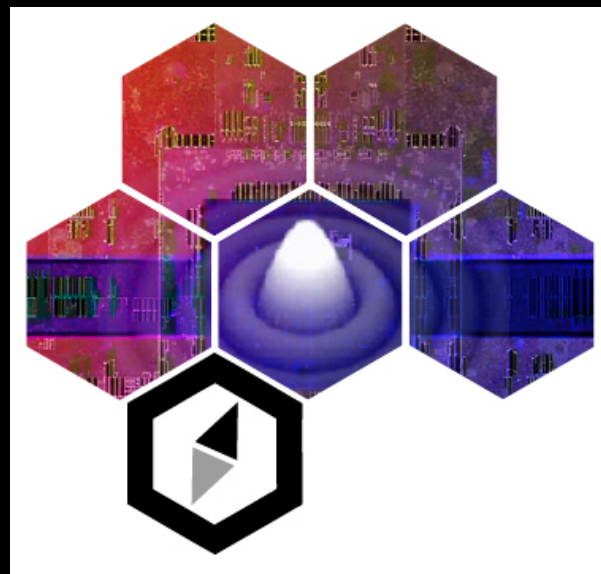


Contrôle temps-réel de l'optique adaptative à l'aide de GPU

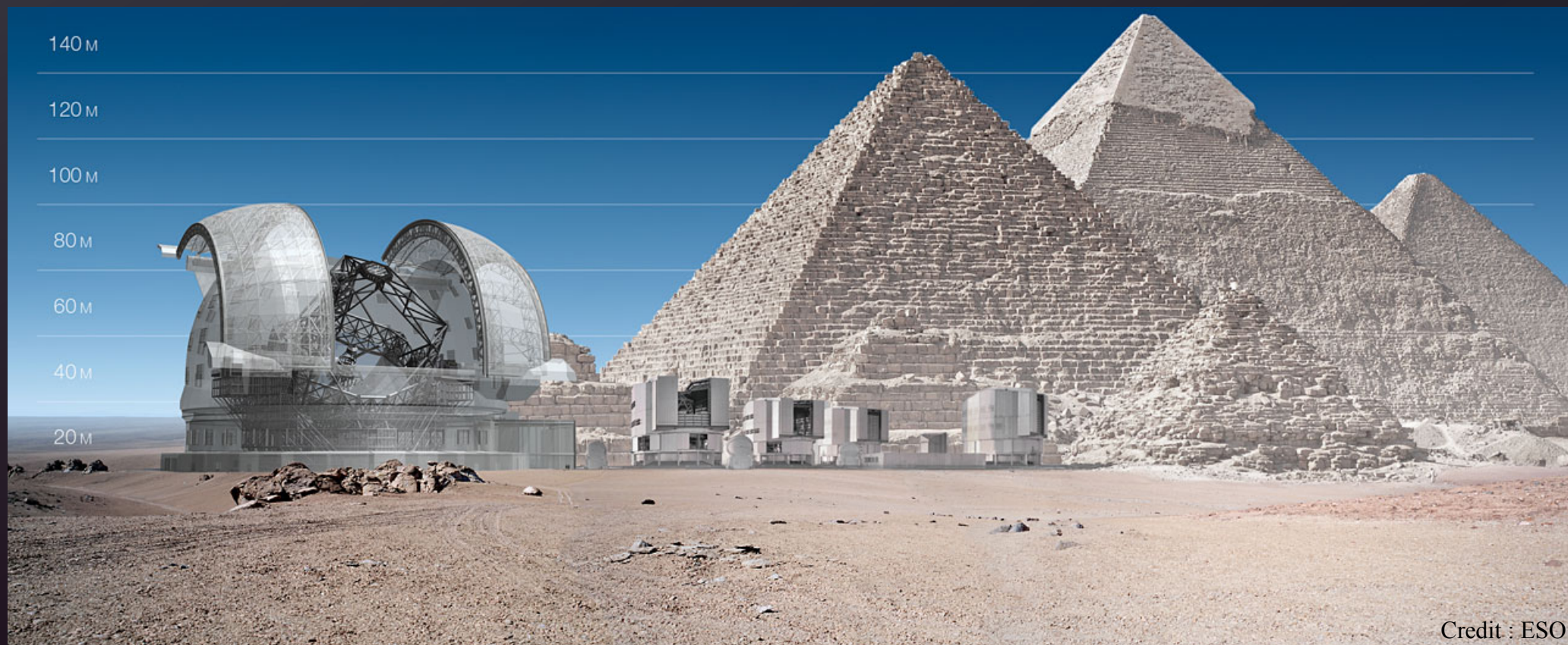
Damien Gratadour, LESIA Observatoire de Paris



- ◉ L'E-ELT et l'Optique Adaptative
- ◉ Unifier les simulations et le cœur temps-réel
- ◉ Le projet COMPASS
- ◉ La plateforme de simulation
- ◉ La plateforme temps-réel
- ◉ Le projet Green FLASH
- ◉ Échelle extrême : reconstructeurs tomographiques pour la MOAO
- ◉ Conclusions / perspectives

European Extremely Large Telescope (E-ELT)

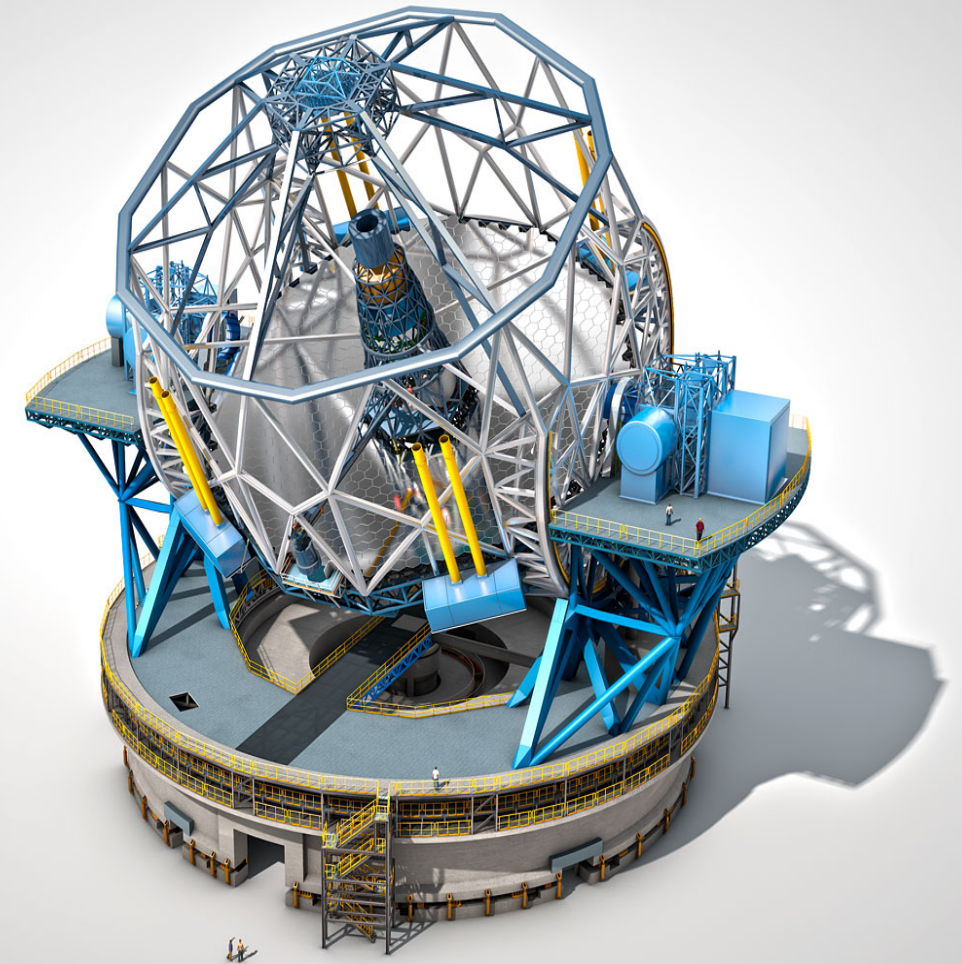
- ◉ Télescope de 40m de diamètre



- ◉ Miroir primaire : ~ 800 segments de 1.4m de diamètre, résolution théorique dans l'IR proche : 10 mas, i.e. 2×10^5 plus petit que la pleine lune (~ 30 arcminutes)
- ◉ Environ 1000 m^2 de surface collectrice, i.e. 15 fois plus sensible que les télescopes de la classe 10m en opération
- ◉ Premier télescope entièrement adaptatif

E-ELT

- ◉ Extremely Large !
 - ◉ 100m dome, 2800 tones structure rotating @ 360°, seismic safe (Chile)
 - ◉ 1.2 G€ project, first light foreseen in 2024
 - ◉ Construction led by ESO (European Southern Observatory), international organisation funded by 15 European countries
 - ◉ Telescope components + science instruments built by european research labs + industrial partners
 - ◉ Non-European partners : Chile (site) and Brazil (under negociations to become an ESO funding partner)



Credit : ESO

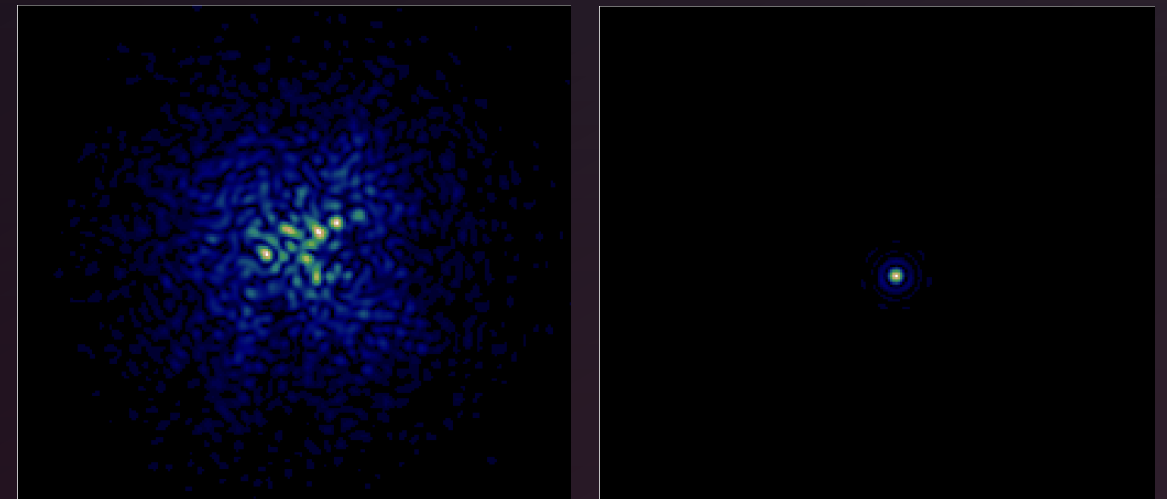
Optique adaptative

- Observer l'Univers à travers la turbulence atmosphérique
 - Atmosphère terrestre = mélange de masses d'air à différentes températures (variations locale $< 1^\circ\text{C}$) = « bulles d'air » avec différents indexes de réfraction
 - La traversée de l'atmosphère déforme le front d'onde = réduction de la qualité d'image
 - La résolution n'est plus limitée par le diamètre du télescope mais par la qualité de l'atmosphère (equiv. à un télescope de 20cm sur les meilleurs sites astronomiques, e.g. Hawaii, Chile)

- Compenser dynamiquement les aberrations

- Grâce à un ou plusieurs miroirs déformables : optique adaptative

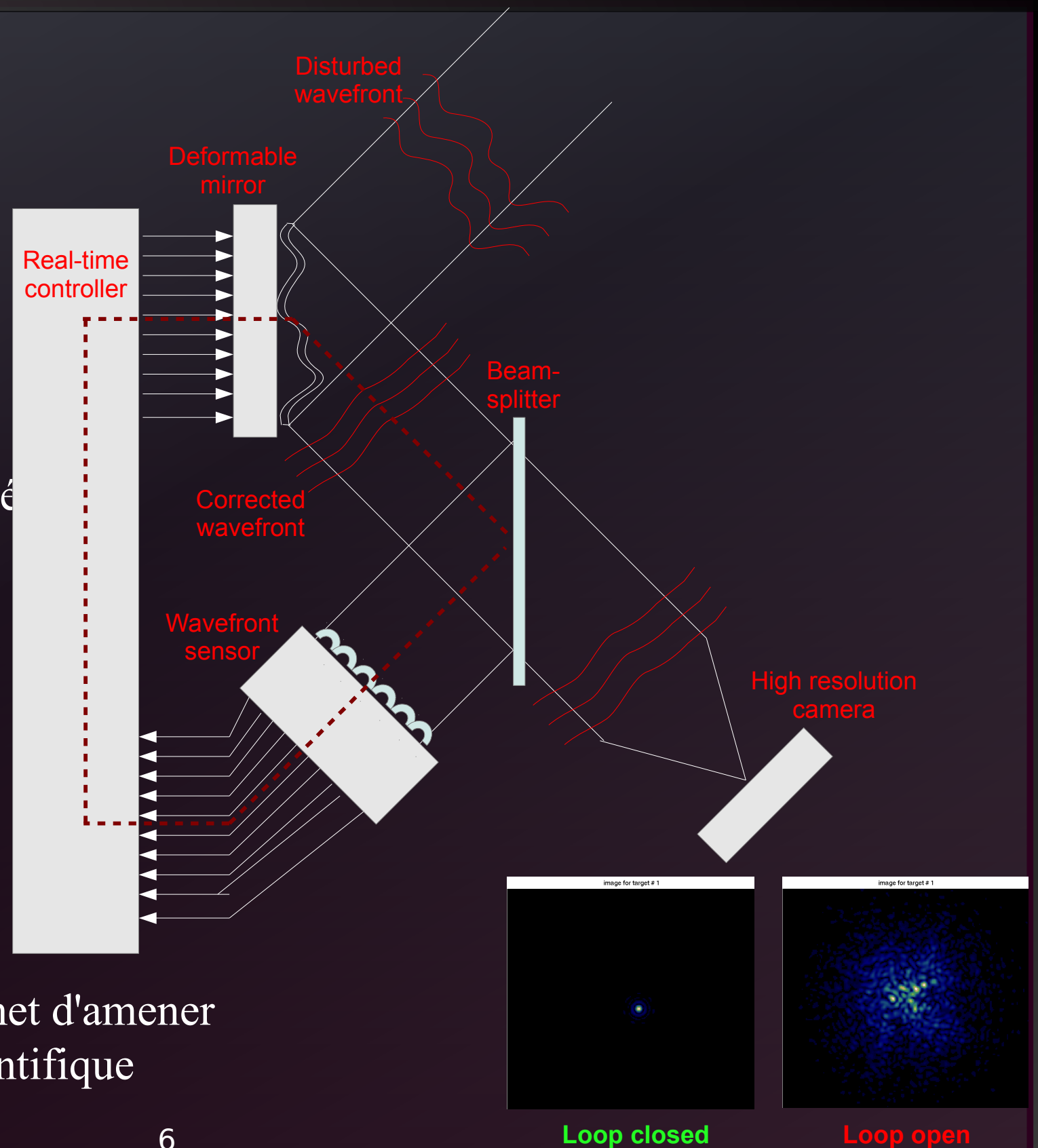
- Applications non limitées à l'astronomie : imagerie biomédicale, focalisation de laser de puissance (telecom, manufacturing, etc ..), satellite / space debris tracking, applications militaires, etc ...



