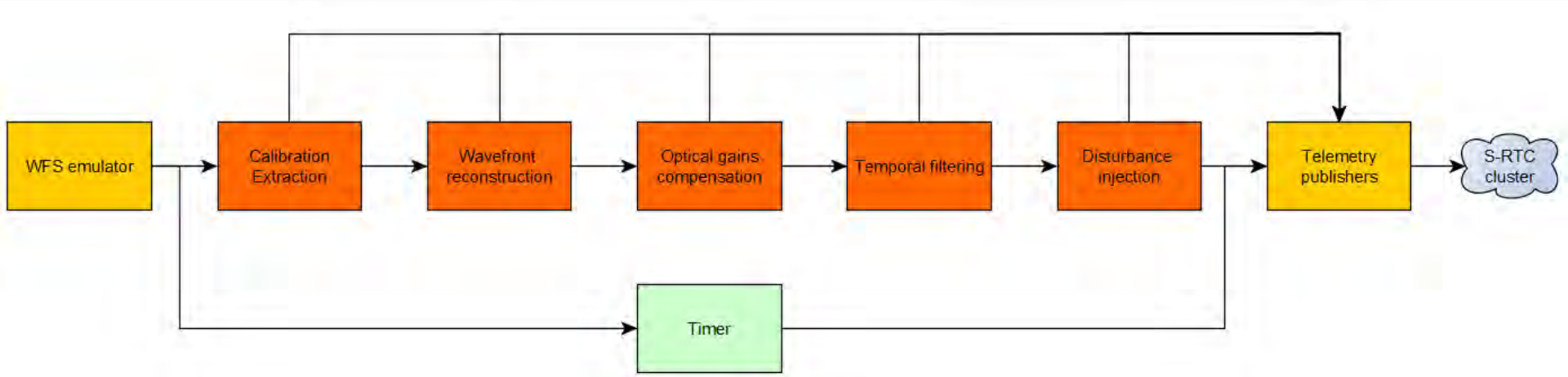
- ELT Standard implementation
- COSMIC-based H-RTC
- ESO RTC Toolkit based S-RTC

H-RTC OVERVIEW

- COSMIC with GPU-based AO pipeline
- GPUDirect RTMS acquisition
- MUDPI Telemetry publishers



H-RTC PERFORMANCE

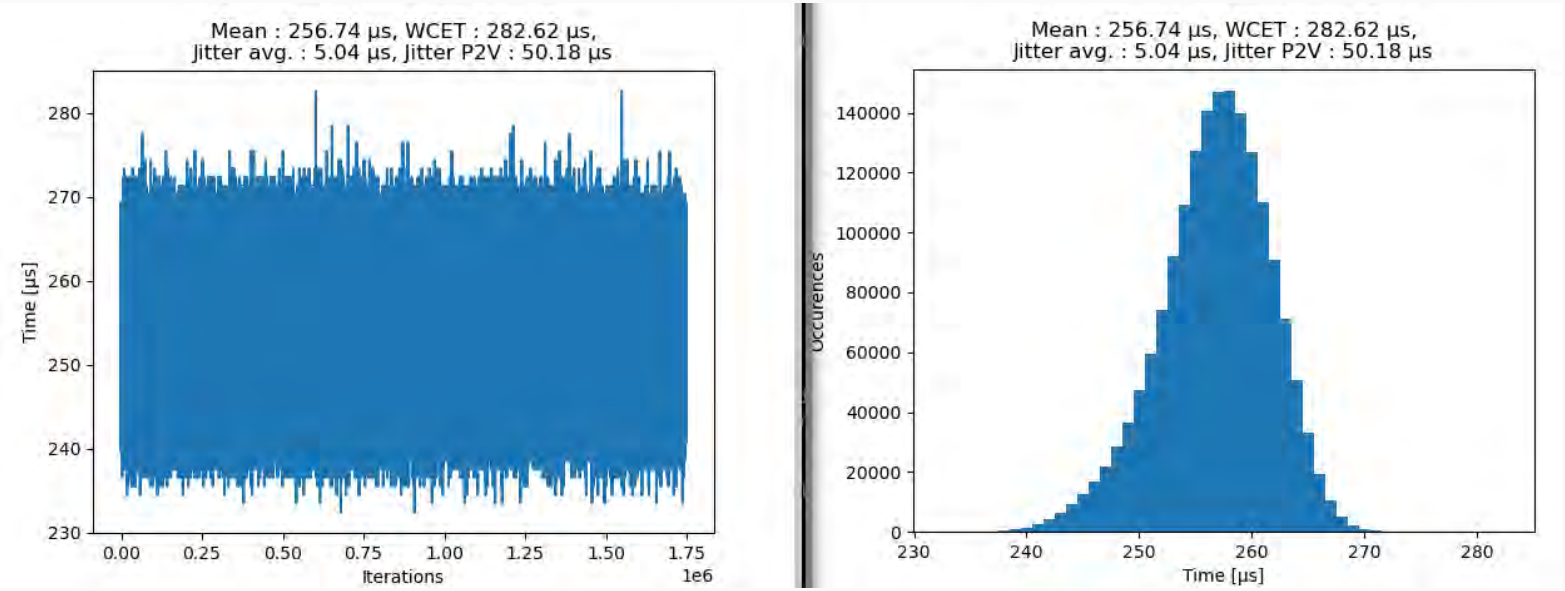


- Emulator mode @ 500 Hz (H-RTC producing ramp images)
- 4,868 modes and 24,416 pixels (PWFS full-pixels algorithm), full pipeline
- Measurement of the H-RTC computation time from image availability to commands computed
- H-RTC server under ELT devEnv 4 (Fedora 34) with 2x NVIDIA A100

USEFUL REAL-TIME METRICS

- **Mean latency:** mean of the measured execution time (as defined just before)
- **Worst-Case Execution Time (WCET):** self-explenatory…
- **Best-Case Execution Time (BCET):** again, you got it…
- **Mean jitter:** mean of the execution time deviation (wrt to the mean latency)
  → i.e. standard deviation of the measured execution time
- **Maximum jitter:** maximum deviation from the mean execution time
  → i.e. WCET – mean latency
- **Peak-to-Peak jitter:** difference between WCET and BCET

