April 24-25

Nick Kornweibel

Control System Project Manager

http://www.eso.org/public/

@eso

@esoastronomy

The ELT
Control System, progress report

- Nick Kornweibel
  Control System Project Manager

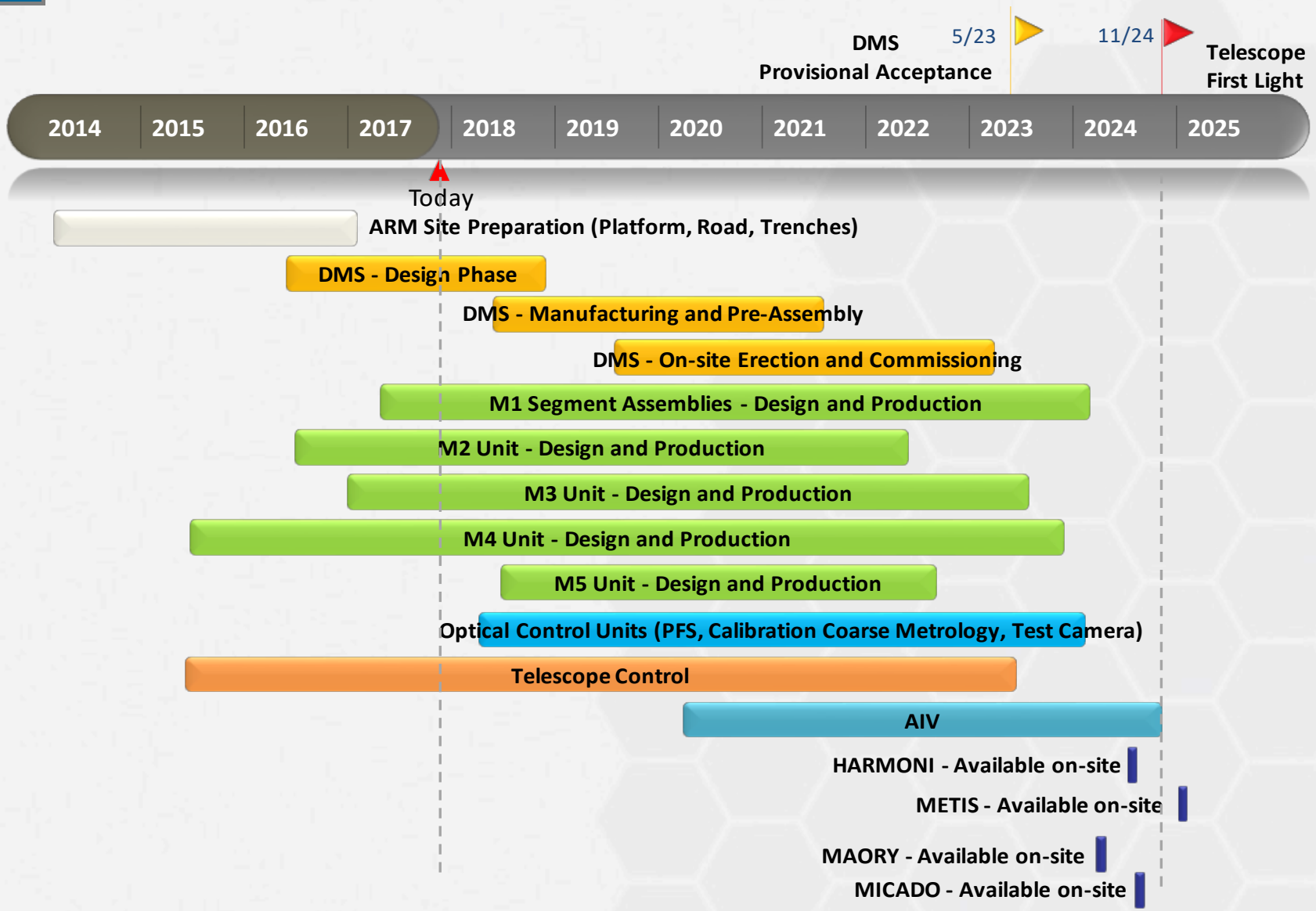
Armazones and Paranal

E-ELT
(Armazones)