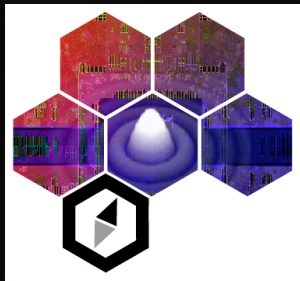
Optique adaptative

Principe

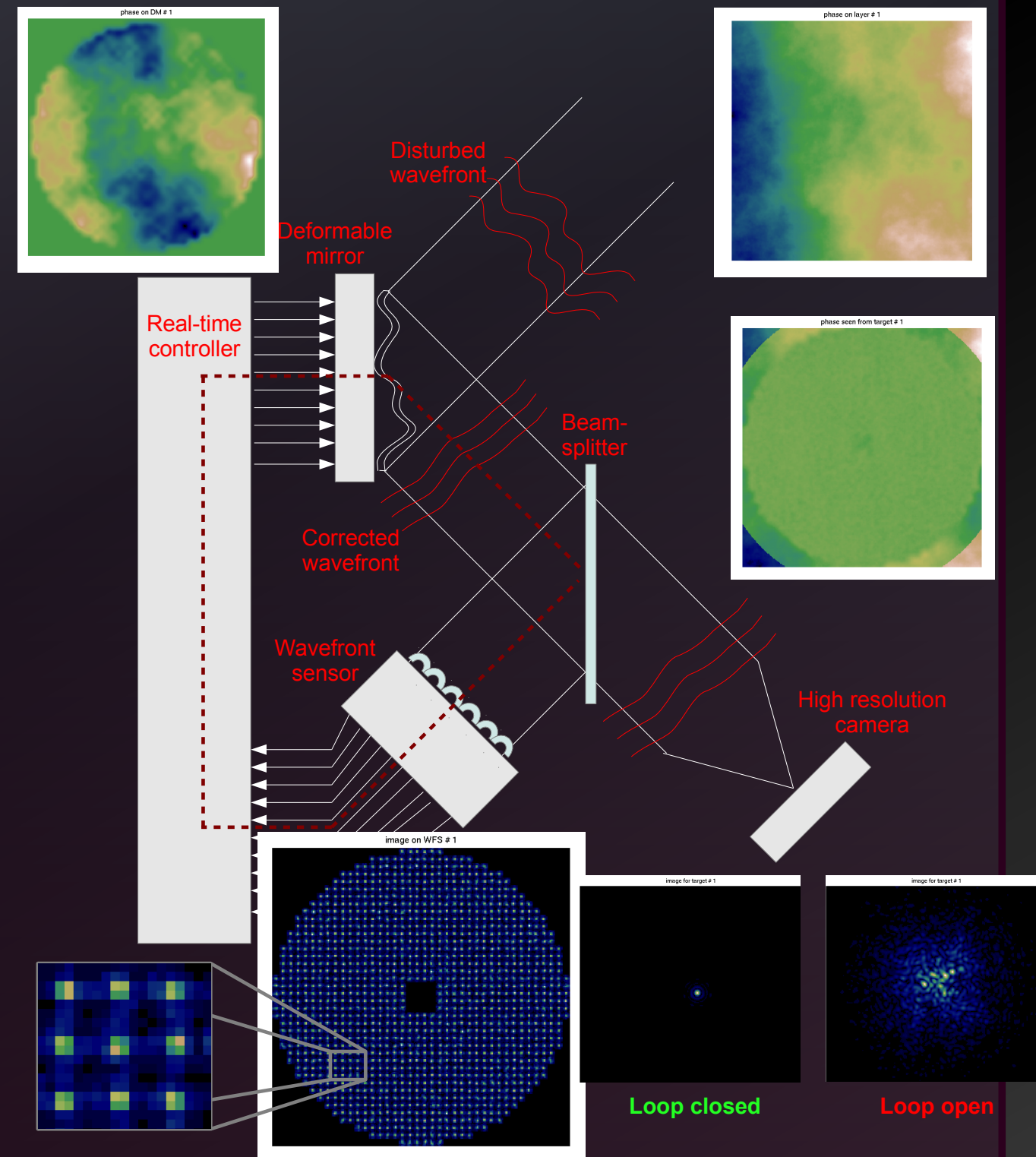
- Analyser les perturbations du front d'onde (analyseur de surface d'onde – WFS)
- Calculer (en temps réel) la compensation à appliquer en utilisant un ordinateur temps-réel (RTC)
- Appliquer la compensation en utilisant un miroir déformable (DM)
- Système bouclé, opérations en temps-réel (latence $\sim 1\text{ms}$)
- Un séparateur de faisceau permet d'amener la lumière sur l'instrument scientifique



La simulation de l'OA

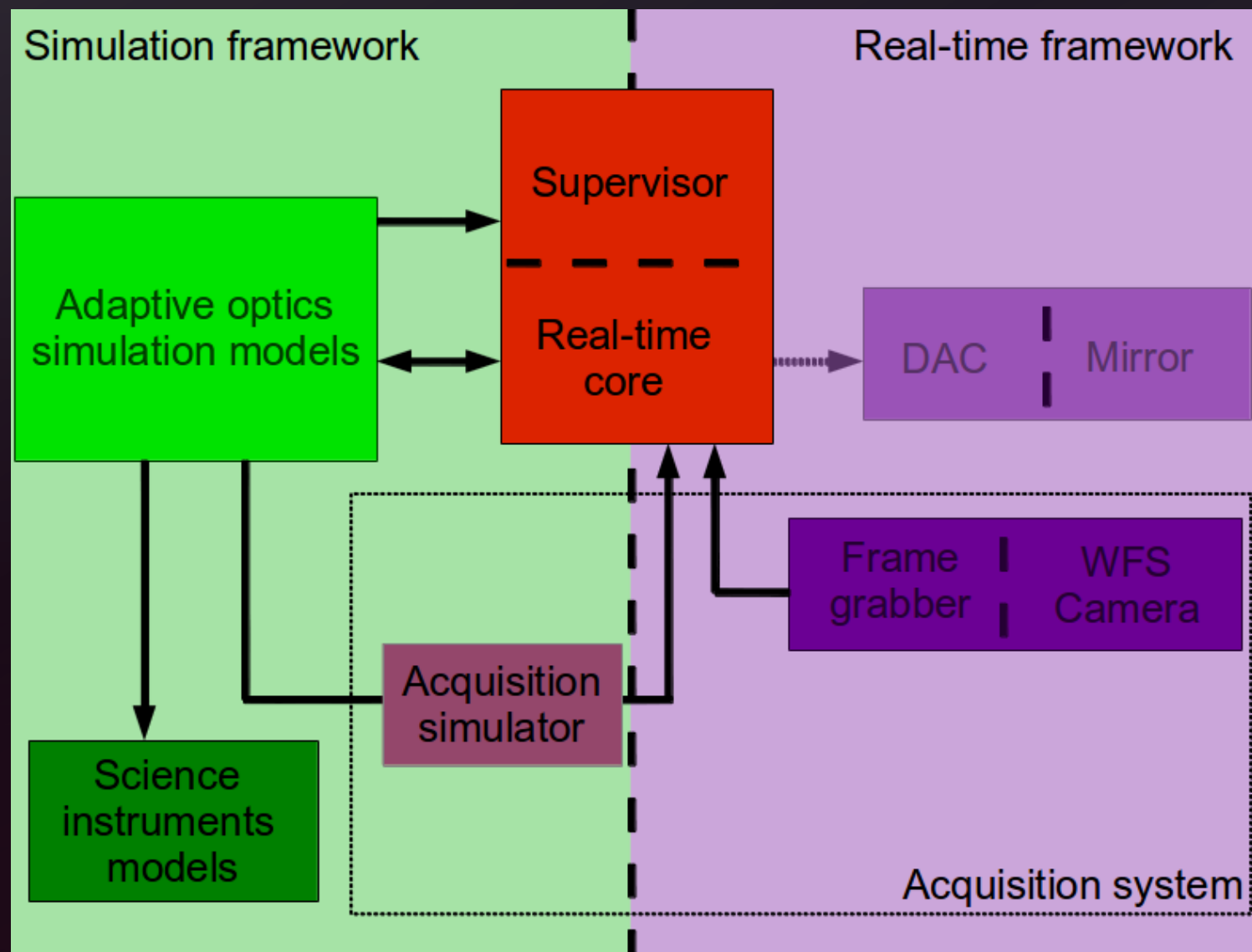


- ◉ Compenser la turbulence
 - ◉ Écrans de phase turbulente
 - ◉ Analyseur de surface d'onde
 - ◉ Shack-Hartmann
 - ◉ Pyramide
 - ◉ Calculateur temps-réel
 - ◉ CdG, Correlation
 - ◉ MC, MV, LQG
 - ◉ Miroir déformable
 - ◉ Cible scientifique



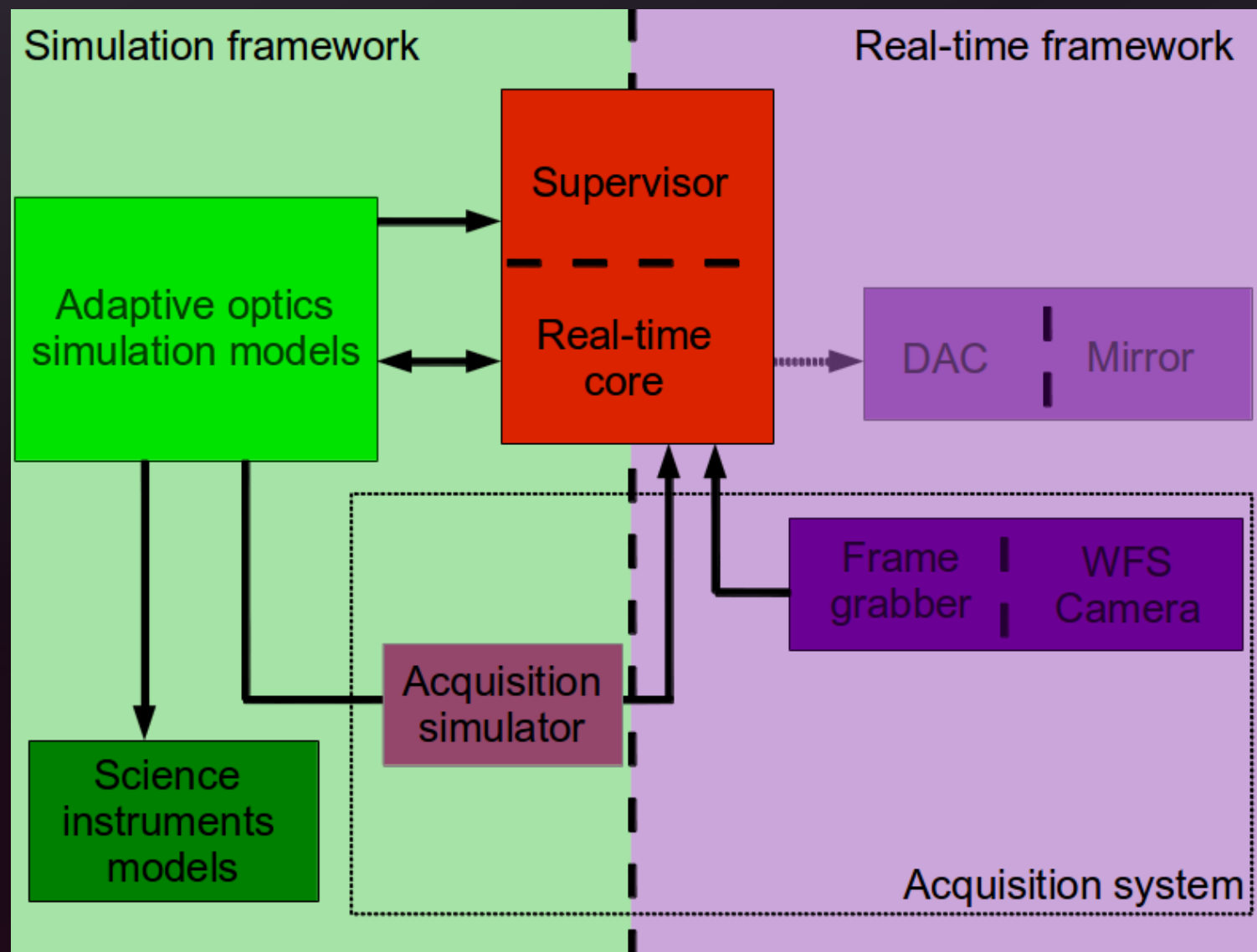
Unifier simulations et temps-réel

- Challenge: accorder les spécifications
 - Temps-réel : faible latence, faible *jitter*
 - Simulations : puissance de calcul maximale

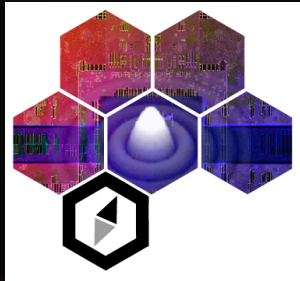


Unifier simulations et temps-réel

- Challenge : accorder les spécifications
 - Réduire le coût de développement tout en augmentant résilience, possibilité de maintenance et de mise à jour

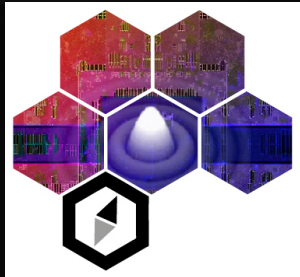


Le projet COMPASS



- Fédérer les efforts en France autour d'une plateforme de développement optimisée
 - Basée sur l'utilisation des GPU comme accélérateurs
 - Partenariat PHASE + partenaire associée : Maison de la simulation (HPC)
 - Collaboration multi-disciplinaire OA + astrophysique + HPC
 - Produit final : outil de simulation haute-performance extensible + prototype temps-réel + environnement unifié avec outils optimisés pour la recherche en OA
- 36 mois, démarrage en mars 2013
 - Répondre aux besoins des études de phase B de l'instrumentation de l'E-ELT
 - 6 partenaires en France : LESIA, GEPI (Paris), IPAG (Grenoble), LAM (Marseille), ONERA, MdlS (Paris)
 - 2 groupes de travail : SWG et CWG pour la supervision des activités

Le projet COMPASS



- Travail collaboratif, projet ouvert, plan de maintenance à long terme

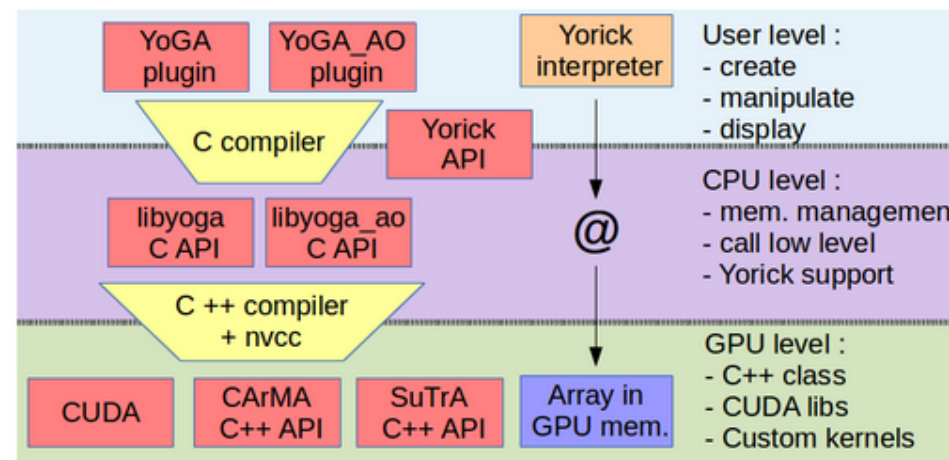
~/home/brujo/Yorick/compass

- branches
- tags
- trunk
 - lam
 - libcarma
 - libsutra
 - svi-pc
 - wrapper_mpi
 - yoga
 - yoga_ao
 - yoga_py
 - yoga_template



Welcome to the COMPASS wiki.

The COMPASS platform relies on several layers of software. This stack is described in the following figure.



The lower level is the GPU domain. This is where resides the computing power and the memory that needs to be addressed to perform the computations. To manipulate GPU arrays and use advanced and optimized numerical algorithms, COMPASS relies on 3 main pillars :

- the CUDA toolkit from NVIDIA including optimized libraries : CUBLAS, CUFFT, CURAND
- the CARMA (C++ Api for Massively parallel Applications)
- SuTrA (Simulation Tool for Adaptive optics).

The intermediate layer is the binding domain. This is where the set of optimized tools are bound to interpreted languages in order to build evolutive and modular higher level applications. For now only the Yorick interpreted language is supported through the Yorick API but plans exist to add Python support.

At the higher level, we find the Yorick plugins accessible to the user through the interpreter. This includes the YoGA plugin, providing general utilities to the user such as random number generation, fast Fourier transform or various BLAS levels. Additionally, the YoGA_AO extension provides a whole environment, including GUIs, to simulate adaptive optics systems.

Follow the links to get more information on each components of the COMPASS project.