H-RTC PERFORMANCE



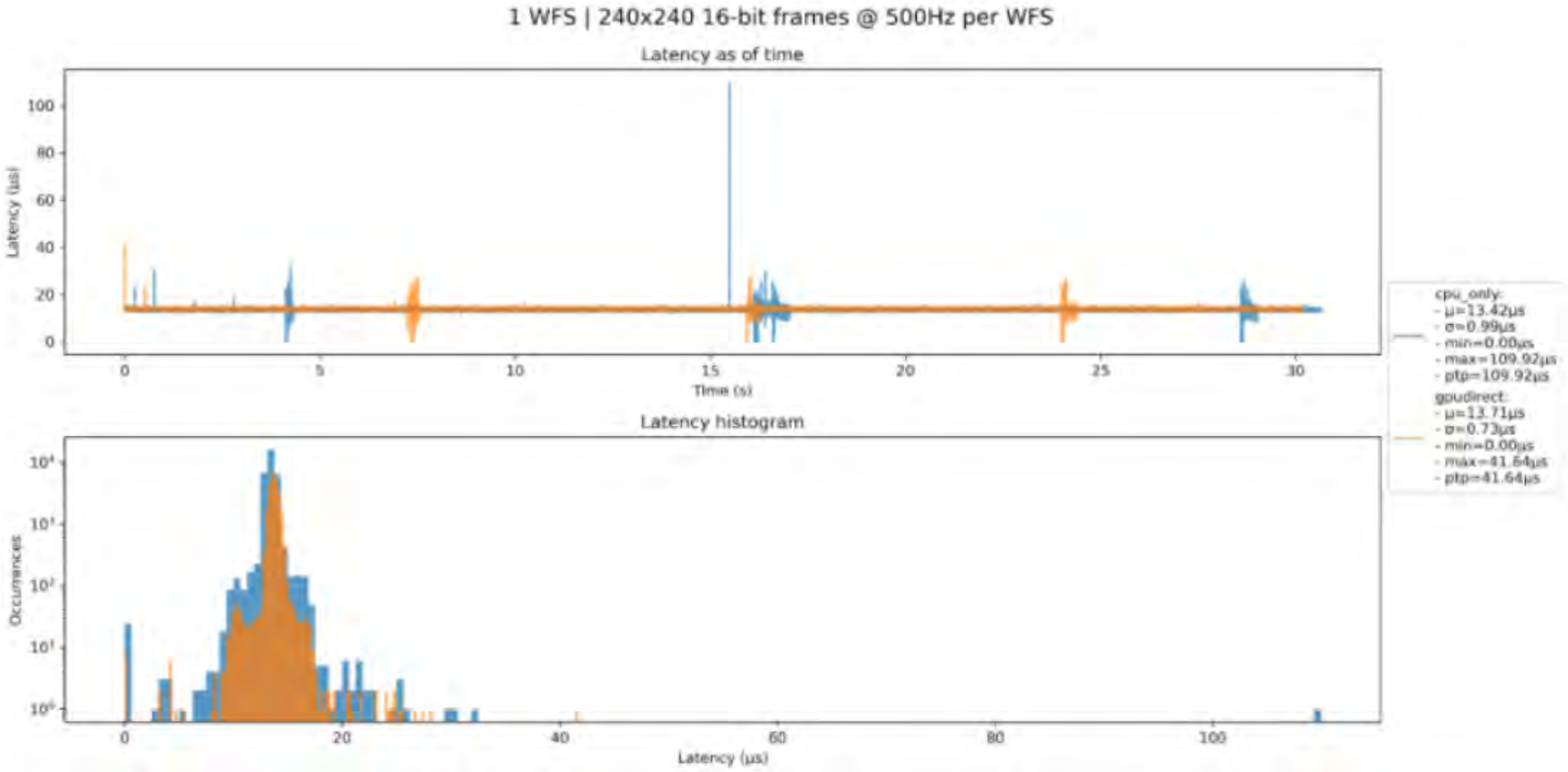
- **Specs:**
  - **lat. < 305 µs**
  - **Max. jitter < 10% mean lat.**
- Mean latency ~257 µs
- WCET ~283 µs
- Average jitter ~5µs
- Max. jitter ~25 µs (~10% lat.)

~2 TB/s of sustained memory bandwidth

60% of max. measured memory bandwidth (1.6 TB/s per GPU)

Trade-off toward maintainability: standard implementation only, no custom CUDA kernels

H-RTC PERFORMANCE



- GPUDirect acquistion mean latency between sent of first packet and receival of the last packets: ~14 µs, max. 42 µs

S-RTC OVERVIEW

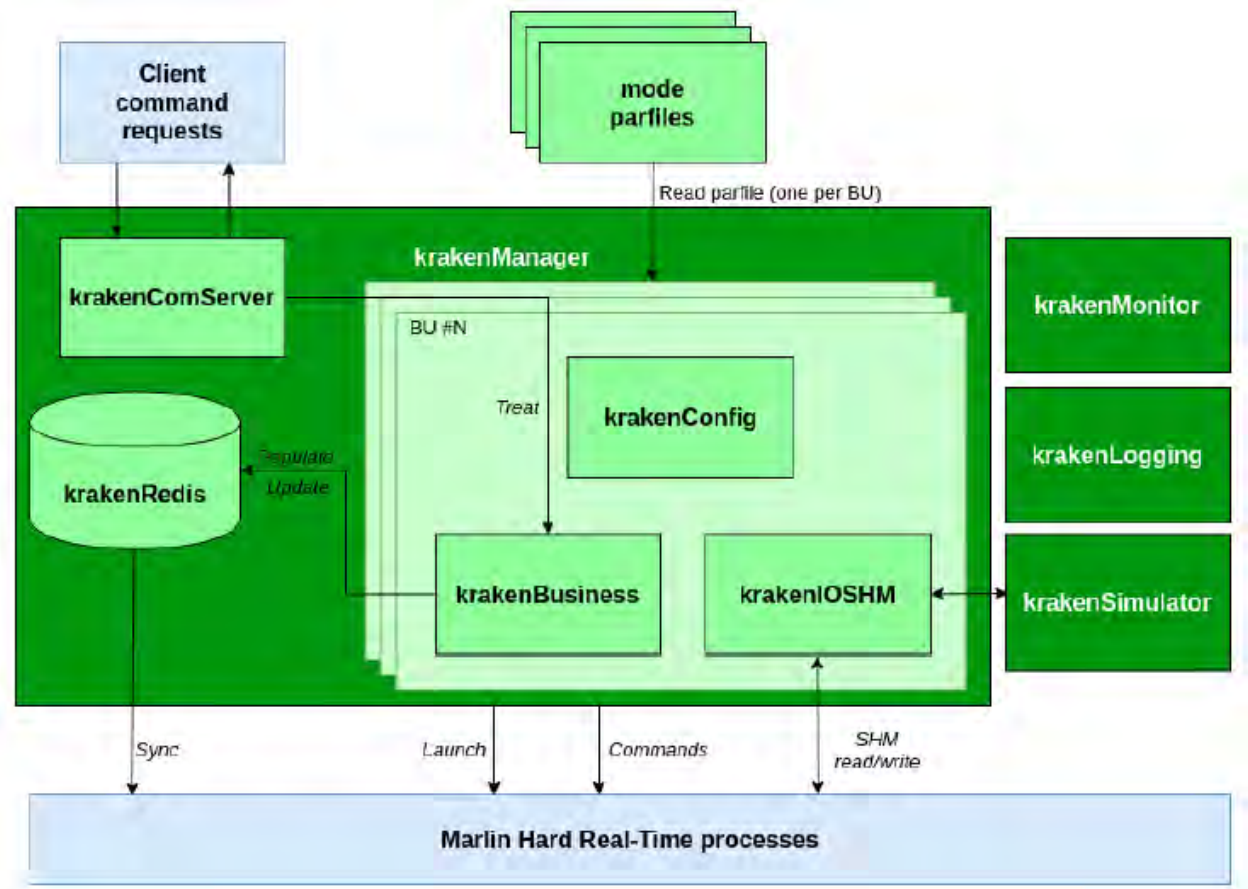
- Based on ESO RTC Toolkit
- …but leveraging in-house S-RTC python-based software
- 4 physical nodes, aligned with ESO RTC Tk reference design:
    - **H-RTC Gateway**: hosting Telemetry Republishers (MUDPI → DDS) and H-RTC Supervisors
    - **S-RTC Gateway**: hosting the RTC supervisor and main connection point with AOCS
    - **Storage node**: hosting Telemetry Recorders
    - **Compute node**: hosting Data Tasks
- Design includes :
    - 12 telemetry topics
    - 33 Data Tasks
    - 12 H-RTC supervisors
- 2 setups for development: full-scale simulation & bench