Road and platform completed

VLT (Paranal)

ELT Schedule

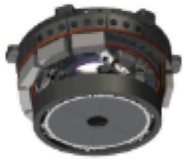


ELT Optomechanics

M1 Unit
39-m
Concave – Aspheric f/0.9
Segmented (798 Segments)
Active + Segment shape Control

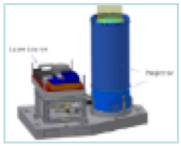
M2 Unit
4-m
Convex Aspheric f/1.1
Passive + Position Control

M3 Unit
4-m – Concave – Aspheric f/2.6
Active + Position Control



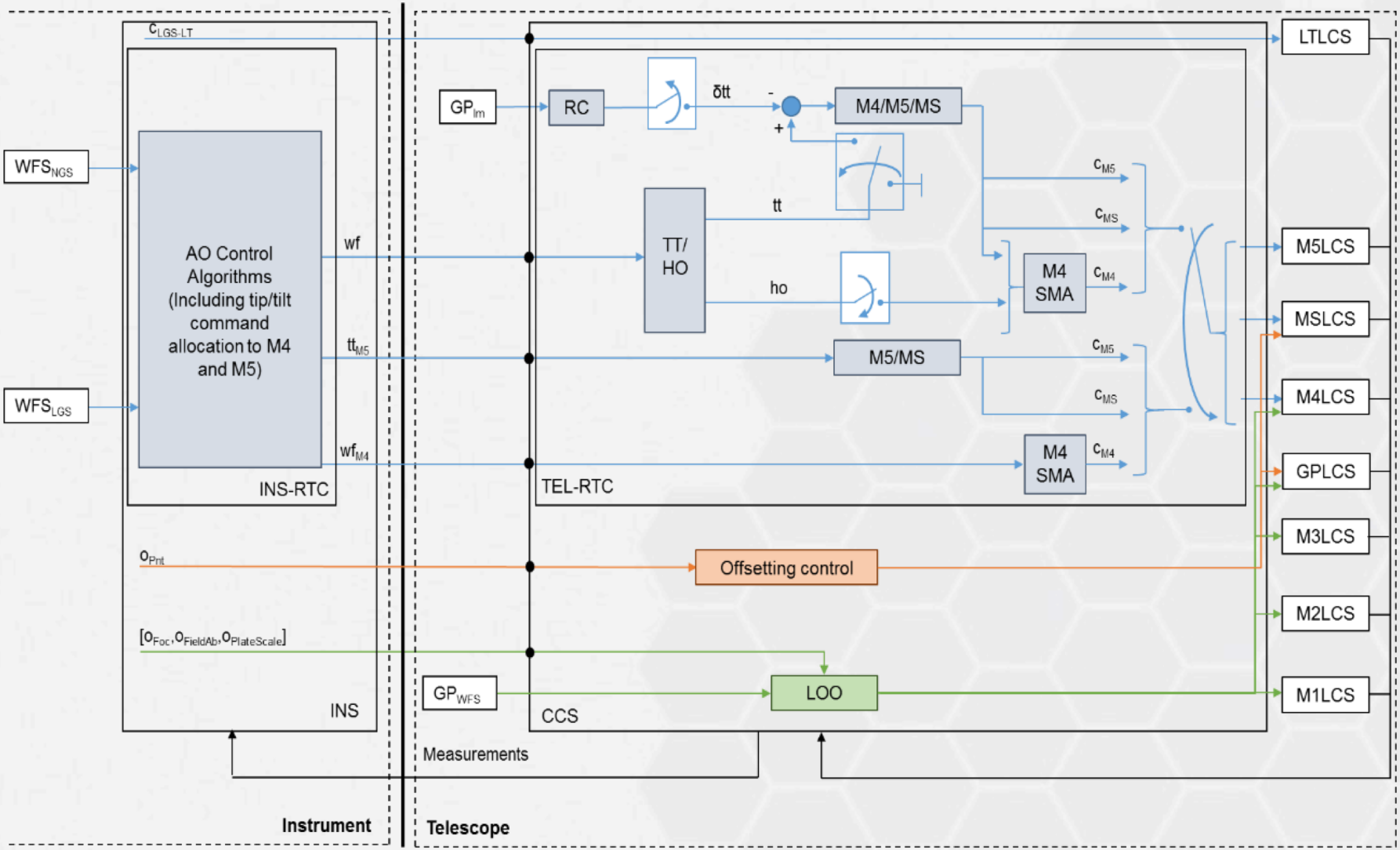
M4 Unit
2.4-m
Flat
Segmented (6 petals)
Adaptive + Position Control