[Install the platform](#)

[CARMA](#) : the C++ API for a user-friendly GPU

[SuTrA](#) : the AO simulation tool

[YoGA](#) : the Yorick plugin

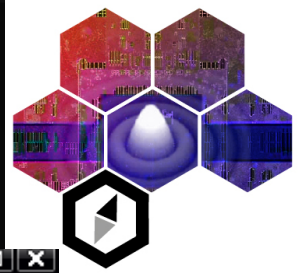
[YoGA_AO](#) : the YoGA extension to manipulates SuTrA objects and run AO simulations

[PRANA](#) : the real-time control development platform

[Websim](#) : the astrophysical observations simulator



La plateforme de simulation



widget_ao.py

File Edit View Help

Phase - DM DM # 0 Enable Display Init unzoom

System : 1 (-9.7276, 298.0000)

phase on DM # 1

Telescope
Diam. (m) 8.0 Zenith (°) 0
Central obs. 0.10
Load default system xSH

Atmosphere WFS DM RTC Target

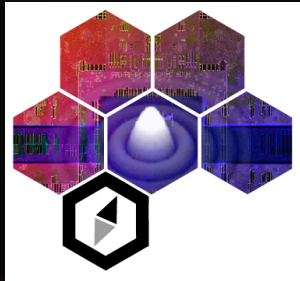
Centroiding
Load default cog
centro # 1
Type cog
Set

Controler
Load default
Type LS
Cond. 1000.0
Delay 1
Gain 0.40
Set update

Device 0

Framerate : 683.60

La plateforme de simulation



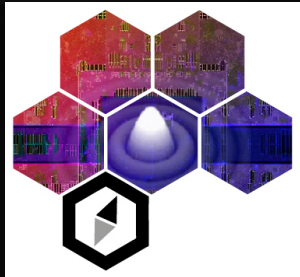
◉ Développements en cours

- ◉ Algorithmes de contrôle optimisés (MV, LQG, ETKF, FrIM) : LESIA, LAM, ONERA
- ◉ Modèles haute précision d'ASO pyramide / roof : IPAG
- ◉ Ajout de fonctionnalités (pupille ELT, modèle de M4, algèbre creux, etc ..) : Tous avec contribution notable IPAG et LAM
- ◉ Optimisation des performances (auto-tuning vers de nouvelles architectures, MPI pour environnements distribués) : MdlS, LESIA
- ◉ Nouveaux algorithmes de (CuReD et p-CuReD, etc ..) : LESIA + AAO

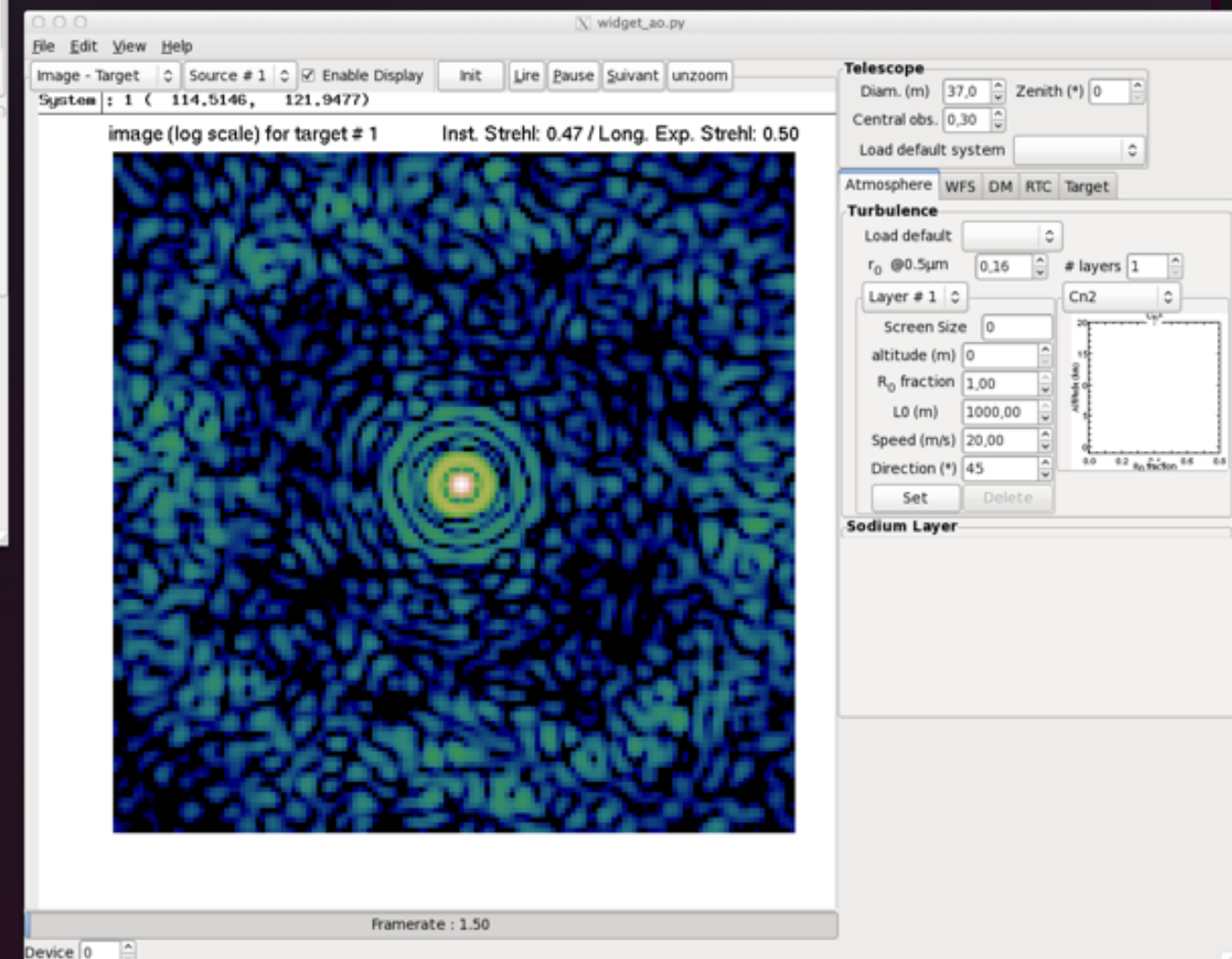
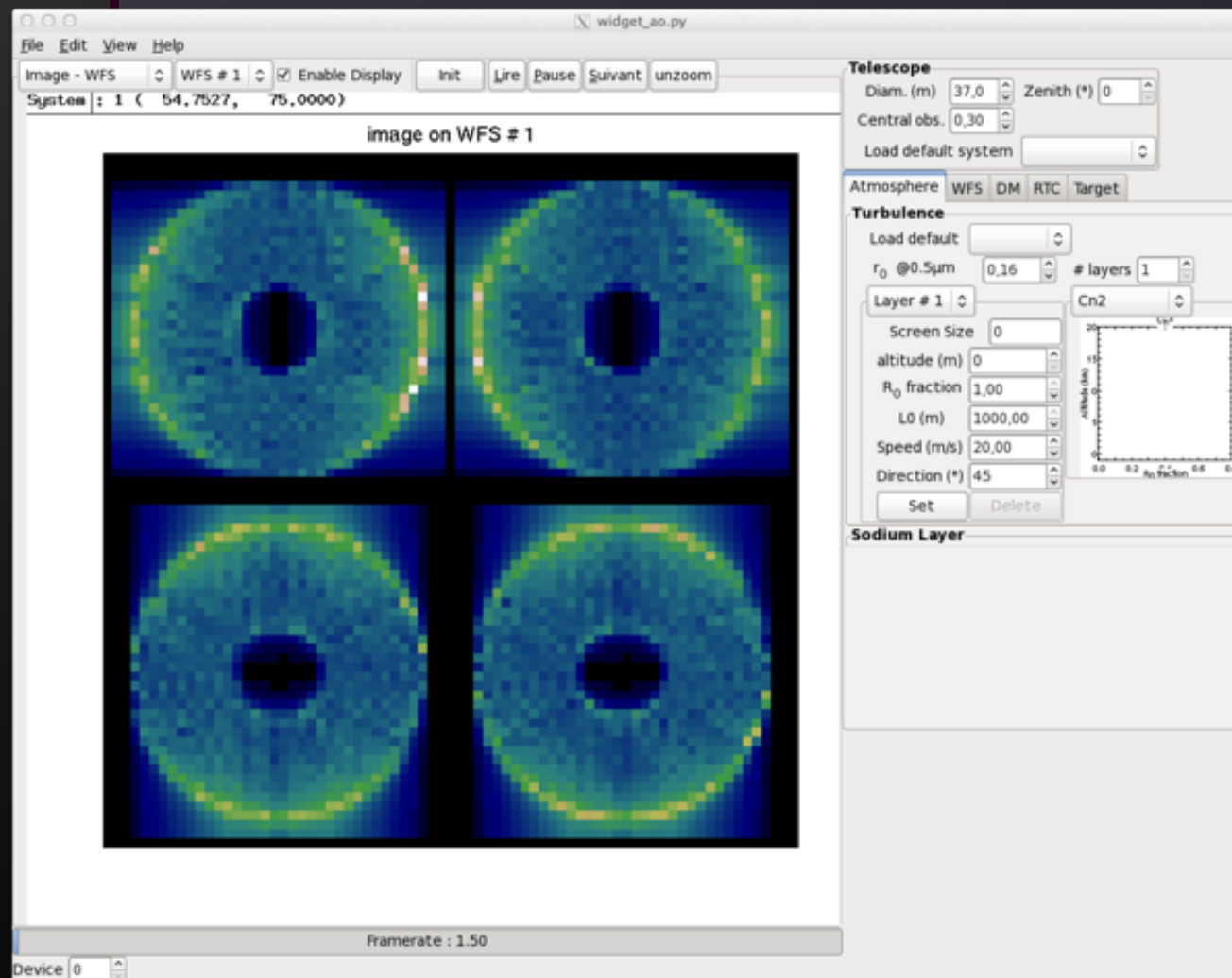
◉ Principaux challenges

- ◉ Maintenance / support à long terme
- ◉ Interface Yorick
- ◉ Suivre l'évolution de l'architecture pour maintenir / augmenter les performances

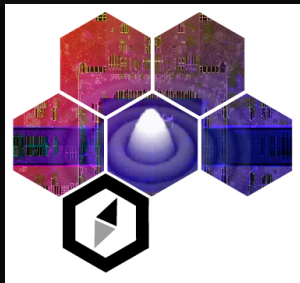
La plateforme de simulation



- ◉ Simulation originale : modèle de *roof sensor*



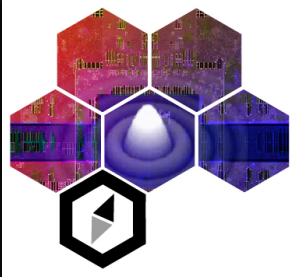
La plateforme de simulation



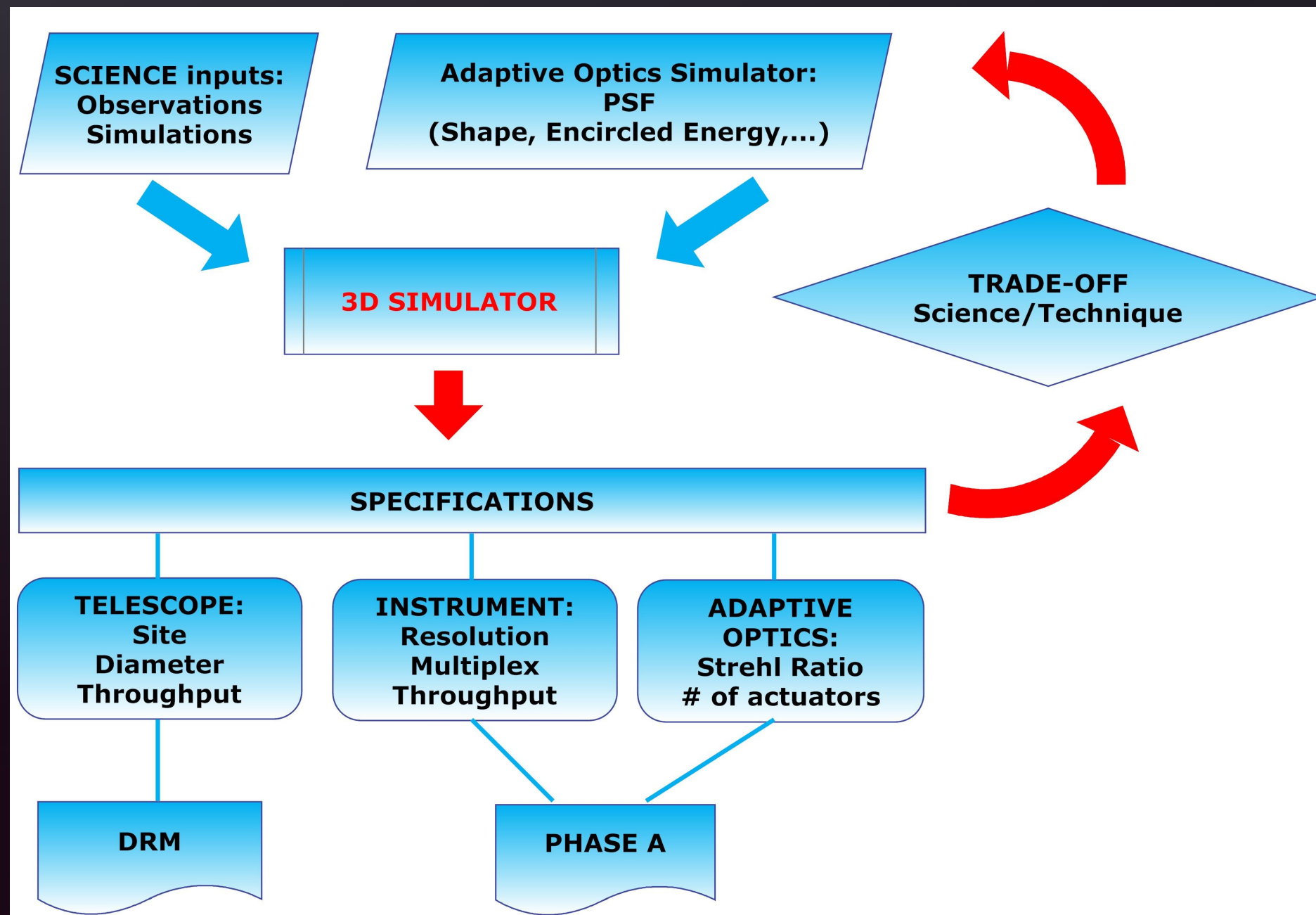
- Profils de performance sur une carte Tesla M2090 (en ms)
 - Système type SCAO pour différentes tailles de télescope
 - Profils dominés par les tâches de simulations
 - Performances du contrôle temps-réel compatible avec spécifications du système

Telescope diam.	Turbu generation	Raytracing turbu	Raytracing DM	WFS	COG	Control	DM shape computation	Raytracing target
4m	0.107	0.008	0.008	0.138	0.013	0.019	0.137	0.008
8m	0.192	0.022	0.023	0.459	0.031	0.060	0.562	0.023
20m	0.550	0.135	0.136	3.07	0.079	0.363	3.22	0.137
30m	0.927	0.299	0.300	6.73	0.168	0.915	7.39	0.302
40m	1.44	0.526	0.525	11.9	0.320	2.263	13.62	0.527