DATA TASK

- Baseline is to use Python code + pybind11 interpreter
  - Allows easy re-use of existing in-house S-RTC algorithm
  - Allows continuous development & integration along in-house S-RTC software
- AO team easily develops and tests algorithms…
- …which are easily integrated into a Data Task

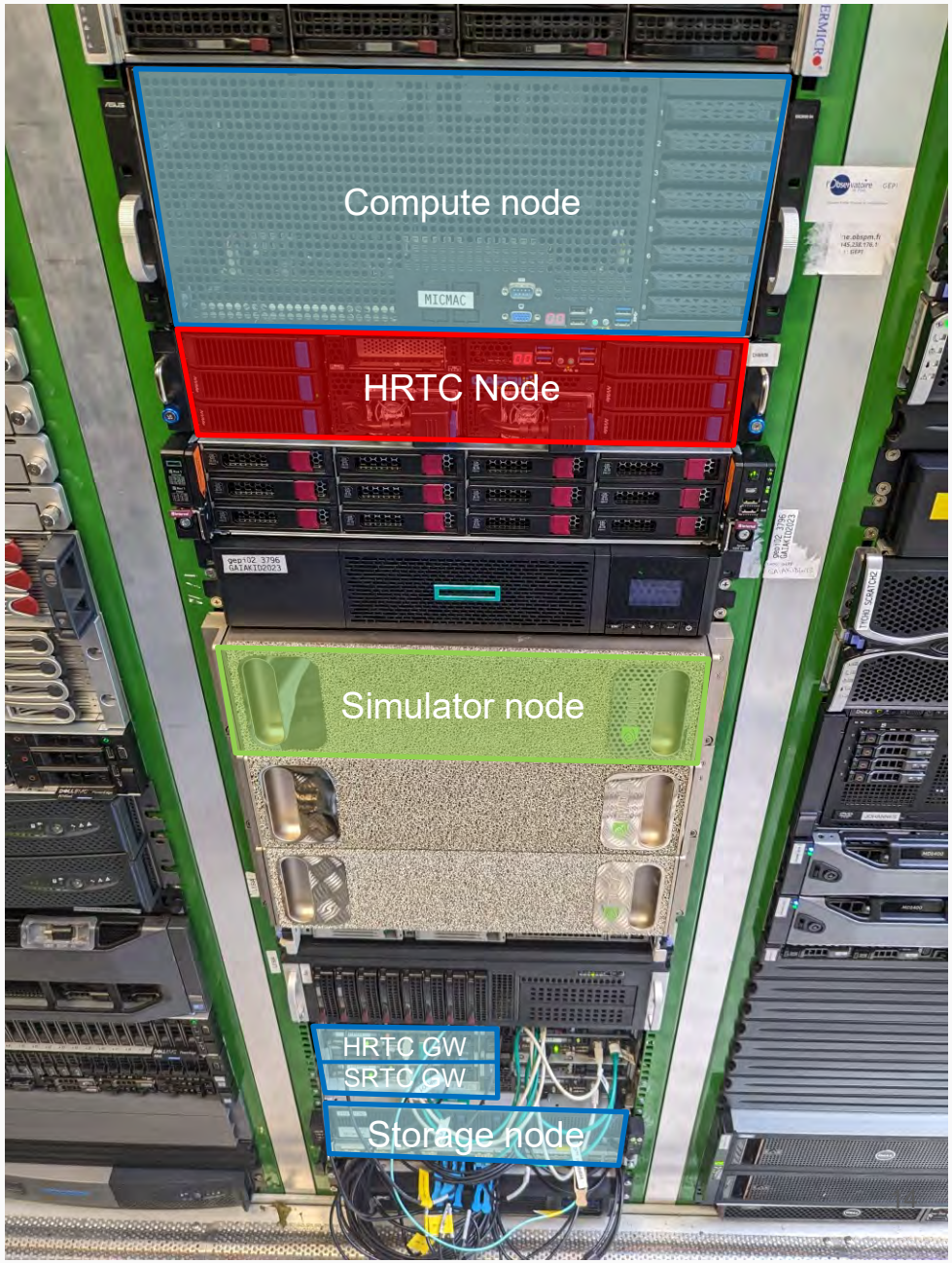
H-RTC / S-RTC COMMAND INTERFACE

- H-RTC Supervisors will rely on COSMIC's Tides software
- Based on ZMQ request/reply + data serialization