M5 Unit
2.7x2.1-m
Flat
Passive + Fast Tip/Tilt



LGSU
(Laser Guide Star Units)
Laser Sources + Laser Beacons
shaping and emitting

ELT Control System

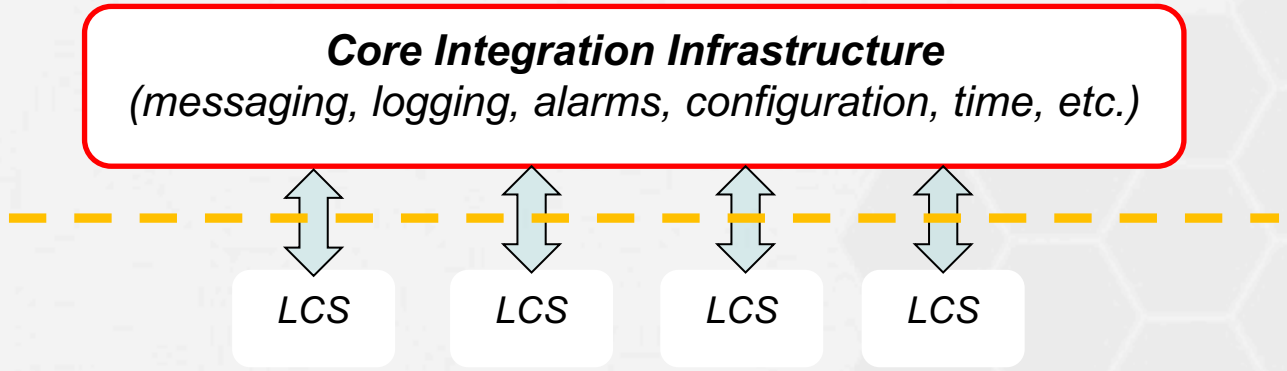


Control System Architecture

- System of Systems:
  - *Local Control System(s)* fully responsible for subsytem function and safety.
  - *Central Control System:* integrated control and telescope level safety.

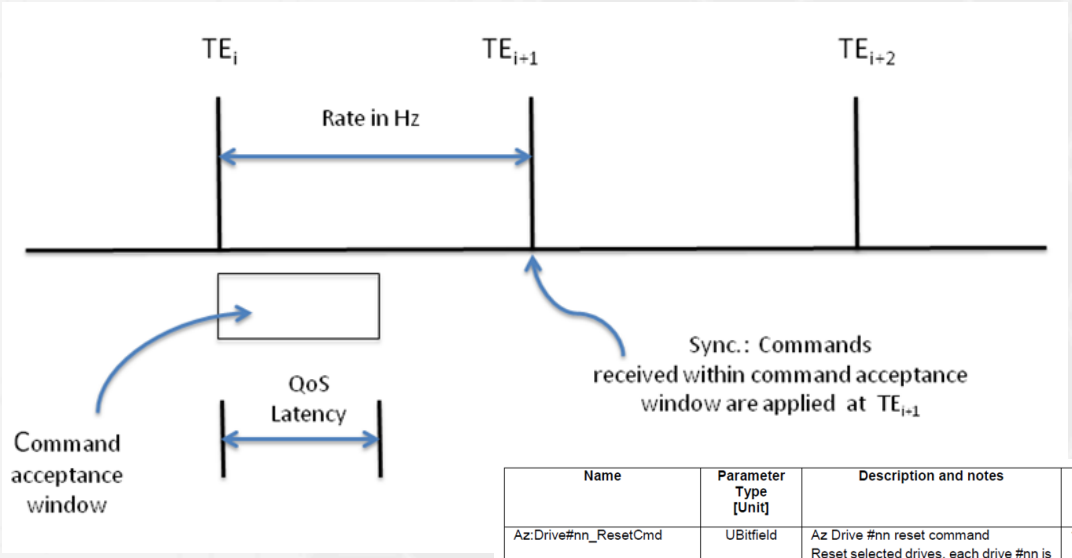
- Principals:
  - Separation of control and safety functions
  - Physical separation between computing units and field devices.
  - Usage of mainstream industrial standards.
  - Usage of mainstream COTS components.