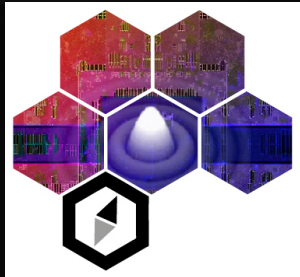
L'instrumentation de l'E-ELT



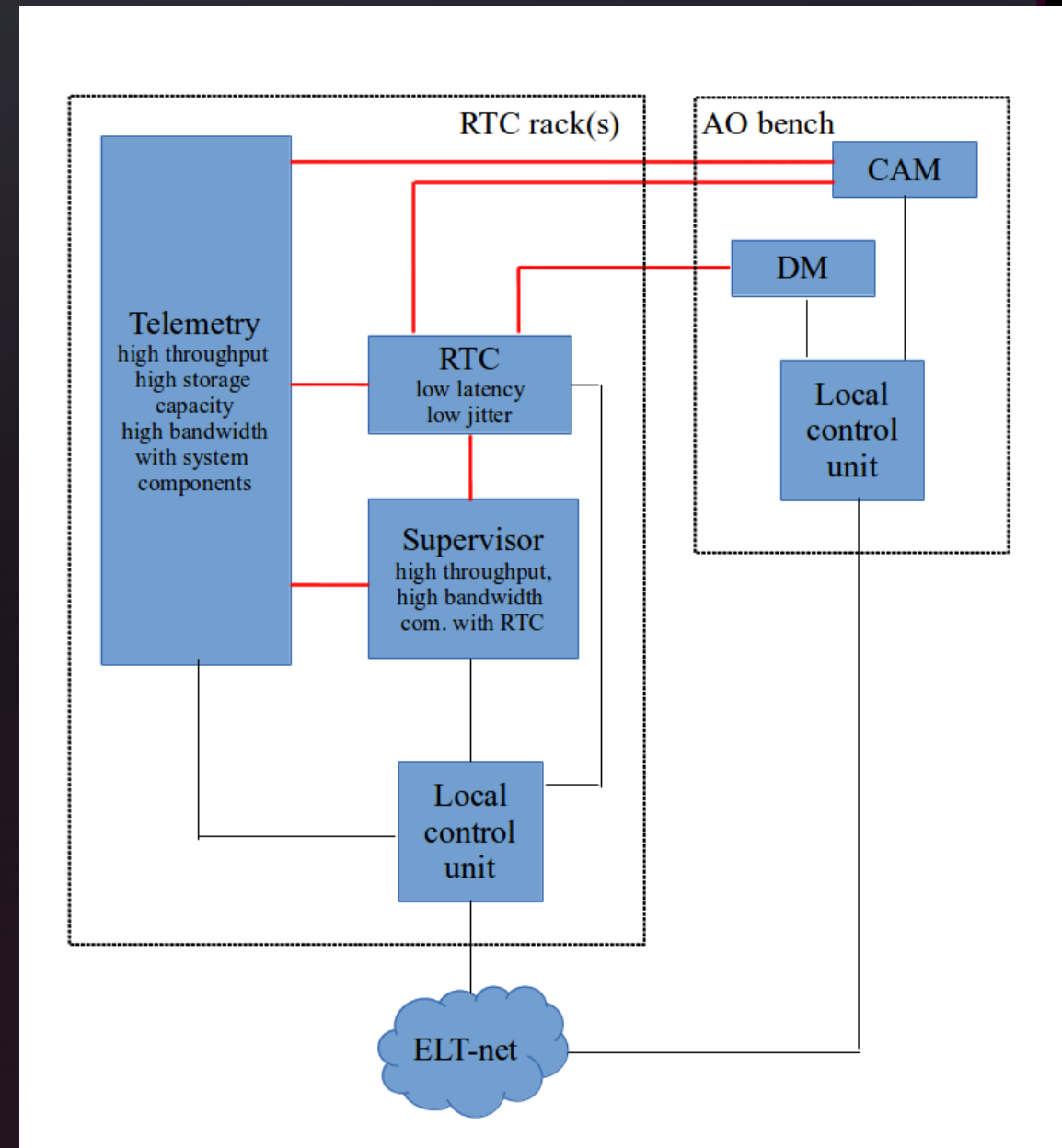
- Étude de compromis pour le design final des instruments



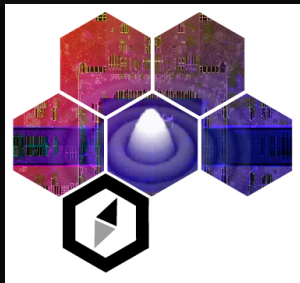
Challenges pour les RTC de l'ELT



- Highly heterogeneous HPC facility
 - 3 main sub-systems: HRT, SRT, Telemetry
 - Different constraints (throughput / latency / bandwidth / storage)
 - Interconnects between RTC sub-systems but also with AO bench sub-systems (WFS, DM)
- Main challenges
 - Cope with highly heterogeneous hardware topology (process synchronization, low latency, high bandwidth, interconnects)
 - Scalability: different instruments / requirements but same standard platform
 - Maintenance, long-term support, upgrades
 - Cost (development and operations), power efficiency



Challenges for ELT AO RTC

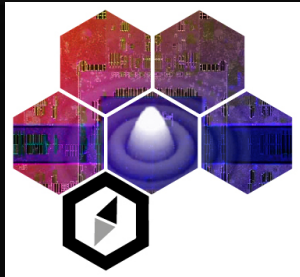


- Choosing the proper technology for each sub-system
 - Concurrent execution on heterogeneous platform: new paradigm in HPC industry
 - 3 mainstream architectures possible: X86 CPU, FPGA , GPU / Intel MIC

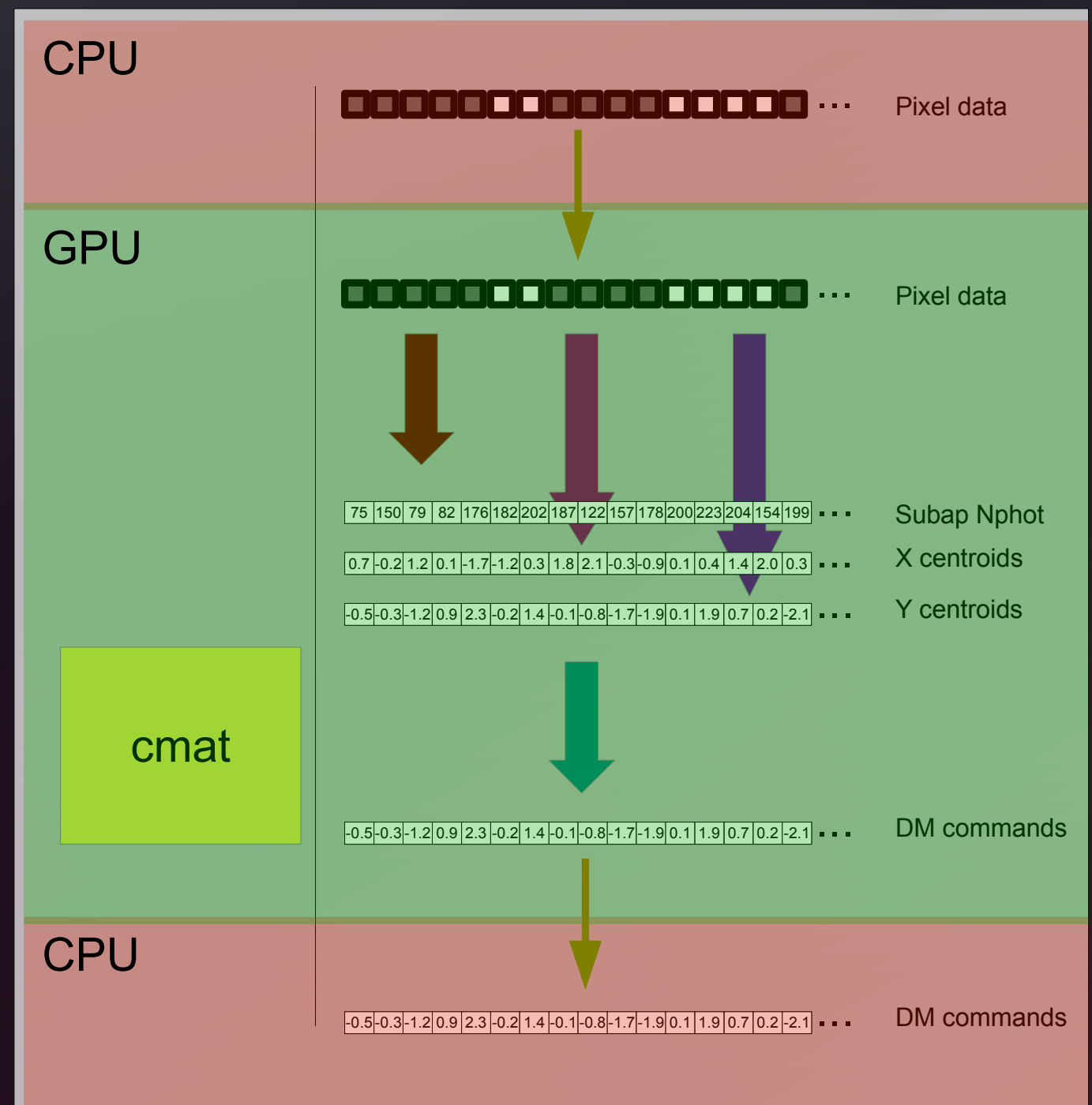
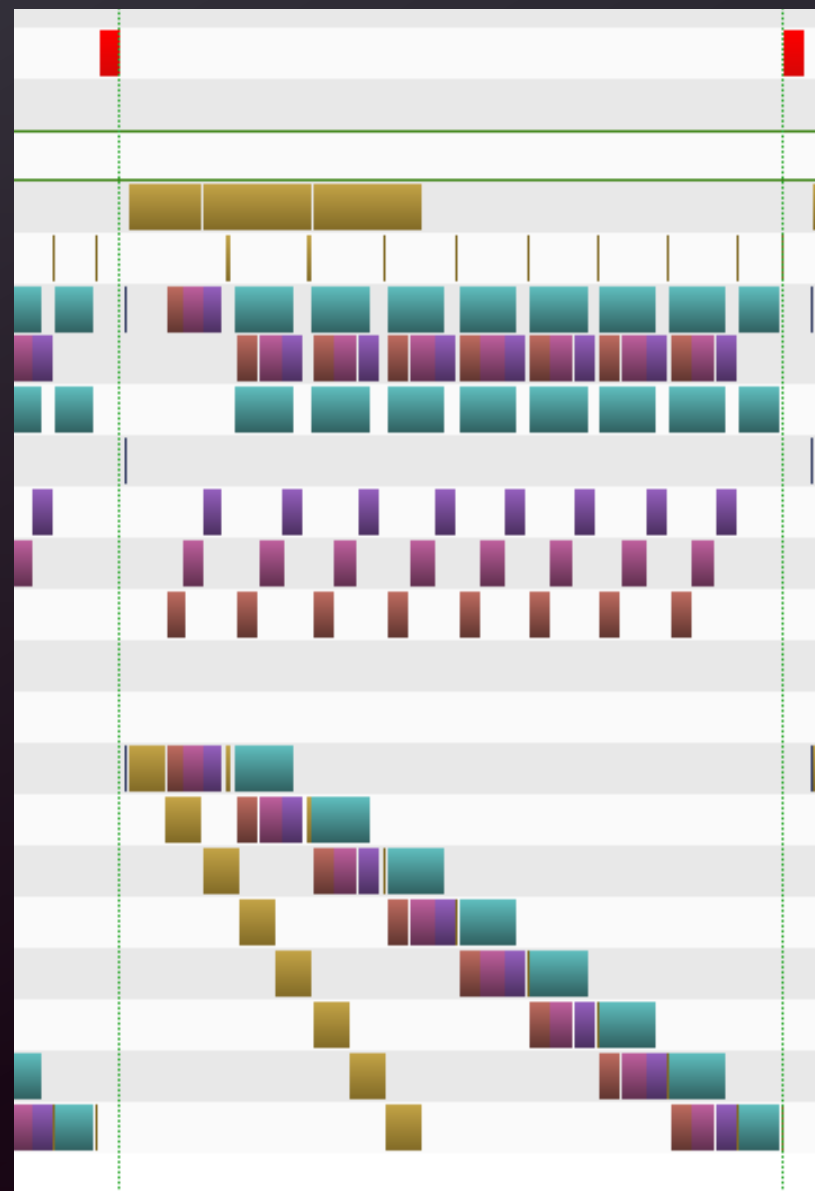
CPU	GPU	FPGA
Flexible programming model	Flexible programming model (CUDA)	Complex programming model
Standalone, host of the OS	Discrete device, closed source driver	Discrete device, closed source driver
High latency (tweakable, not straight forward for multi-core configuration)	High latency	Very low latency
Low performance per Watt	High performance per Watt	Highest performance per Watt
Cheap	Cheap	Expensive (procurement + development + IPs)

- Trade-off not trivial, GPUs represent an interesting alternative: compatible with SRT / Telemetry specs.
- What about HRT ?

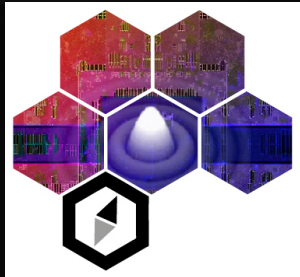
La plateforme temps-réel



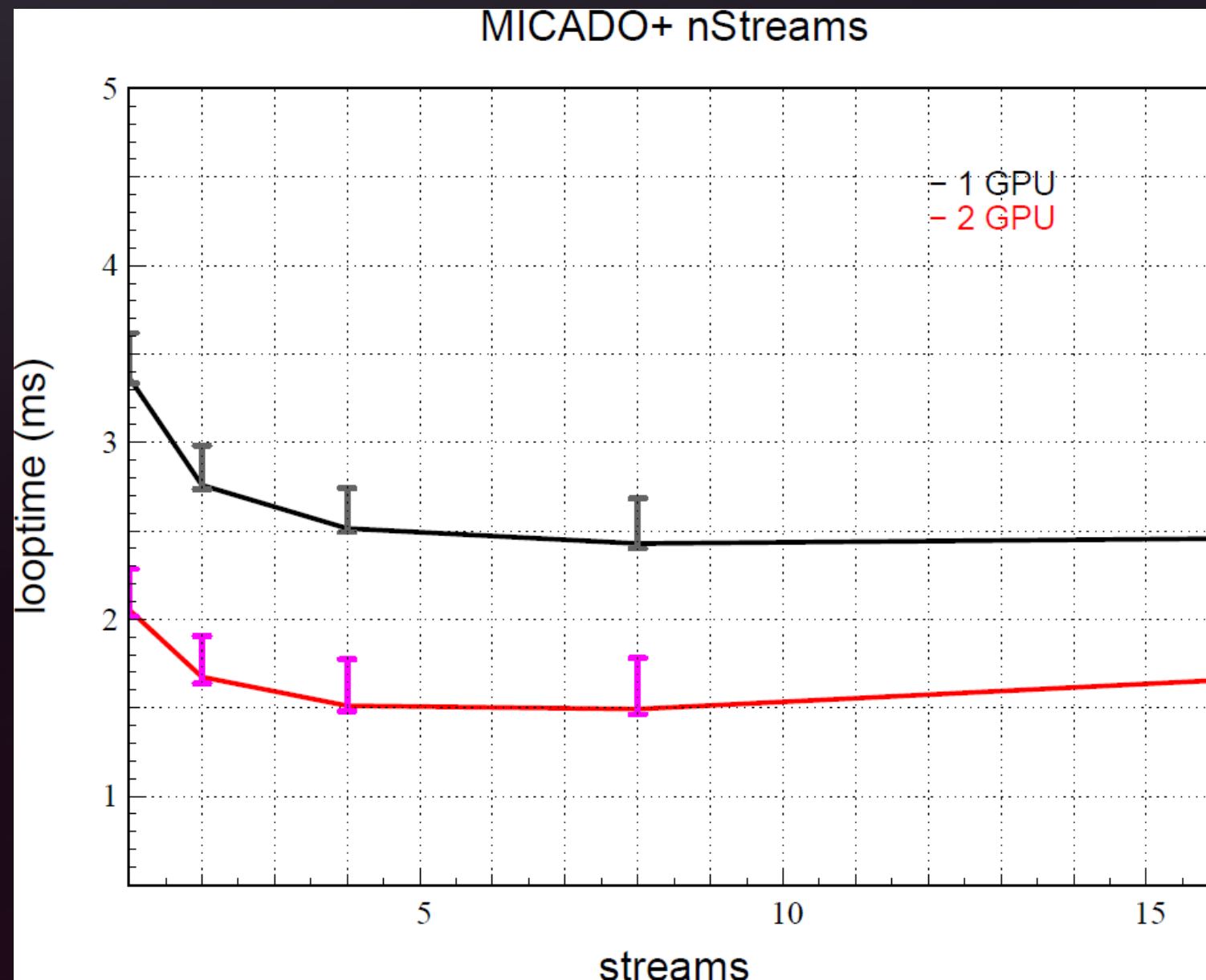
- ◉ Simulateur de contrôleur temps-réel



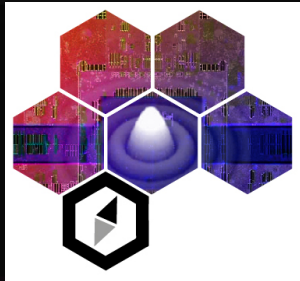
La plateforme temps-réel



- Simulateur de contrôleur temps-réel : performance sur Tesla M2090
 - Compatible avec les spécifications MICADO (> 500 FPS)