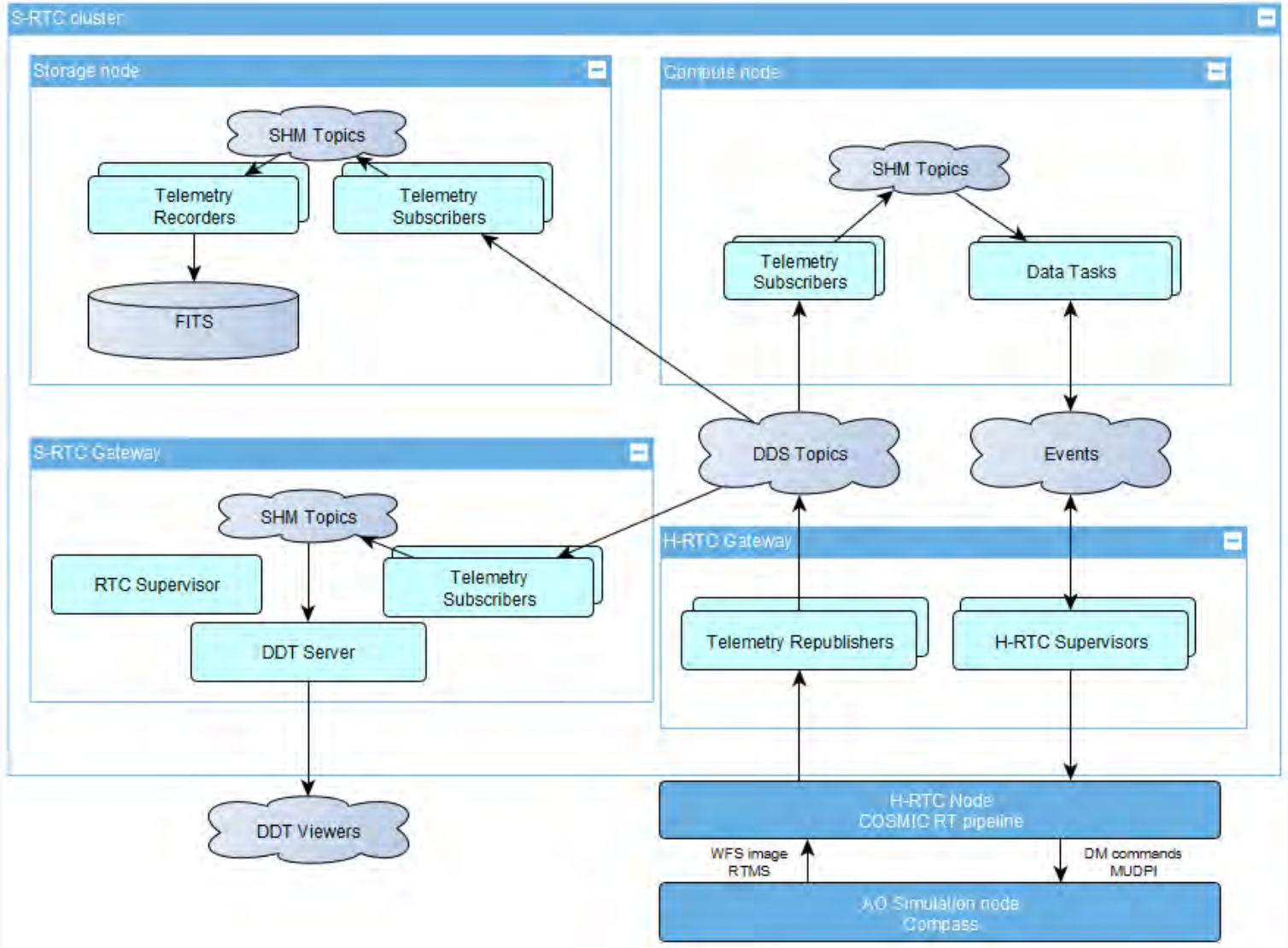


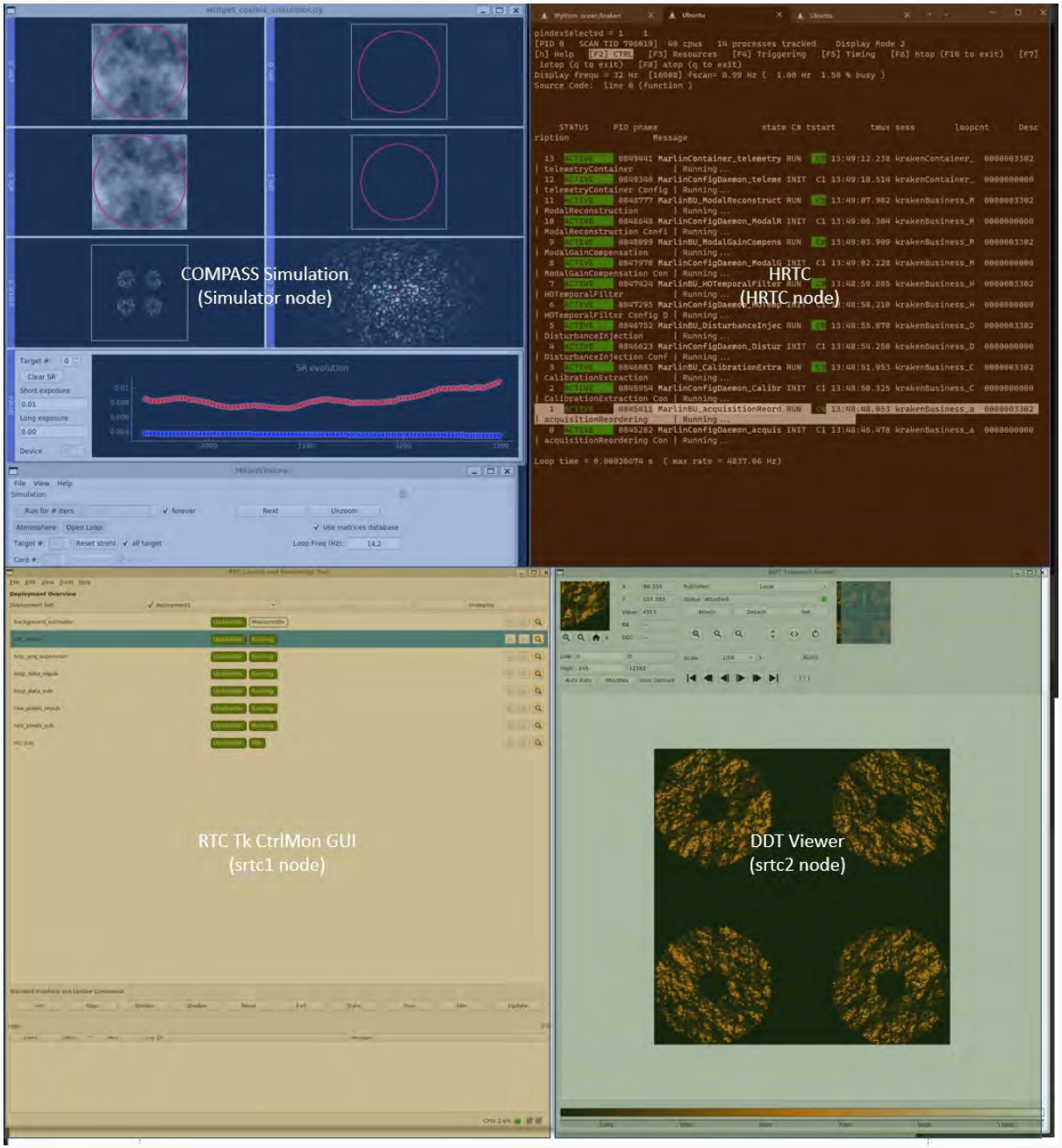
RTC TOOLKIT INTEGRATION

- Prototyping activities full scale setup:
  - 1x WFS simulator (ESO for full-speed, or COMPASS for AO performance)
  - 1x H-RTC
  - 4x S-RTC Nodes
  - 10 Gbe switch for interconnect
- On-bench setup:
  - 1x H-RTC
  - 1x S-RTC Node
  - 1x Workstation
- Allow parallel development wrt to on-bench activities