Interface Definitions



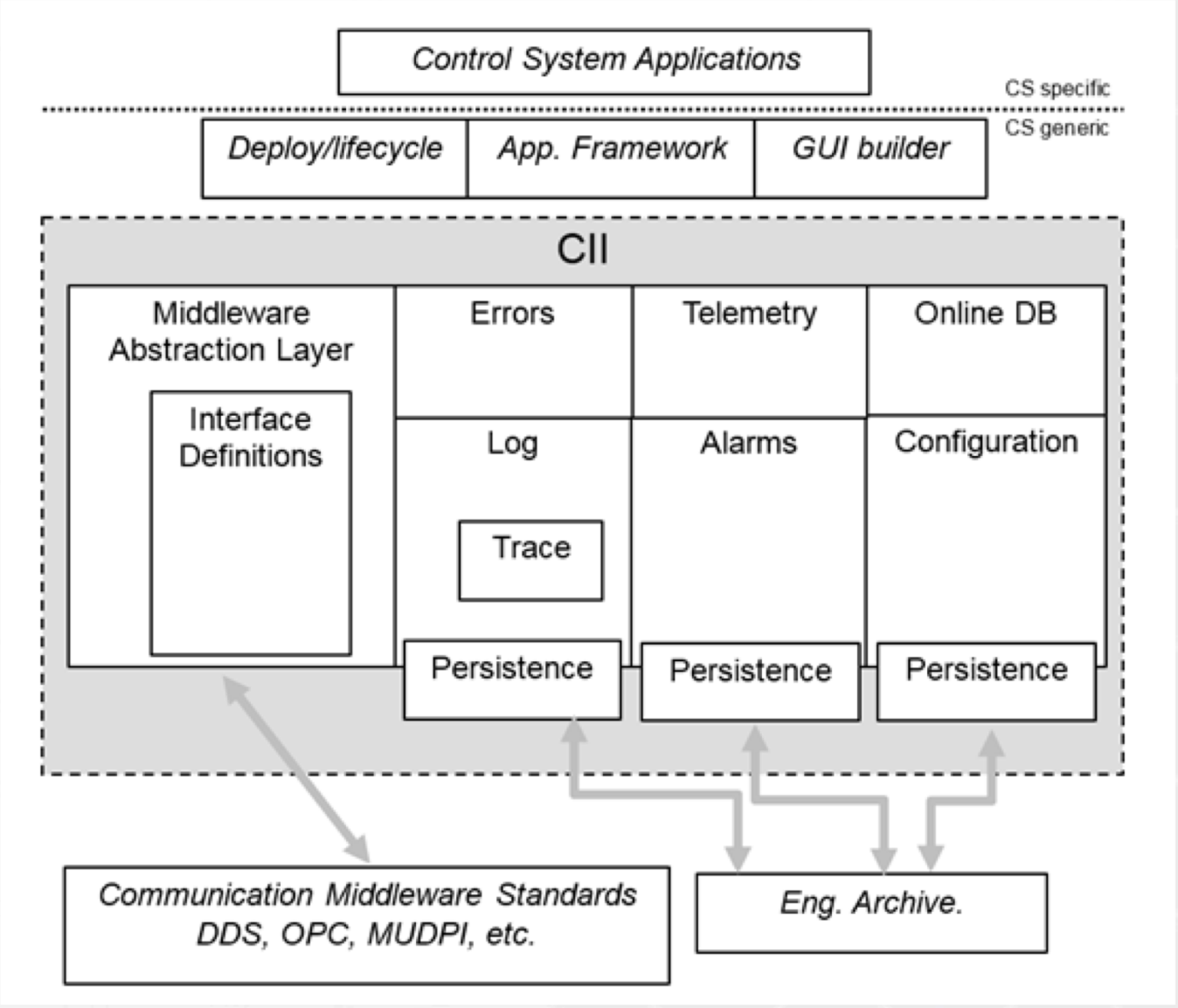
2 Patterns:
- Pub/Sub
- Cmd/Reply

4 Protocols:
- DDS
- OPC/UA
- ZMQ/ProtoBuf
- MUDPI

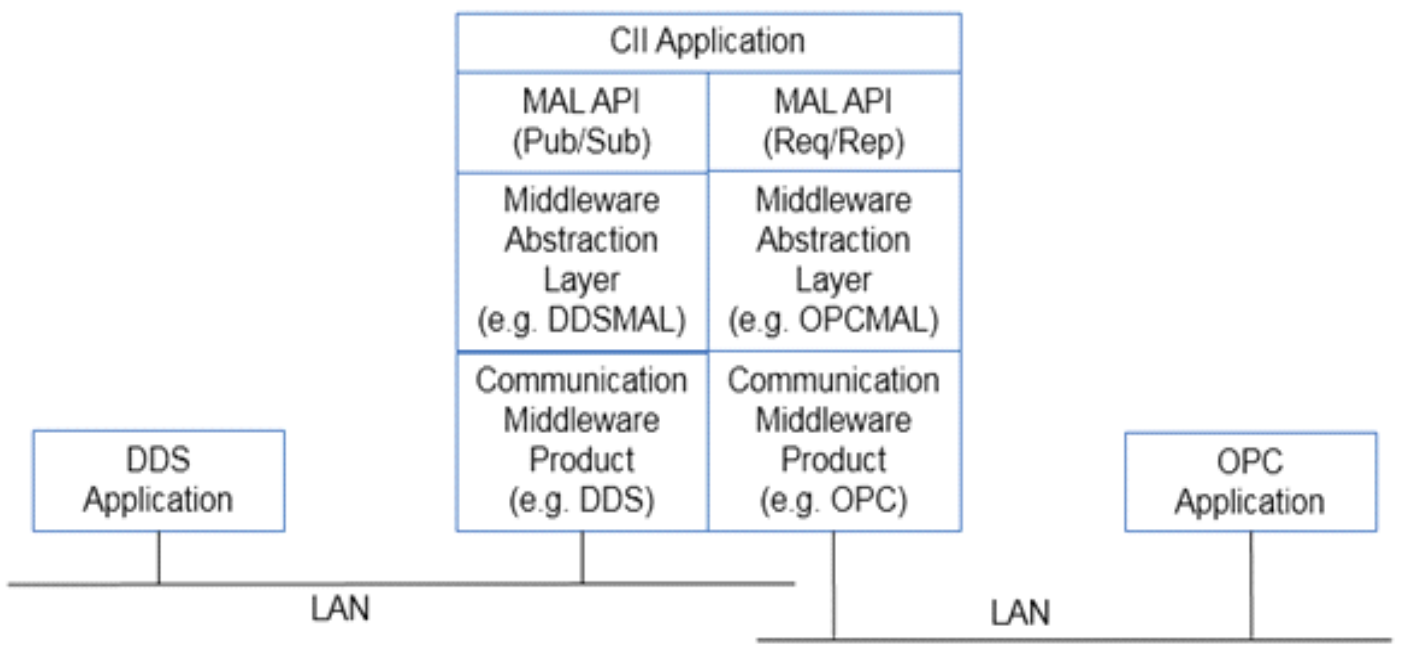


Name	Parameter Type [Unit]	Description and notes	Rate	Sync. /Async.	QoS Latency
Az:Drive#nn_ResetCmd	UBitfield	Az Drive #nn reset command Reset selected drives, each drive #nn is represented by one bit of the bitfiled, 1= drive reset, 0 = no action.	1Hz	Async.	1000ms
Az:Drive#nn_EnablCmd	UBitfield	Az Drive #nn enable command Enable selected drives, each drive #nn is represented by one bit of the bitfiled, 1= drive reset, 0 = no action.	1Hz	Async.	1000ms
Az:Drive#nn_DisablCmd	UBitfield	Az Drive #nn disable command Disable selected drives, each drive #nn is represented by one bit of the bitfiled, 1= drive reset, 0 = no action.	1Hz	Async.	1000ms