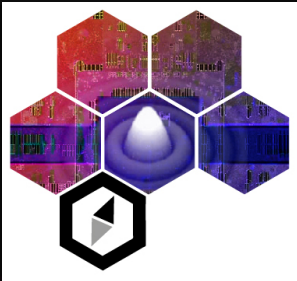


La plateforme temps-réel

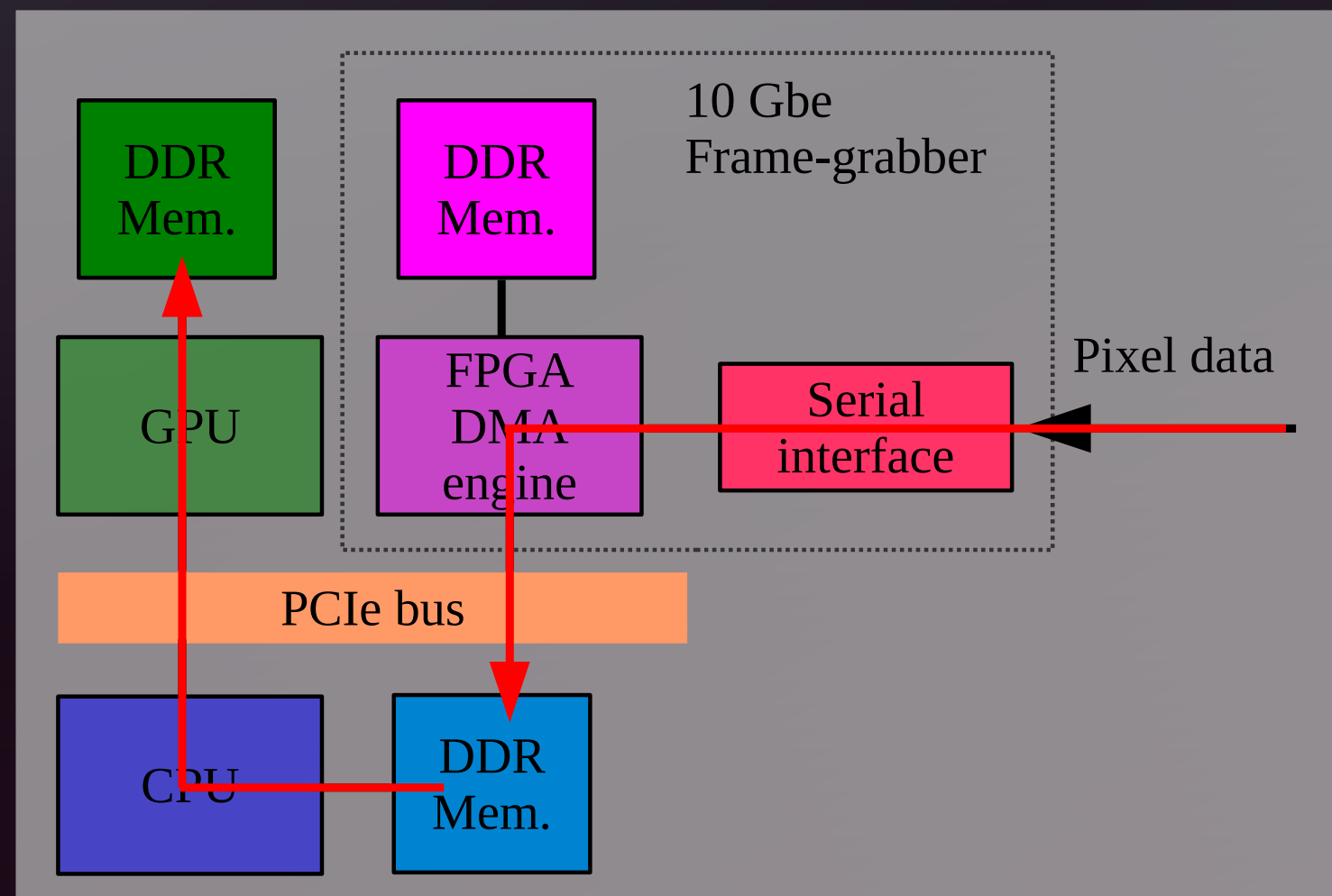


- Challenge majeur : utiliser les GPUs pour des applications temps-réel (latence < 1ms)
 - GPU drivers fermés + pas de notion de priorité des tâches au niveau de l'OS
 - Optimisés pour la puissance de calcul pas pour la minimisation de la latence
 - => minimiser les interactions entre le(s) GPU(s) et l'OS
 - Transférer les données directement vers la mémoire du GPU
- Liens : une interface à faible latence pour l'accès direct à la mémoire du GPU
 - API GPUdirect de NVIDIA pour l'adressage de la mémoire
 - Sujet « très chaud » dans la communauté GPU (nombreuses applications au-delà du HPC)
 - Designer un *interconnect*, compatible avec les spécifications de l'E-ELT pour le transfert faible latence

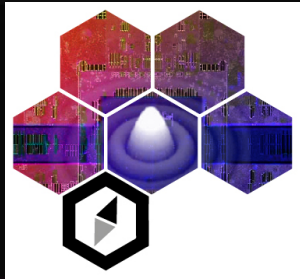
La plateforme temps-réel



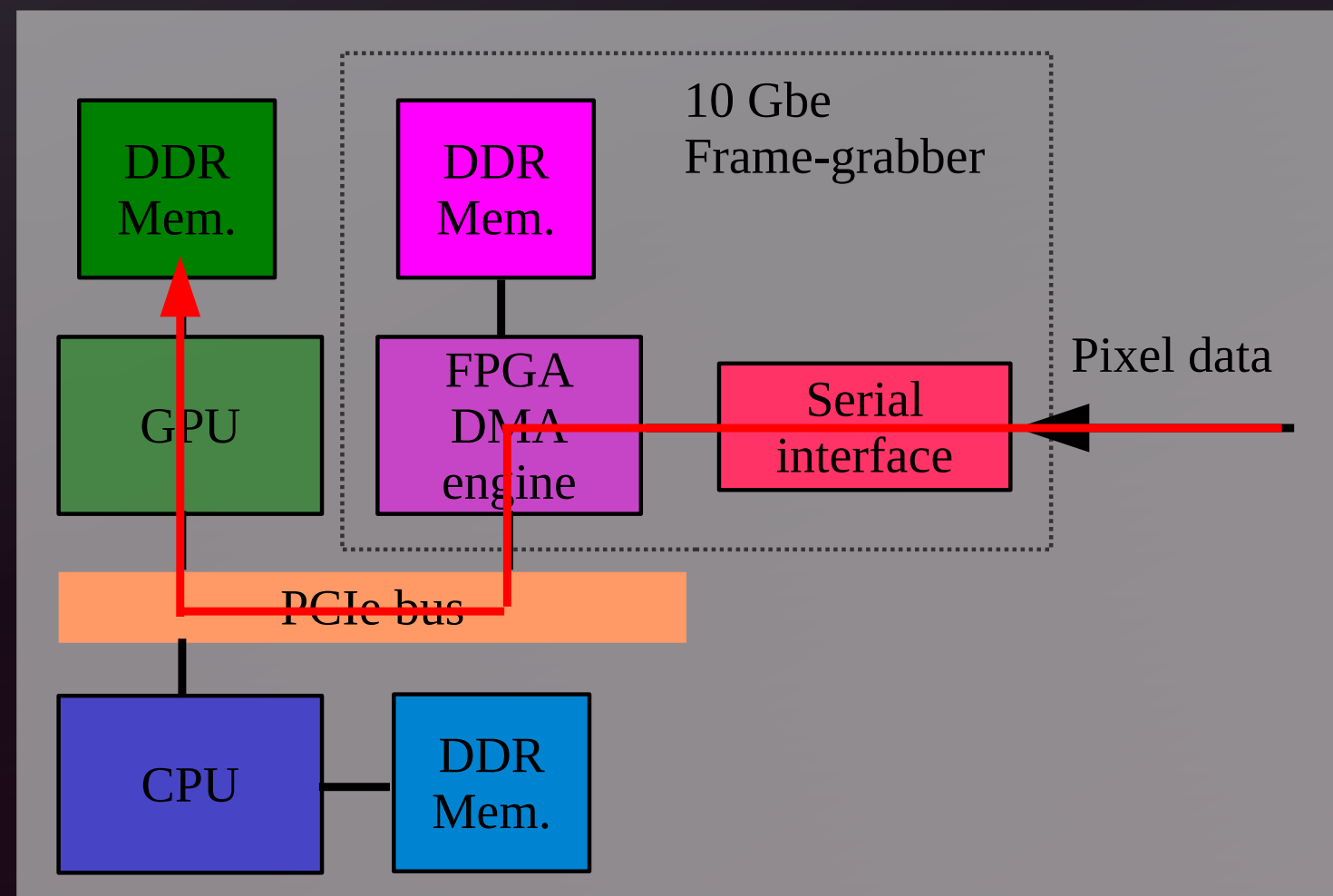
- Déverrouillage technologique pour un contrôleur à base de GPU
 - Interface d'acquisition 10 GbE à faible latence
 - sans GPU RDMA : copie multiple des données = latence



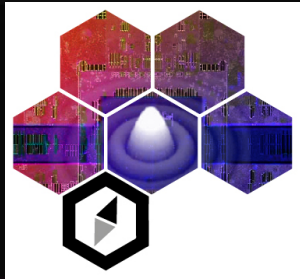
La plateforme temps-réel



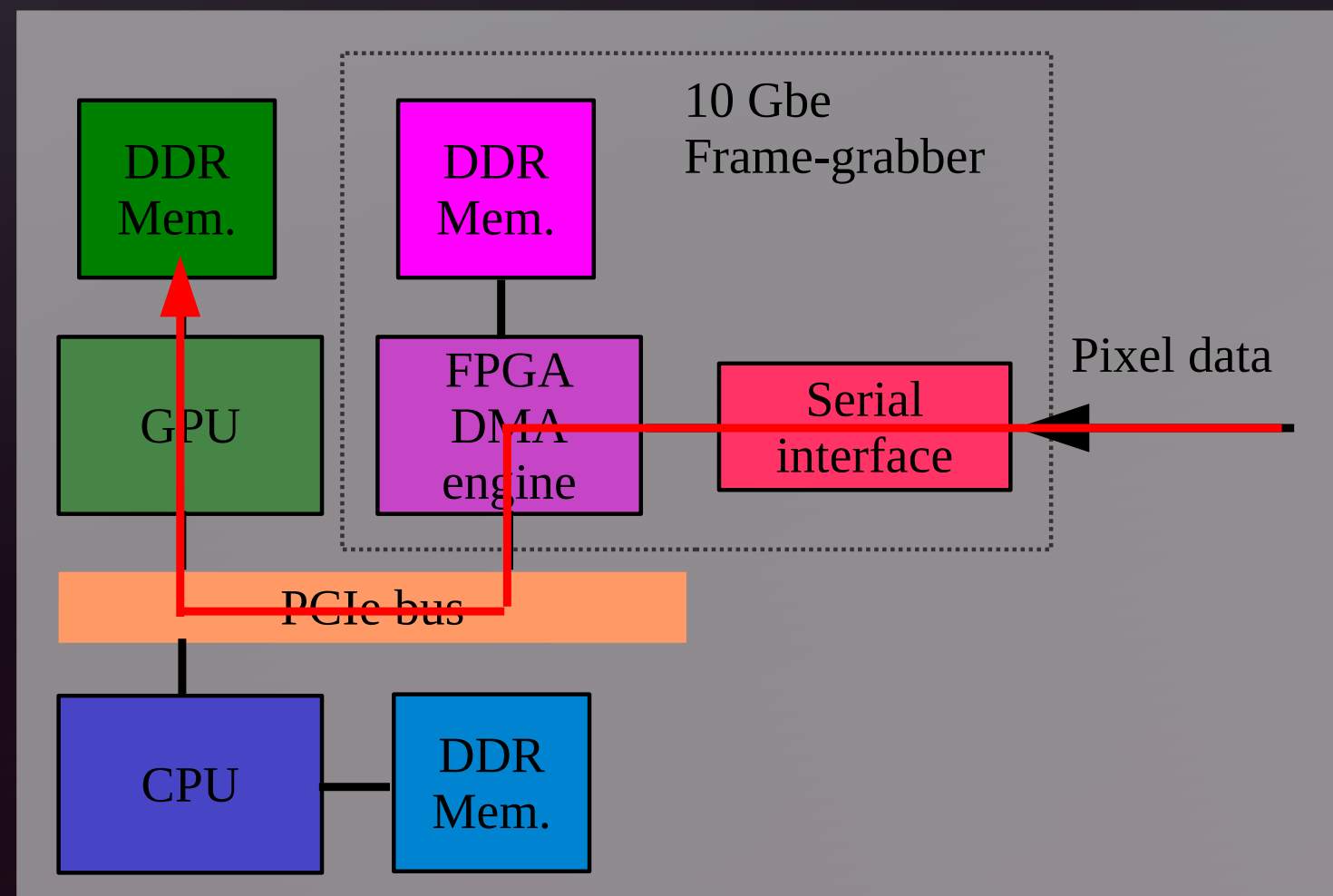
- Déverrouillage technologique pour un contrôleur à base de GPU
 - Interface d'acquisition 10 GbE à faible latence
 - avec GPU RDMA : une seule copie = faible latence, bande passante maximale



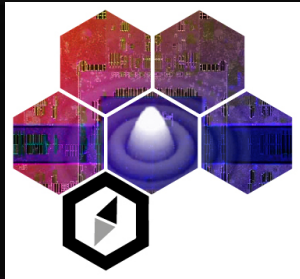
La plateforme temps-réel



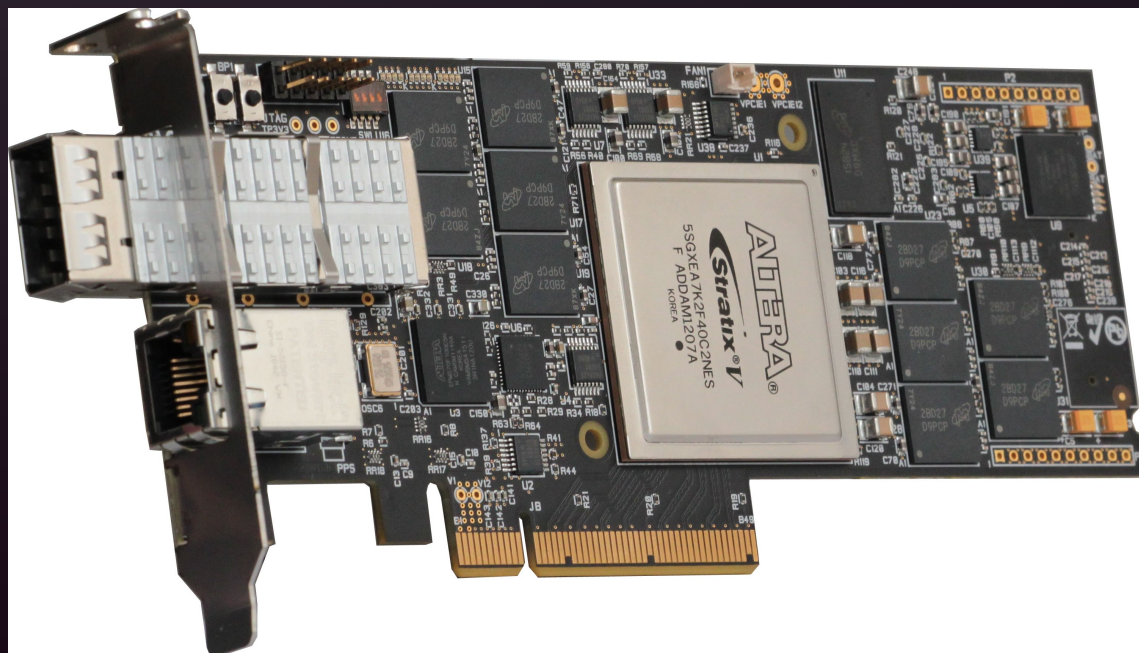
- Déverrouillage technologique pour un contrôleur à base de GPU
 - Interface d'acquisition 10 GbE à faible latence
 - avec GPU RDMA : une seule copie = faible latence, bande passante maximale