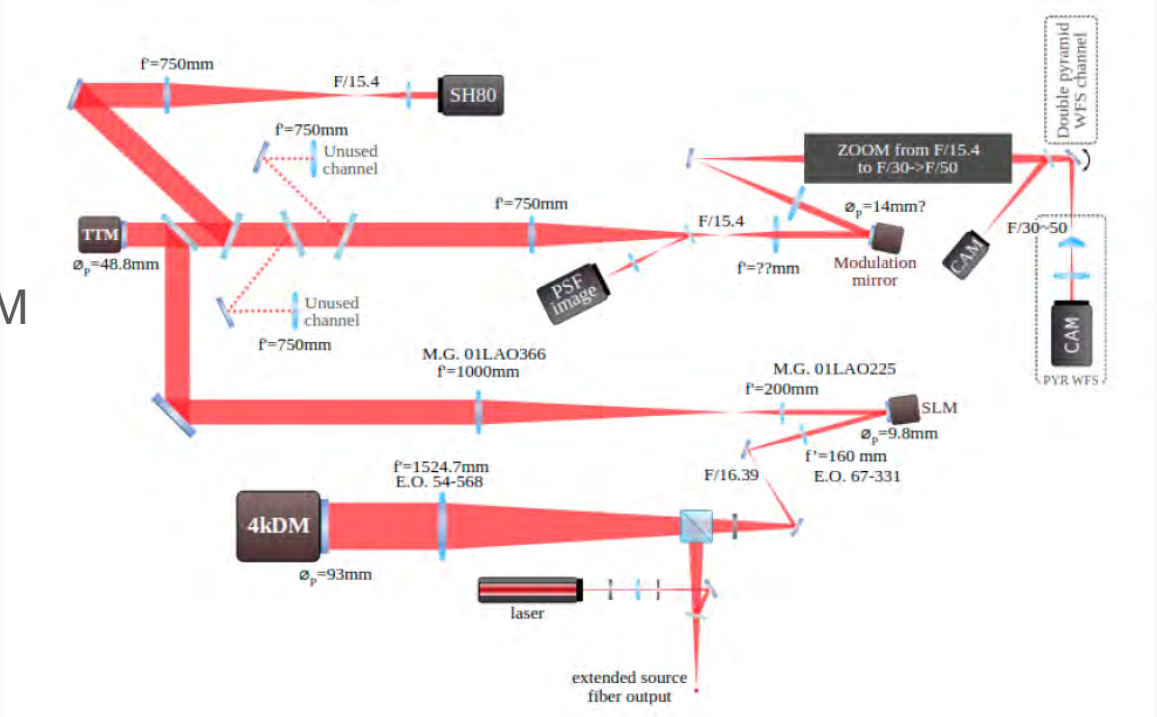
RTC TOOLKIT INTEGRATION: DEPLOYMENT VIEW



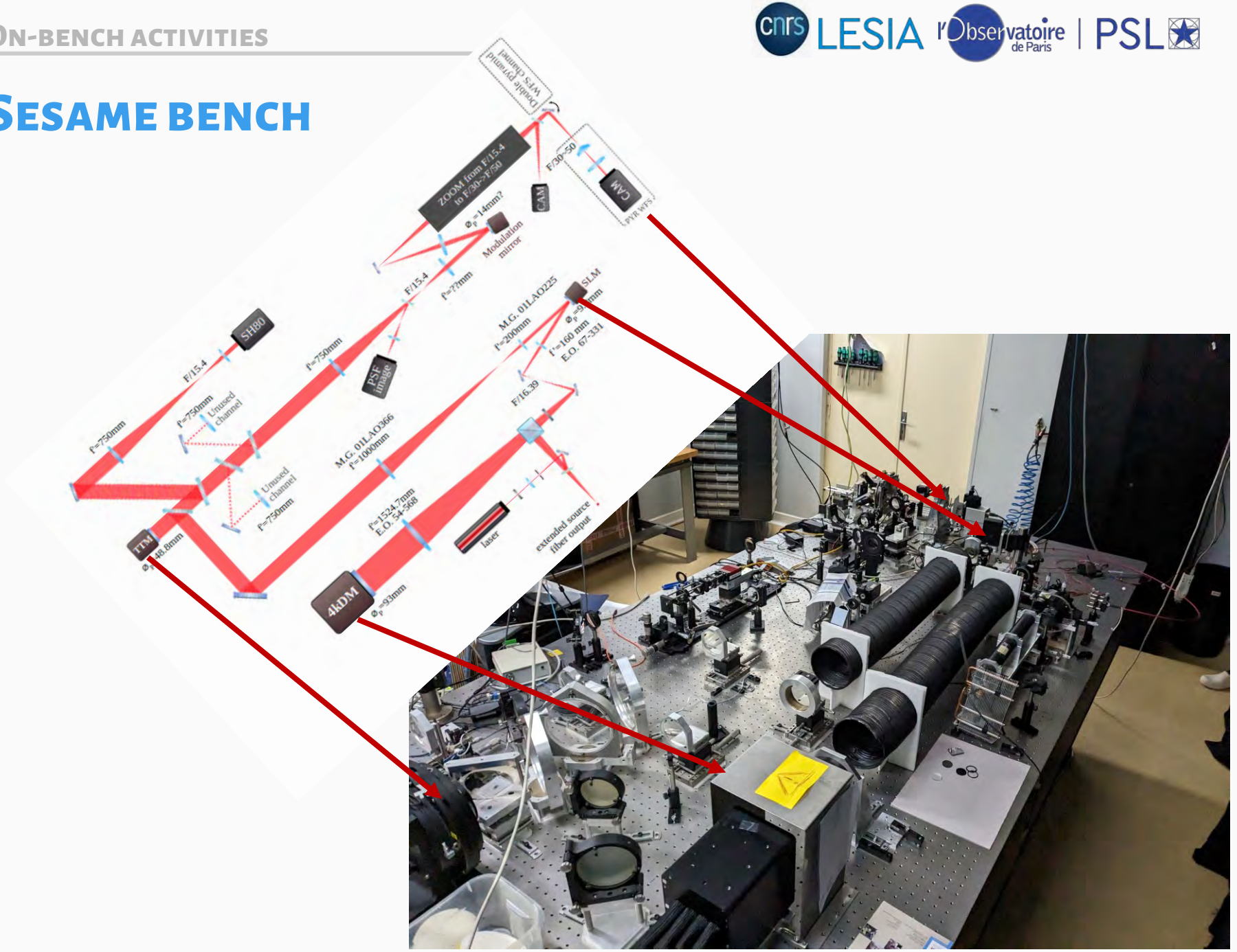

COMPASS Simulation
(Simulator node)
HRTC
(HRTC node)
RTC Tk CtrlMon GUI
(srtc1 node)
DDT Viewer
(srtc2 node)