Core Integration Infrastructure

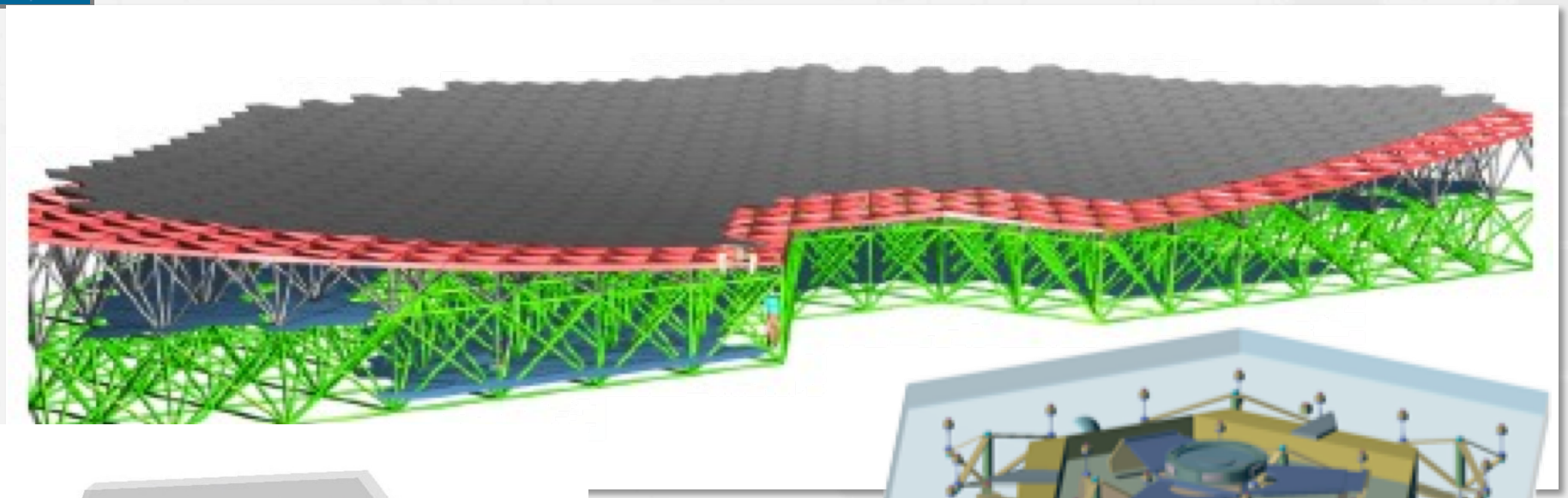


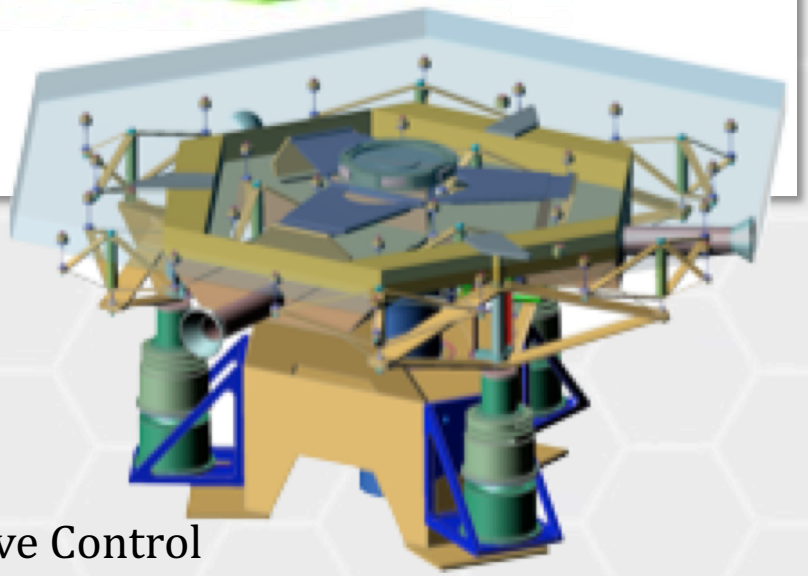
Middleware Abstraction Layer

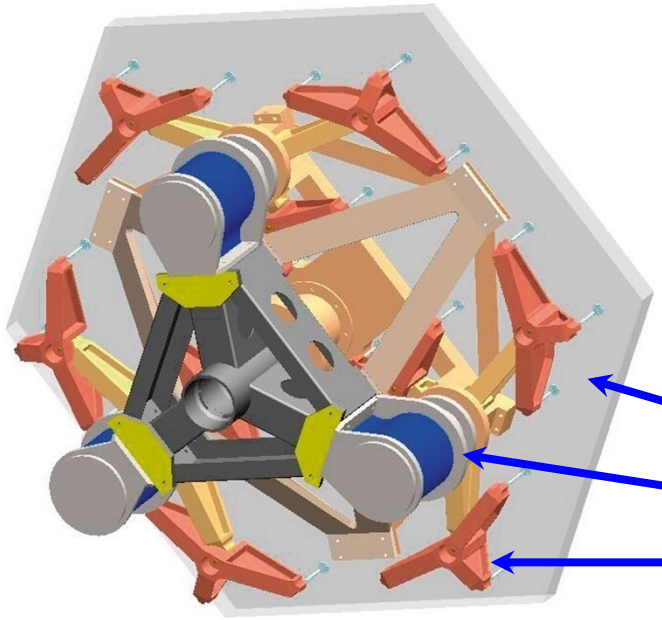


	Publish/Subscribe	Request/Reply
OPC/UA Data Access		
~~OPC/UA History~~		
OPC/UA Methods		