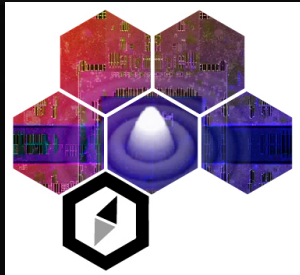
La plateforme temps-réel



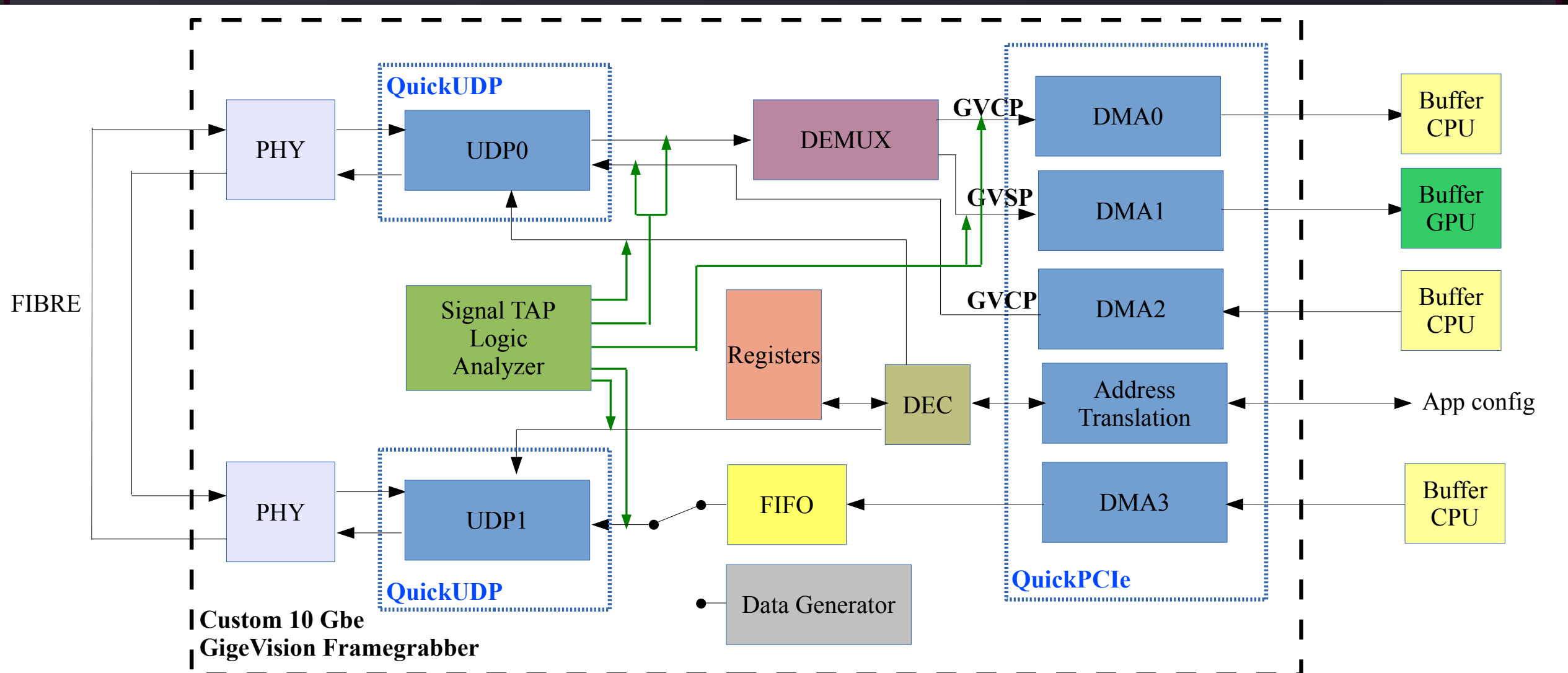
- PRANA: Prototype Real-time Architecture for Next generation Ao
 - Carte de développement PCIe Stratix V de PLDA (+ QuickPCIe, QuickUDP IP cores) 42 Gb/s démontrés entre la carte et un GPU ; 8.8 Gb/s par lien 10GbE en *loopback*
 - Caméra 10 GbE de Emergent Vision Technologies (8.9 Gb/s vers mémoire CPU) 1.5 kFPS en mode 256x256 (330 FPS en 2k x 1k)



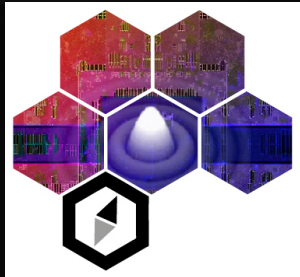
La plateforme temps-réel



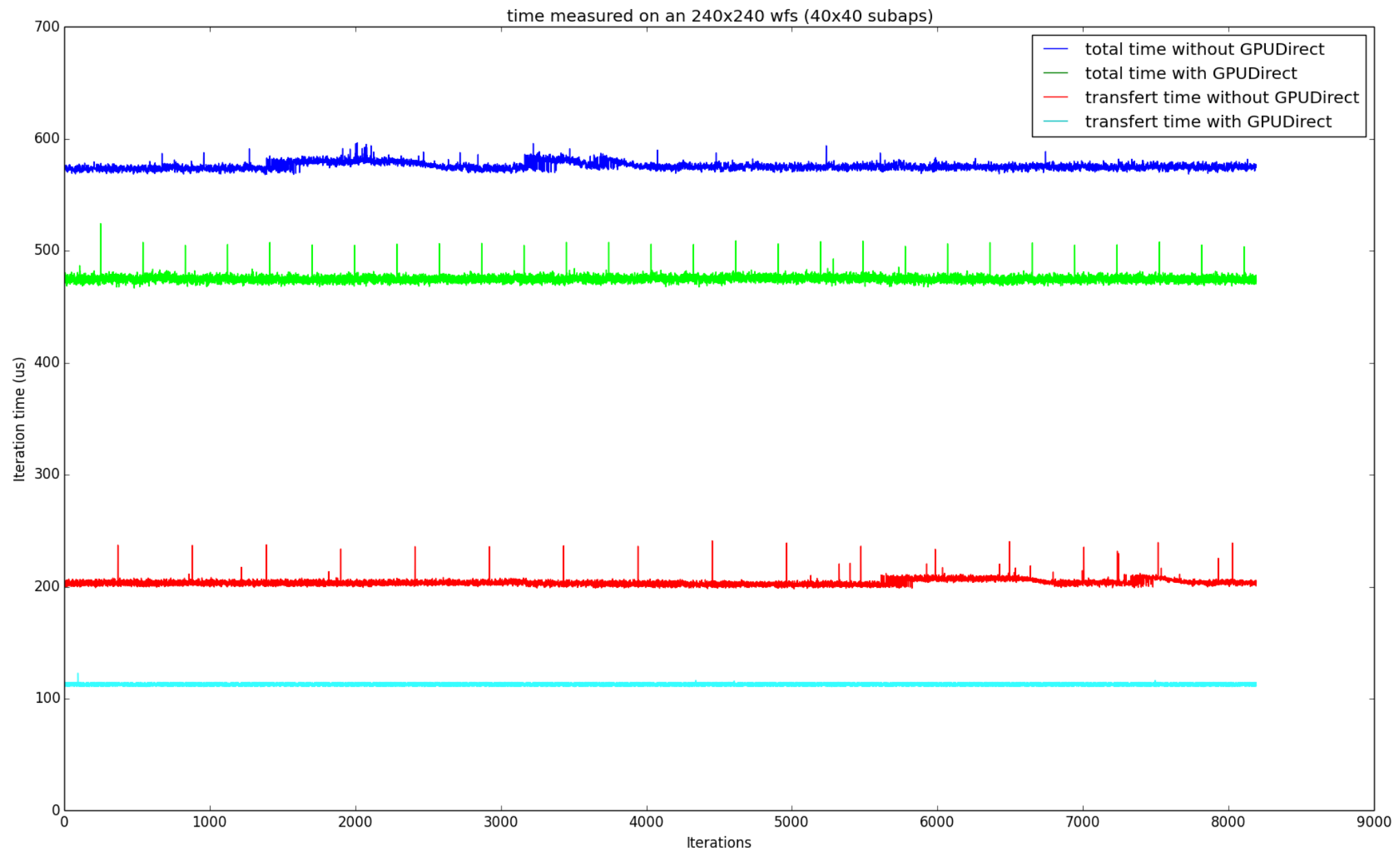
Design de l'interface



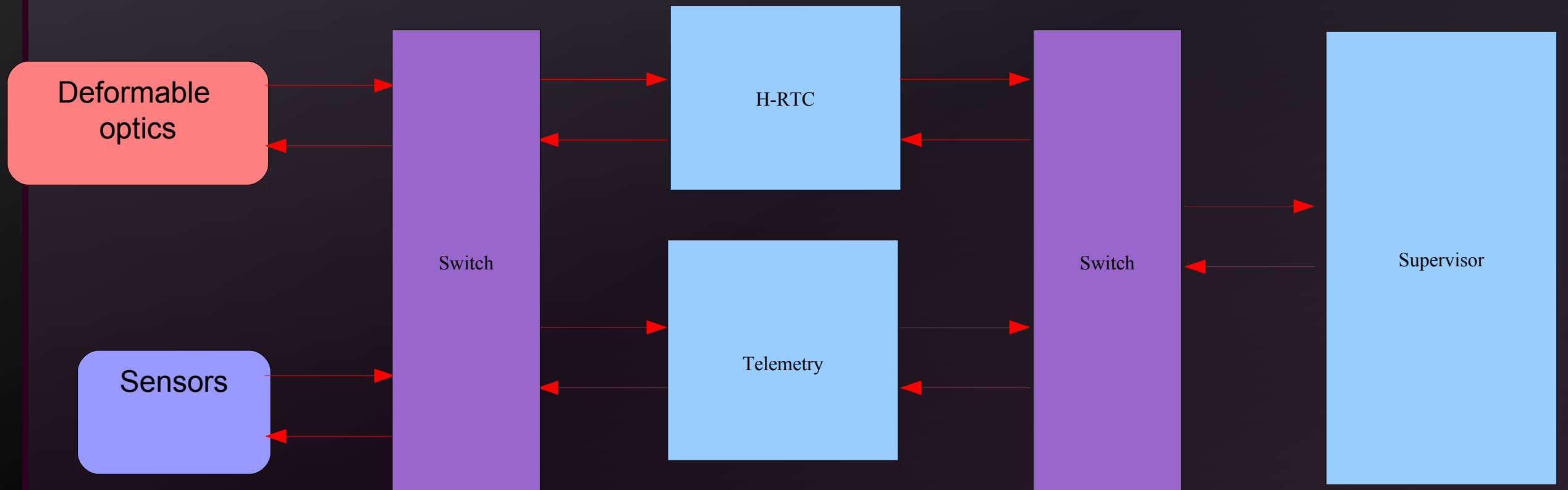
La plateforme temps-réel



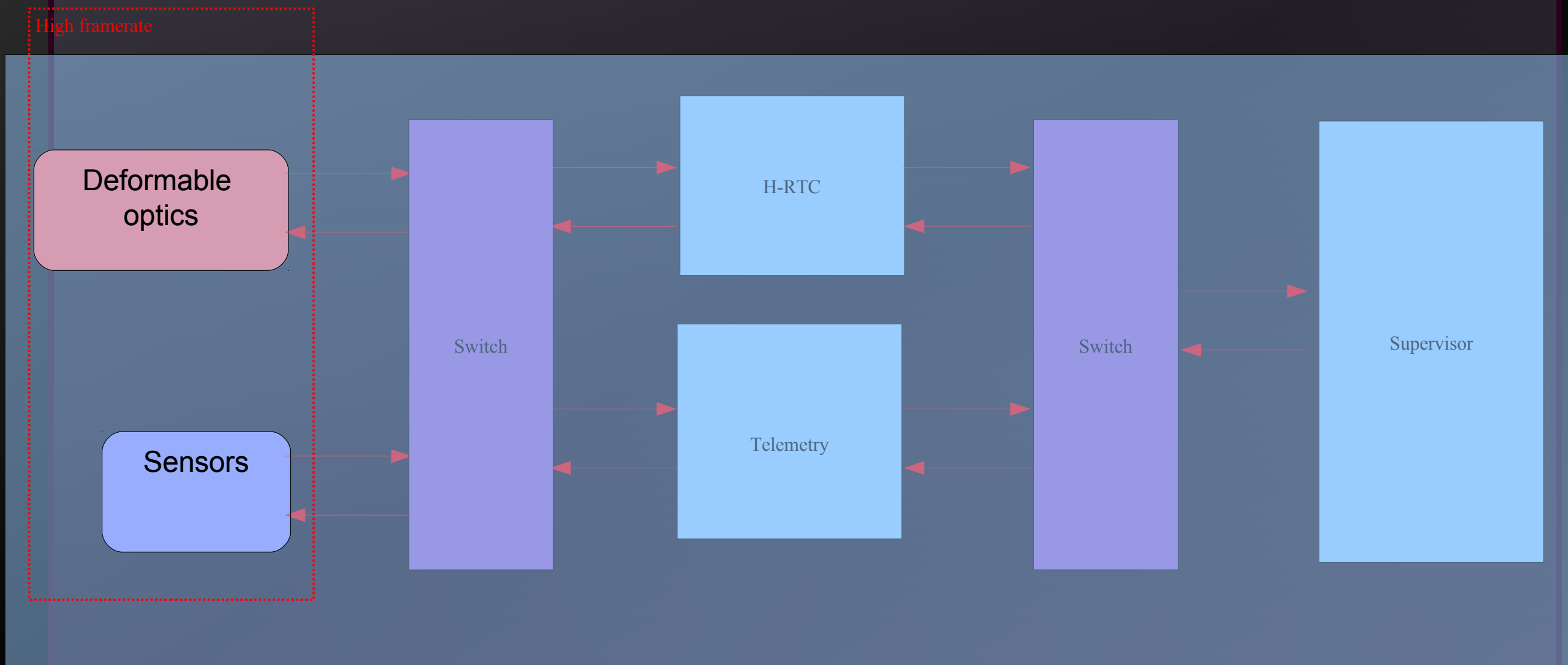
- ◉ Premiers résultats : OA extrême sur un télescope de 8m



Challenges pour les RTC de l'E-ELT



Challenges pour les RTC de l'E-ELT



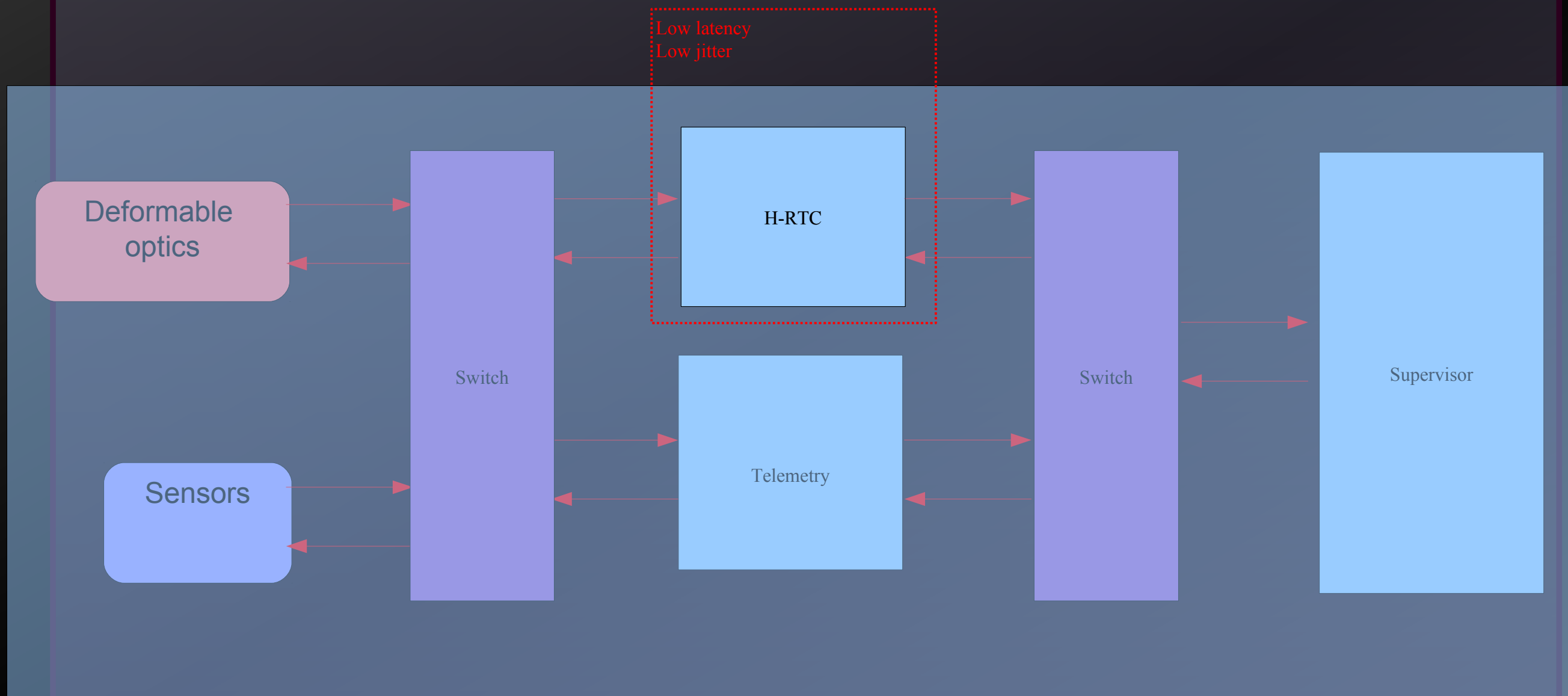
Main driver of the system: data and variable data rate