SESAME BENCH

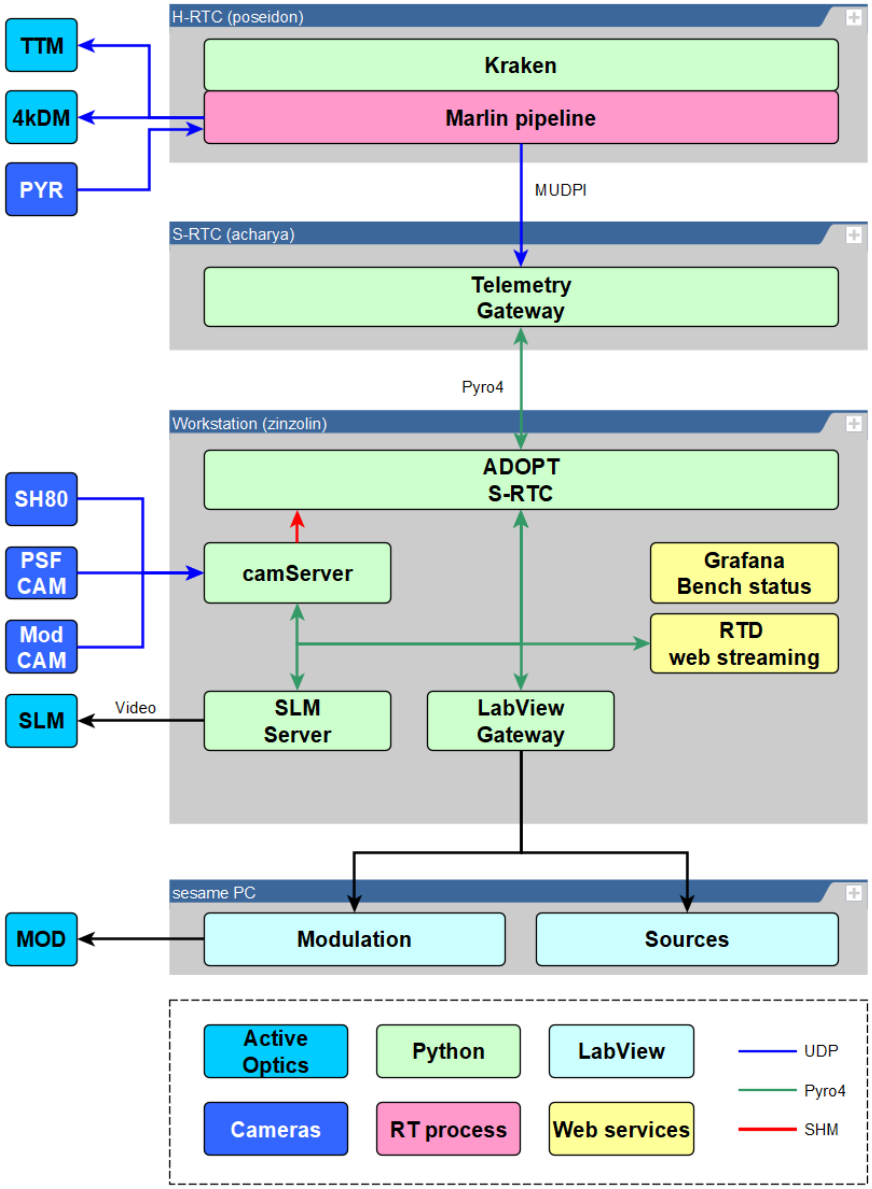
- 633 nm laser point source
- 102x102 pyramid WFS
- 28,320 valid pixels
- 3,228 actuators ALPAO DM
- Tip-tilt mirror
- Pyramid modulation mirror
- PSF camera