OPC/UA Events		
DDS		
ZMQ/Protobuf		
MUDPI		

The Primary Mirror (M1)



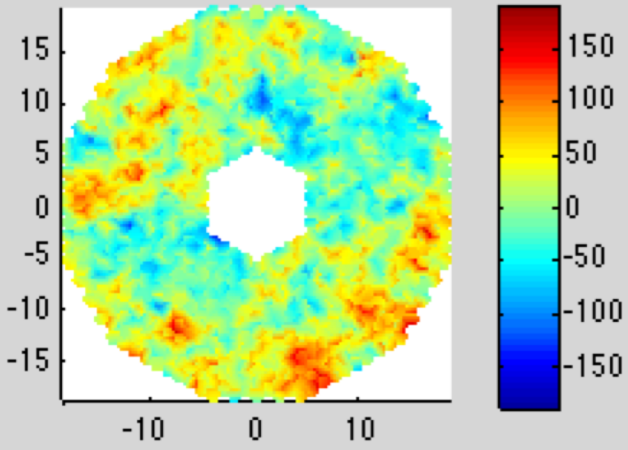




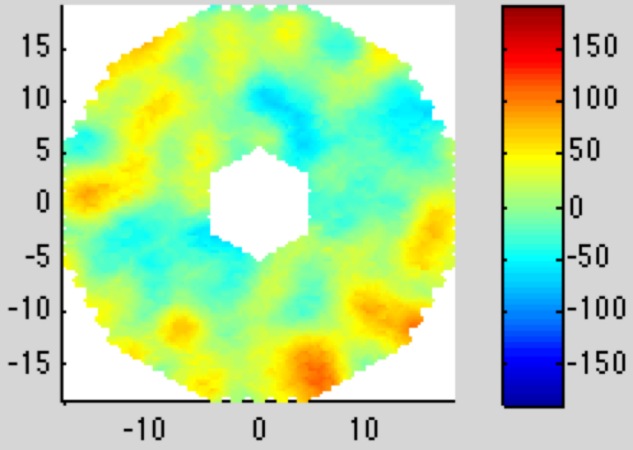
Segment Active Control

- 6 Edge Sensors (Piston/Shear/Gap)
- 3 Actuators (Piston/Tip/Tilt)
- 1 Surface Deformation harness