Sensors produce data to be processed in real-time

Real-time controller sends commands to optics

Data are archived for supervising / post-processing

Challenges pour les RTC de l'E-ELT

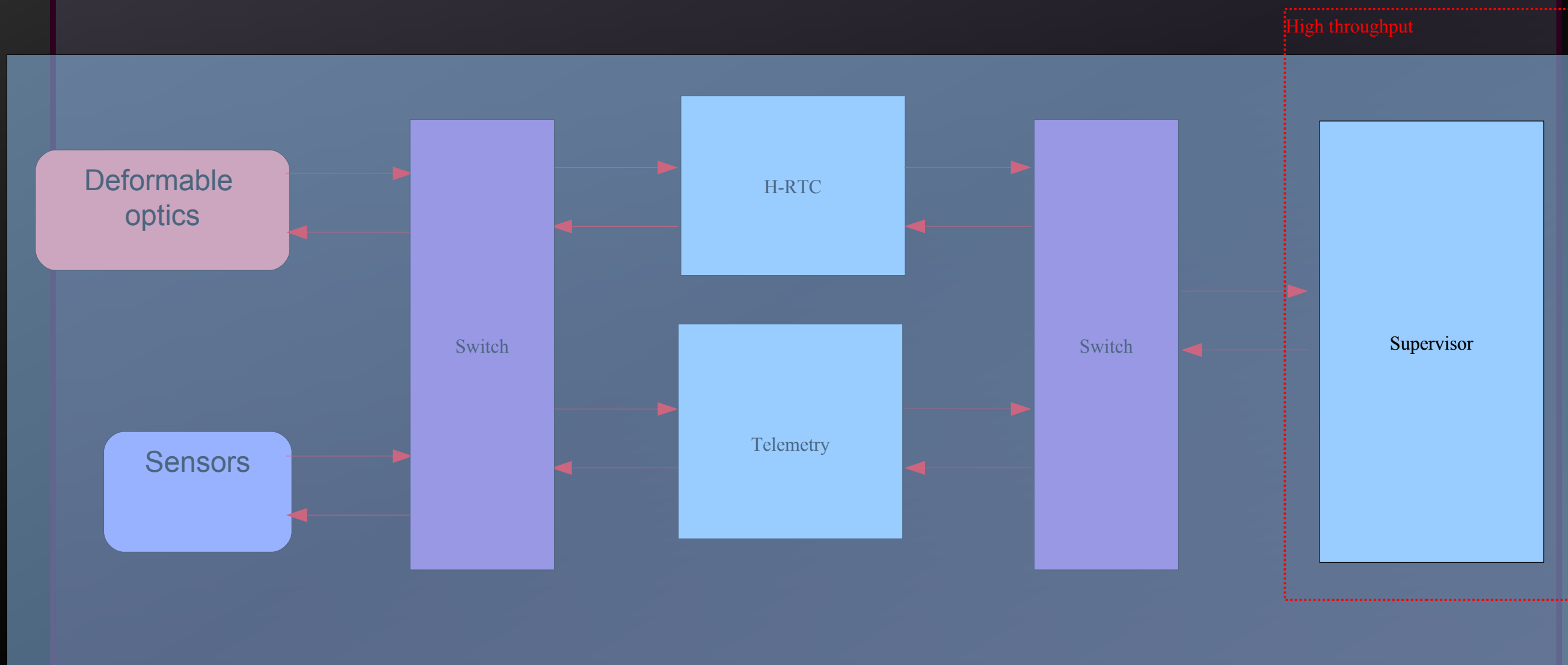


Main challenge for the system: the real-time core

Achieve low-latency and low-jitter @~250 Gb/s data flow and > 1 TFLOPS computational load

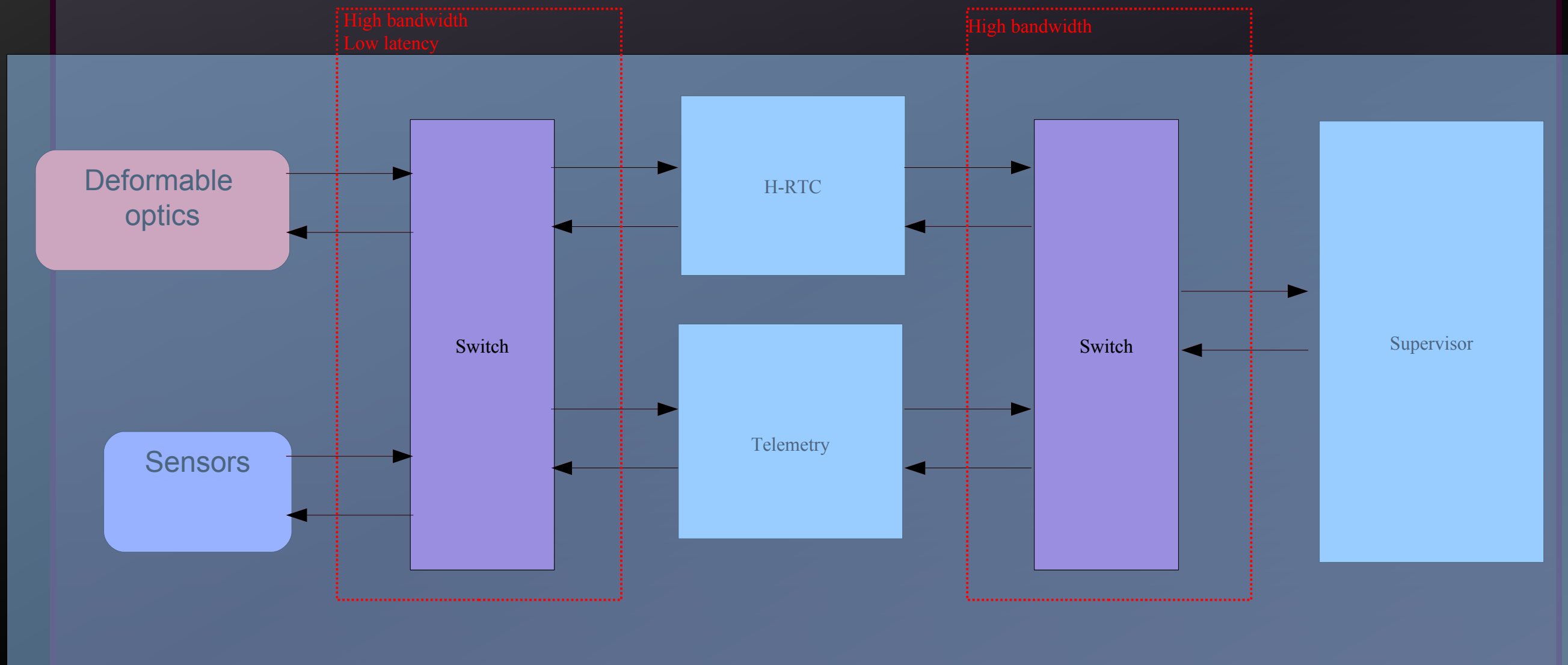
Resilient: central part of the system / durable maintainability

Challenges pour les RTC de l'E-ELT



Main challenge for the performance of the system: high throughput
Reduce time-to-solution in large dense linear algebra problems
Energy efficiency / scalability / maintainability / upgradability

Challenges pour les RTC de l'E-ELT



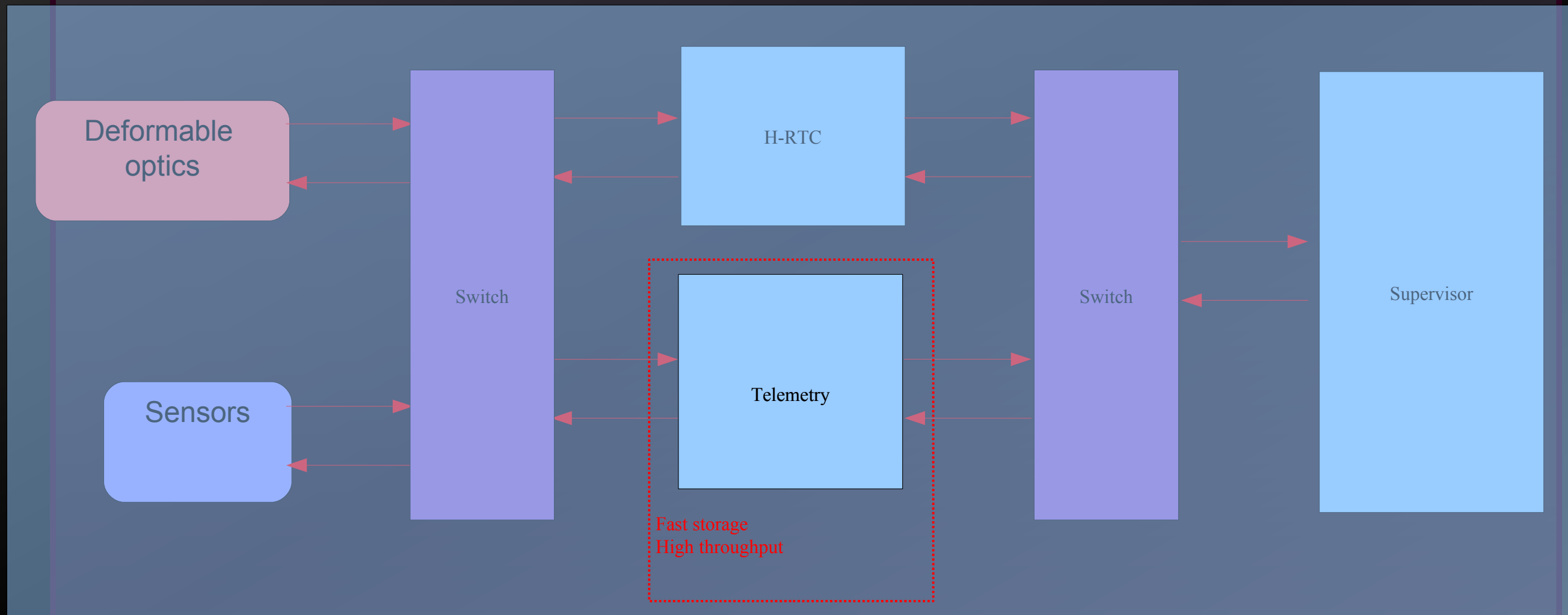
Backbone of the system: interconnects

High bandwidth: fiber / standard protocols

Low latency: optimized memory addressing, data flow reduction

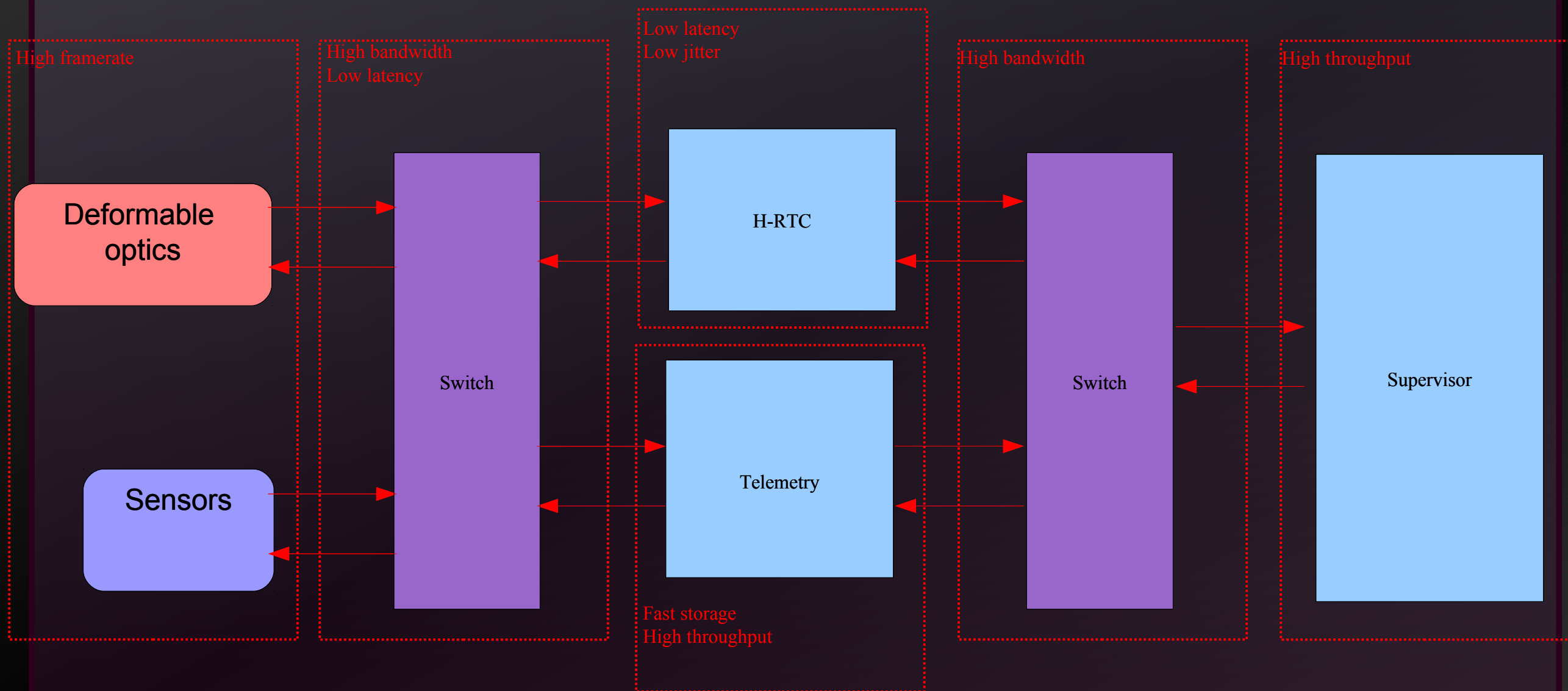
Smart interconnects

Challenges pour les RTC de l'E-ELT



- End-user interface to the data produced by the system
- Efficient storage and data reduction pipeline
- High throughput for real-time system diagnostic
- Energy efficiency, storage resiliency, multi-level storage

Challenges pour les RTC de l'E-ELT



Strong cost and environmental constraints

Energy efficiency: minimize load on overall ELT requirements and operational cost

Maintainability: must be operational in ~10 years and for 20 years

Based on standards: minimize development and maintenance costs

Le projet Green FLASH



- Prototype and test under realistic condition a heterogeneous, energy efficient, HPC platform as an instrument for an extreme scale science facility
- Assess the potential of several emerging technologies for strategic sub-systems including interconnects, real-time operations and minimized time-to-solution for large scale linear algebra problems
- 4 axes of development:
 - FPGA SoC for hard-RT / smart interconnect
 - Scalability of the GPU solution for distributed hard-RT tasks
 - Optimized interconnect management over the whole system
 - Energy efficient supervisor module (ARM + accelerator)