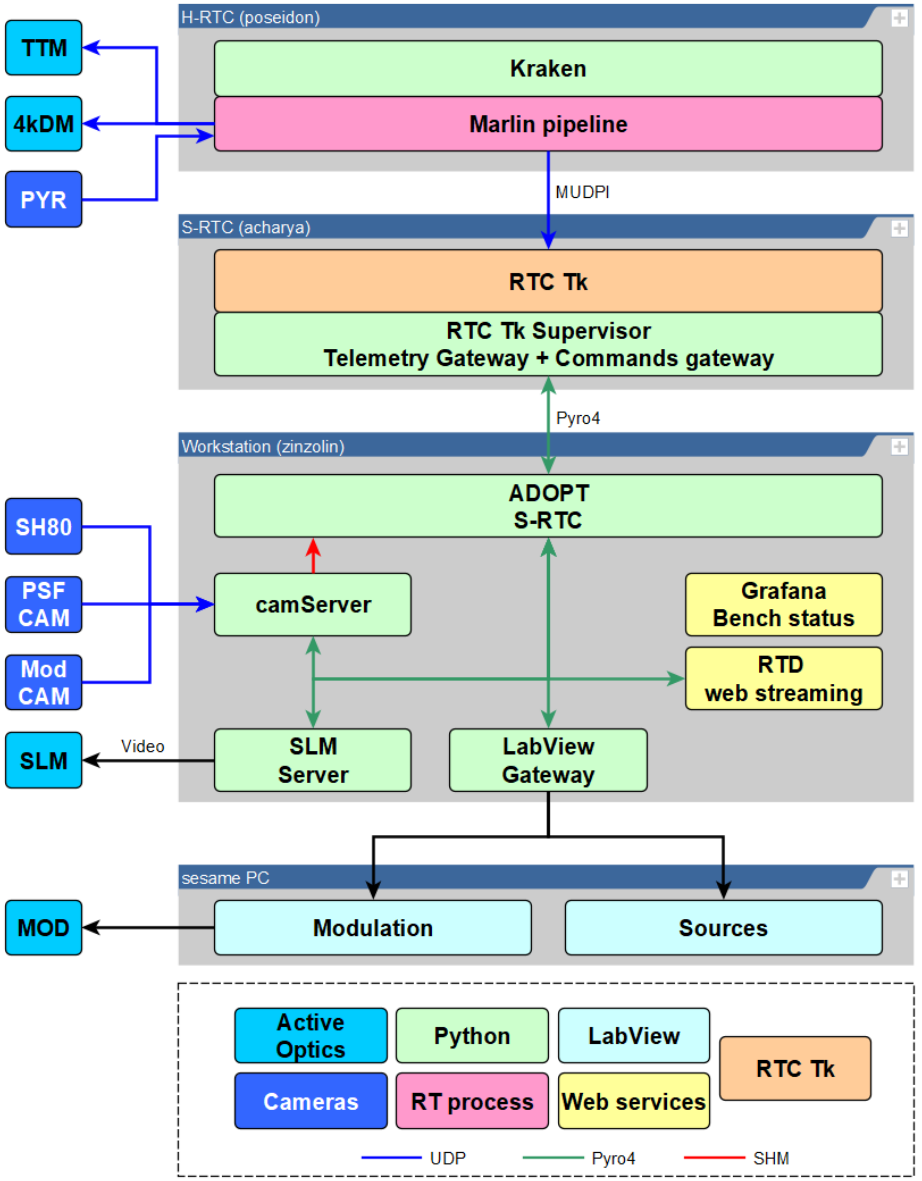


Sesame bench



SESAME BENCH: DEPLOYMENT VIEW

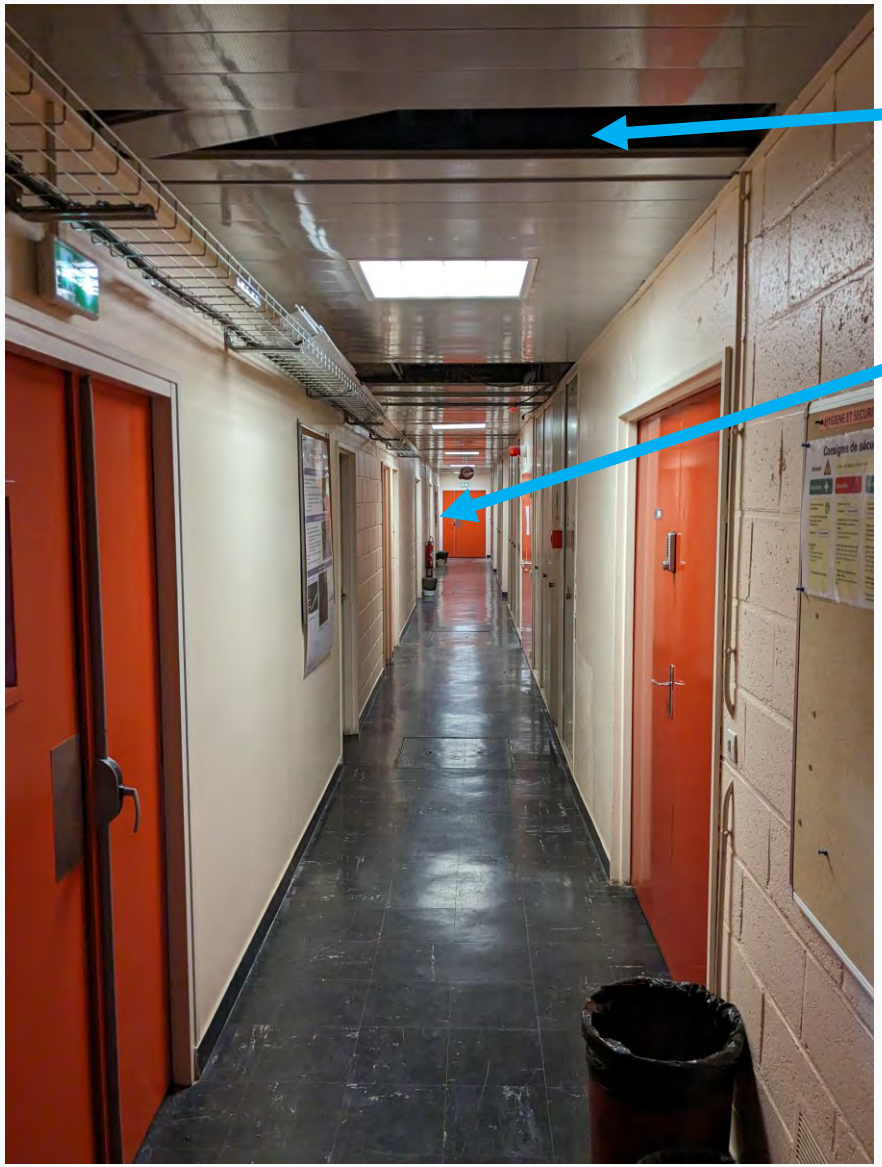




Sesame bench RTC

- 2x 10-cores Intel Xeon E5-2630
- 2x NVIDIA Quadro RTX 8000
  + NVlink bridge
- 1x Dual port ConnectX-6 Dx
- And a lot of noise…
- The AO team:

Sesame bench rtc

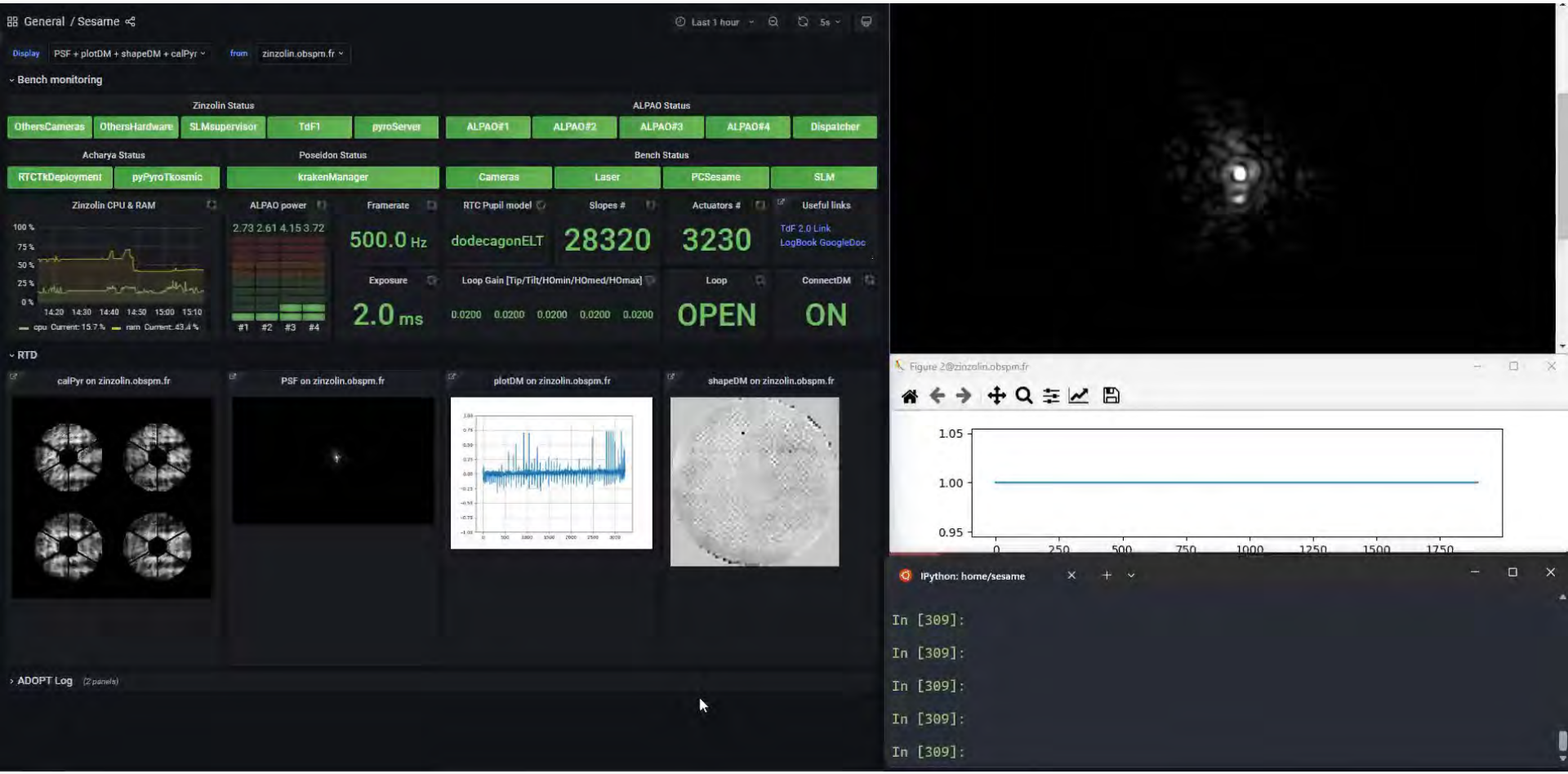


We messed up a bit the ceiling while passing the fibers…

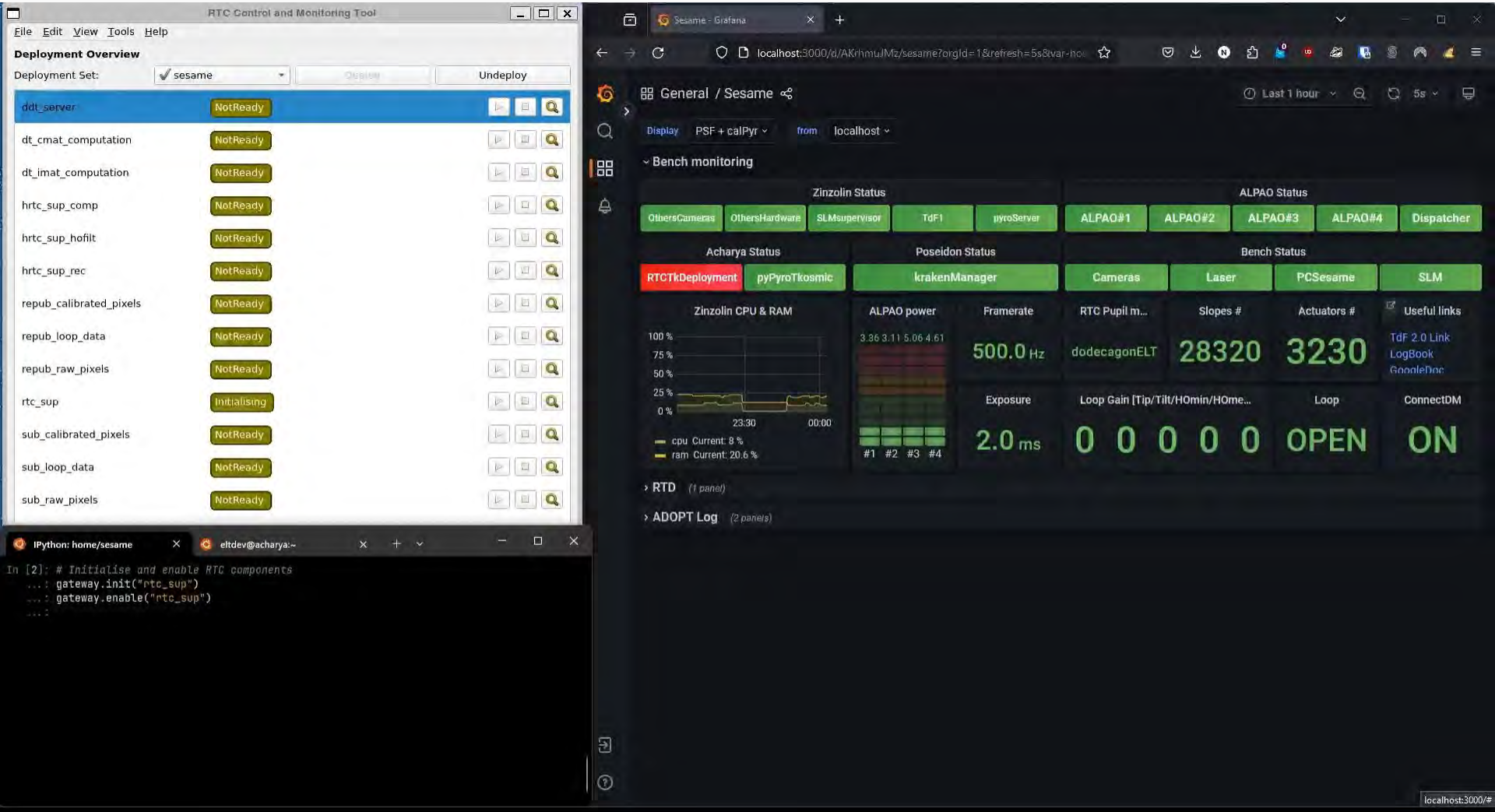
So it ended up there

The AO Team after the rearrangement

Sesame bench: let's close the loop !



SESAME BENCH: LET'S CLOSE THE LOOP AGAIN !



WHAT IF YOU FORGOT TO SHUTDOWN THE BENCH ?

	STATUS	PID	pname	state	C#	tstart	tmux sess	loopcnt	Descr
47	ACTIVE	0897887	MarlinContainer_telemetry	RUN	C0	09:03:31.172	krakenContainer_	0173088972	
46	ACTIVE	0897829	MarlinConfigDaemon_teleme	INIT	C1	09:03:29.530	krakenContainer_	0000000000	
45	ACTIVE	0897187	MarlinContainer_reconstru	RUN	C0	09:03:27.466	krakenContainer_	0173088972	
44	ACTIVE	0897128	MarlinConfigDaemon_recons	INIT	C1	09:03:25.813	krakenContainer_	0000000000	
43	ACTIVE	0896443	MarlinBU_kacou	RUN	C0	09:03:23.975	krakenBusiness_k	0173088972	
42	ACTIVE	0896365	MarlinConfigDaemon_kacou	INIT	C1	09:03:22.308	krakenBusiness_k	0000000000	
41	ACTIVE	0895717	MarlinBU_acquisitionReord	RUN	C1	09:03:20.461	krakenBusiness_a	0173088987	
40	ACTIVE	0895648	MarlinConfigDaemon_acquis	INIT	C1	09:03:18.837	krakenBusiness_a	0000000000	

- 173M frames @ 500 Hz → 4 days run ! (Yes, it was a long week-end…)
- 15 frames lost → 0.0000087 % loss
- Hopefully, AO Team said it was barely acceptable…

Conclusion & future works

- H-RTC development is well advanced: already on bench
  - Performance on specs
  - Reliable
  - Maintainable
- Current activities mainly focused on S-RTC: RTC Tk integration & testing
- Next steps:
  - Logging strategy for H-RTC: CII integration ?
  - Continue developing RTC Tk components
  - Development on simulation setup & Testing on bench