Le projet Green FLASH



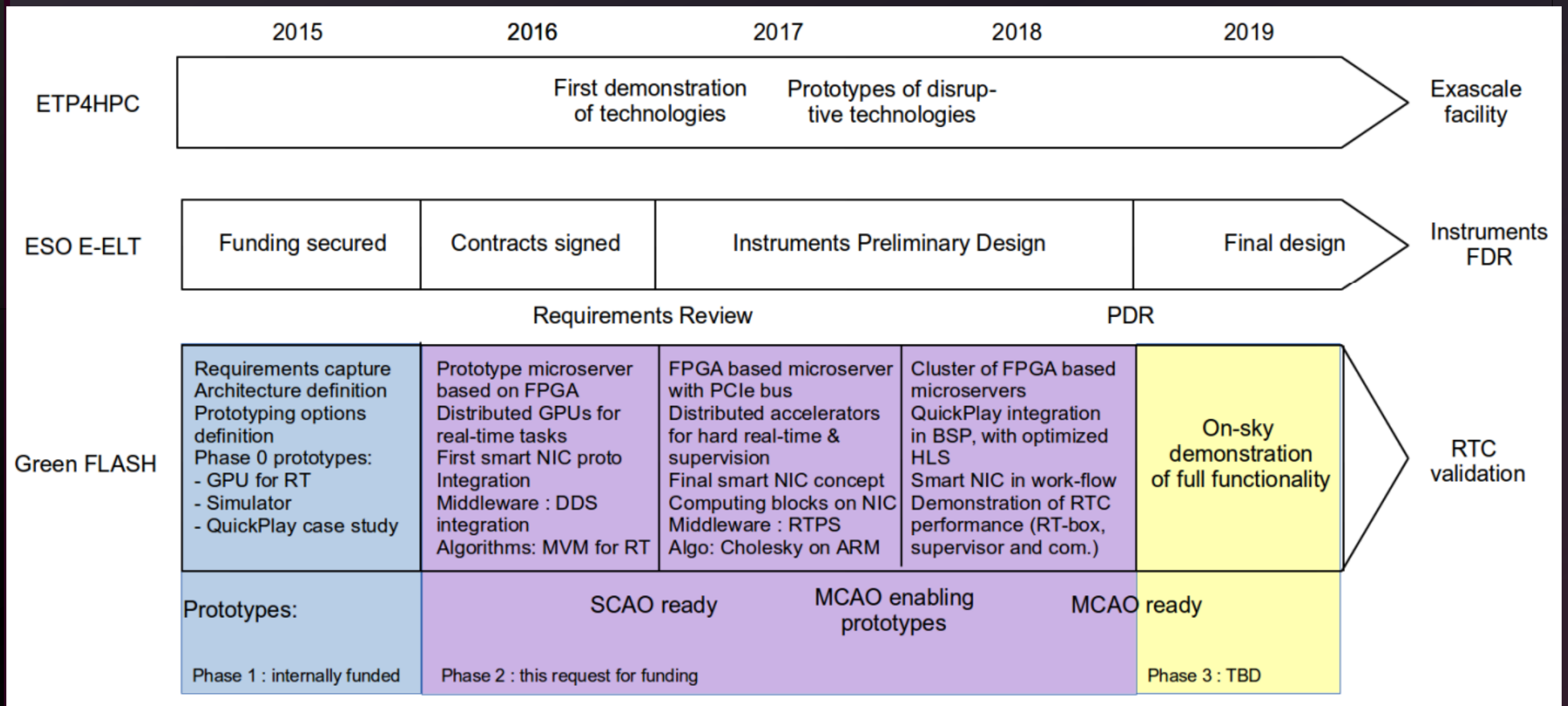
◉ Consortium

- ◉ LESIA, Observatoire de Paris
 - ◉ Scientific management, system architecture, GPU, FPGA NIC / frame-grabber, OpenCL on FPGA, Algorithms, middleware (DDS, MPI)
- ◉ University of Durham
 - ◉ System architecture, GPU, Intel MIC, OpenCL on GPU / MIC, Middleware (DDS, CORBA)
- ◉ Microgate (www.microgate.it)
 - ◉ FPGA IP cores / boards: NIC / accelerator / SoC, application to AO systems (mirror control, real-time computer),
- ◉ PLDA (www.plda.com)
 - ◉ FPGA IP cores / boards, FPGA development environment

Le projet Green FLASH

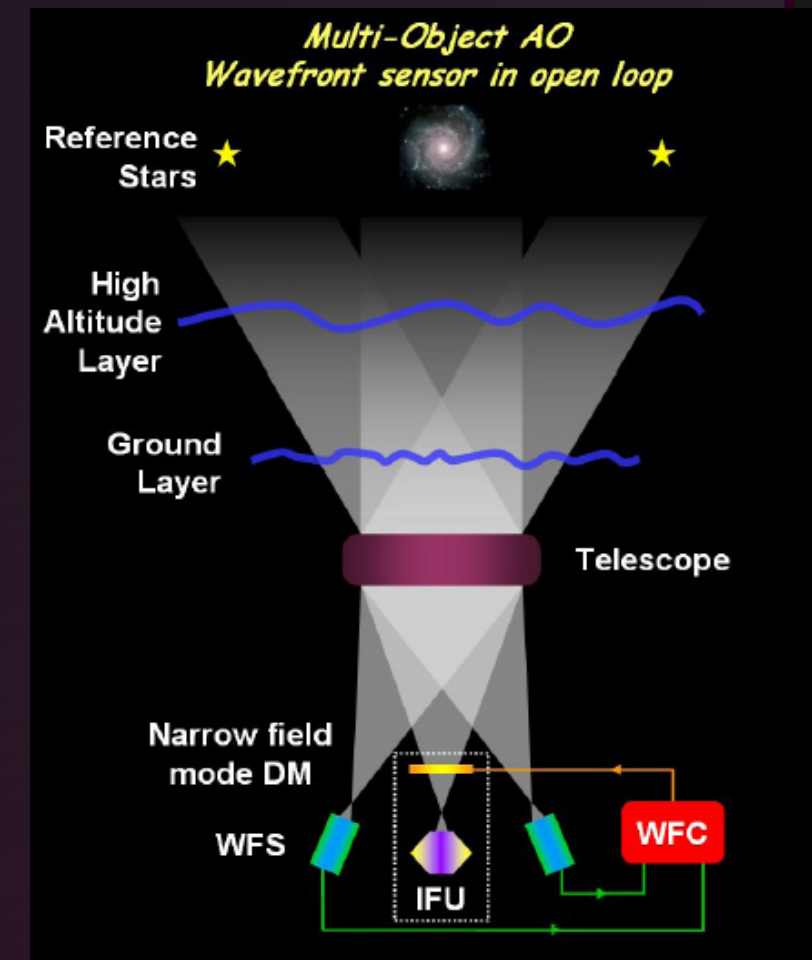
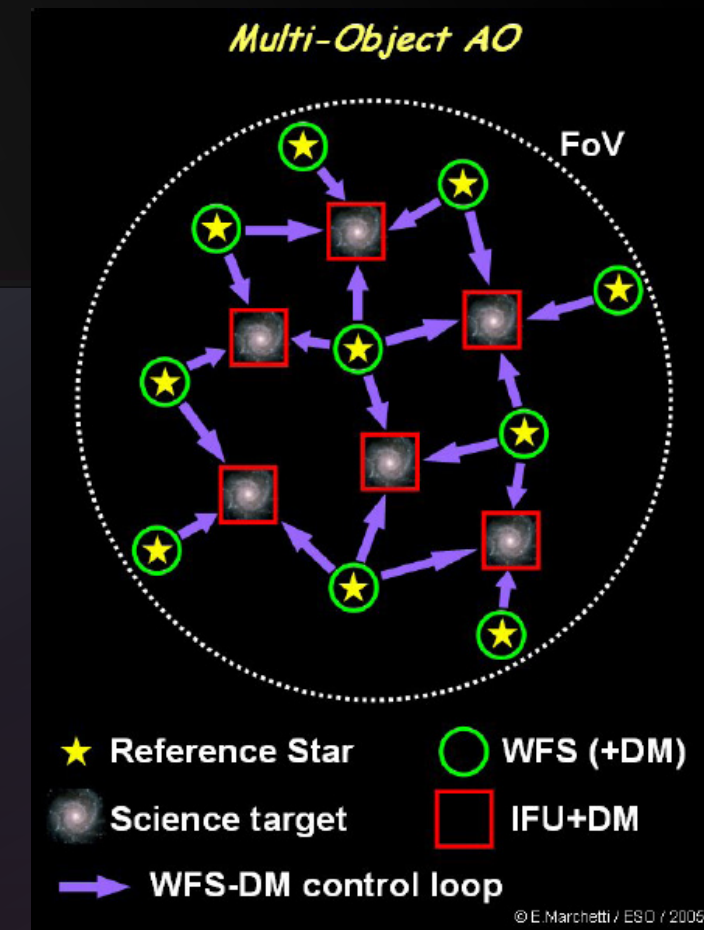


- Financement obtenu : 3.76M€ (call FET-HPC du programme Horizon2020)
- Kick-off du projet prévu en fin d'année, projet de 36 mois, 486 FTE



Vers les échelles extrêmes

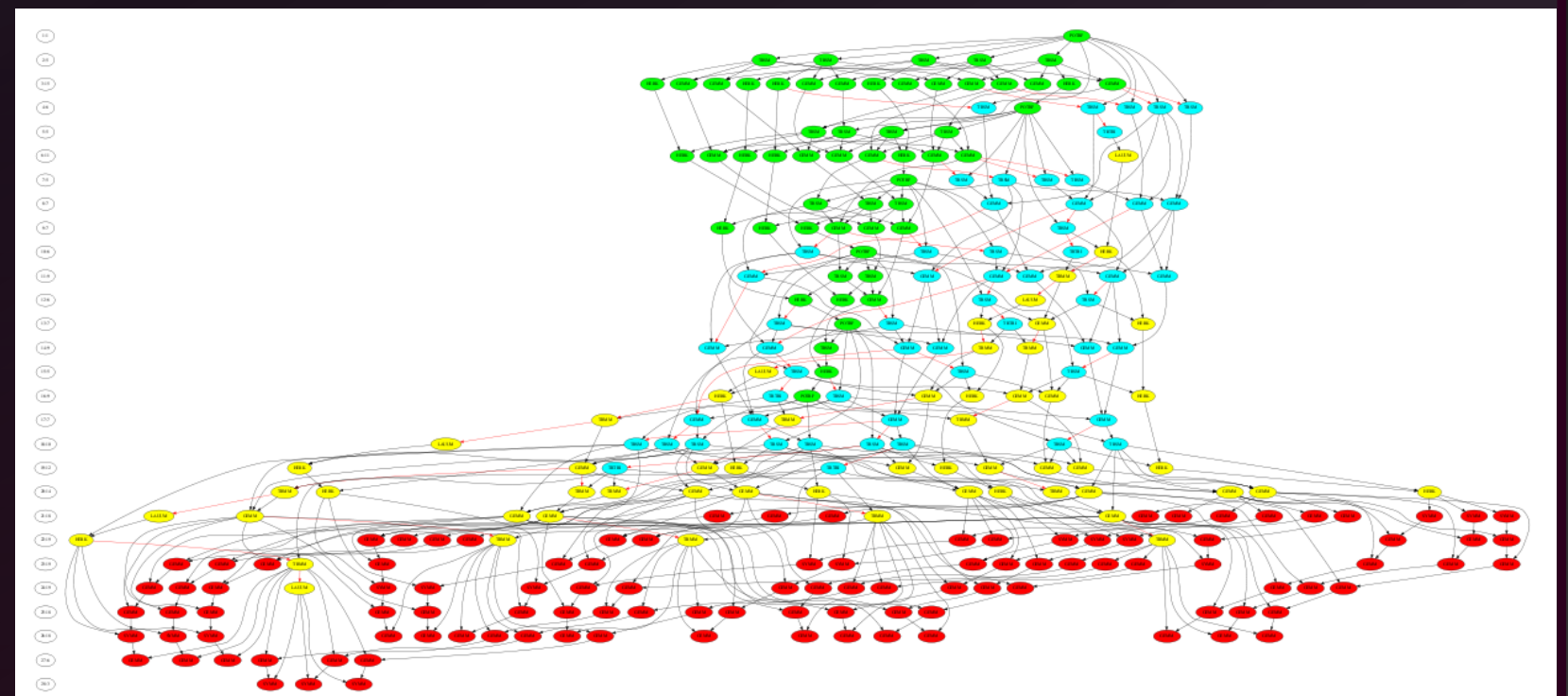
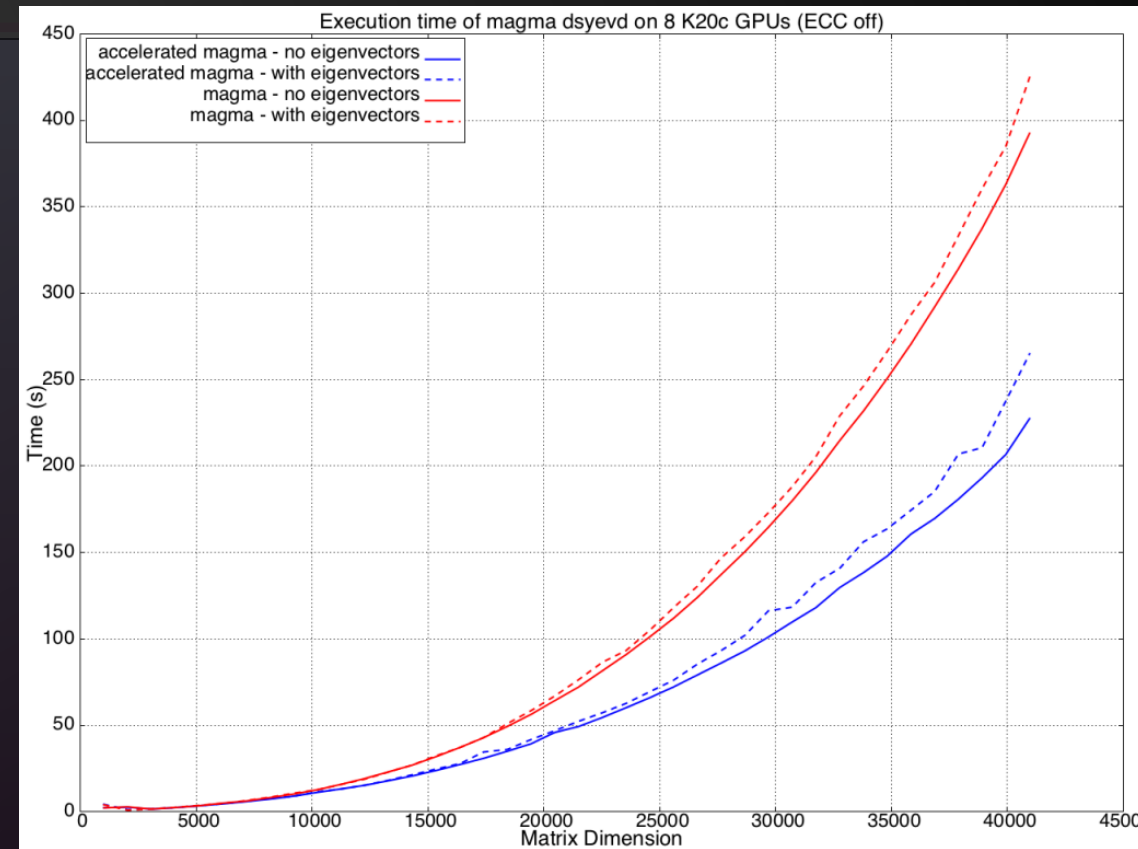
- Reconstructeur tomographique pour la MOAO sur l'E-ELT
 - 6 ASO Laser, 9 ASO « naturels », grand champ
- Opérateur linéaire pour le contrôle (MVM) : challenge majeur
 - Minimum variance: inverse de la matrice de covariance des mesures
 - Défi calculatoire : matrice dense 100k x 100k
- Utilisation d'algèbre linéaire optimisé
 - Décomposition en valeurs propres (symétrie)
 - Inversion directe après régularisation (Cholesky)
 - Collaboration avec Extreme Computing Group @ KAUST



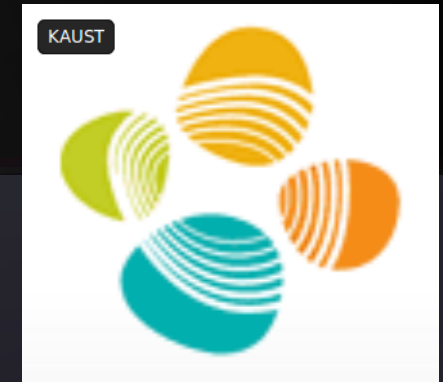
Vers les échelles extrêmes



- Optimiser les performance de DSYEVD
 - Résultats sur 8 GPUs
 - Avec ou sans calcul des vecteurs propres
- Pipeliner* le calcul complet du reconstituteur tomographique
 - Avec Cholesky
 - Travail en cours



Vers les échelles extrêmes



- ◉ Optimiser les performance de DSYEVD
 - ◉ Résultats sur 8 GPUs
 - ◉ Avec ou sans calcul des vecteurs propres
- ◉ *Pipeliner* le calcul complet du reconstructeur tomographique
 - ◉ Avec Cholesky
 - ◉ Travail en cours

Full tomographic reconstructor computation