M1 Mirror Control

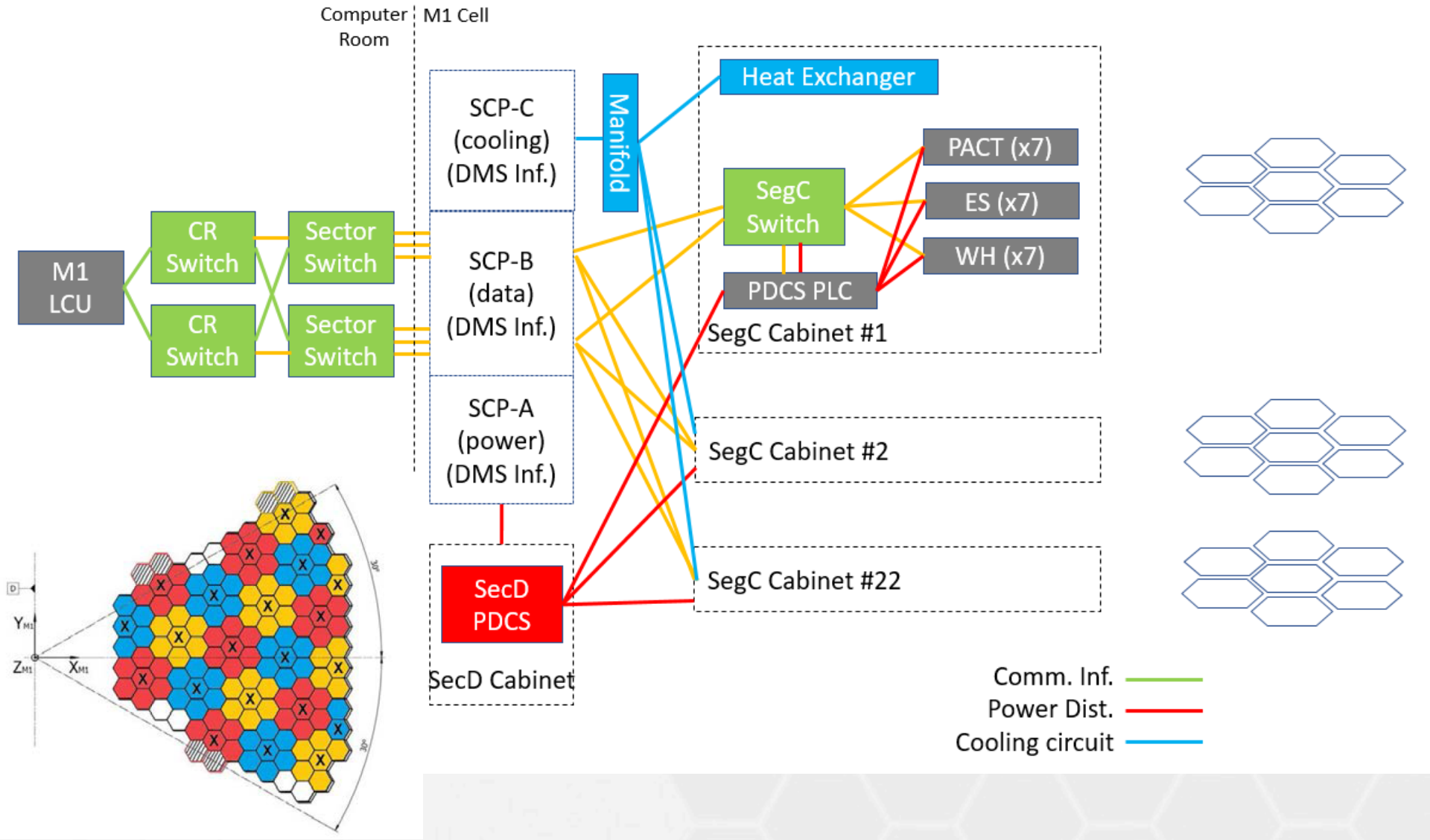


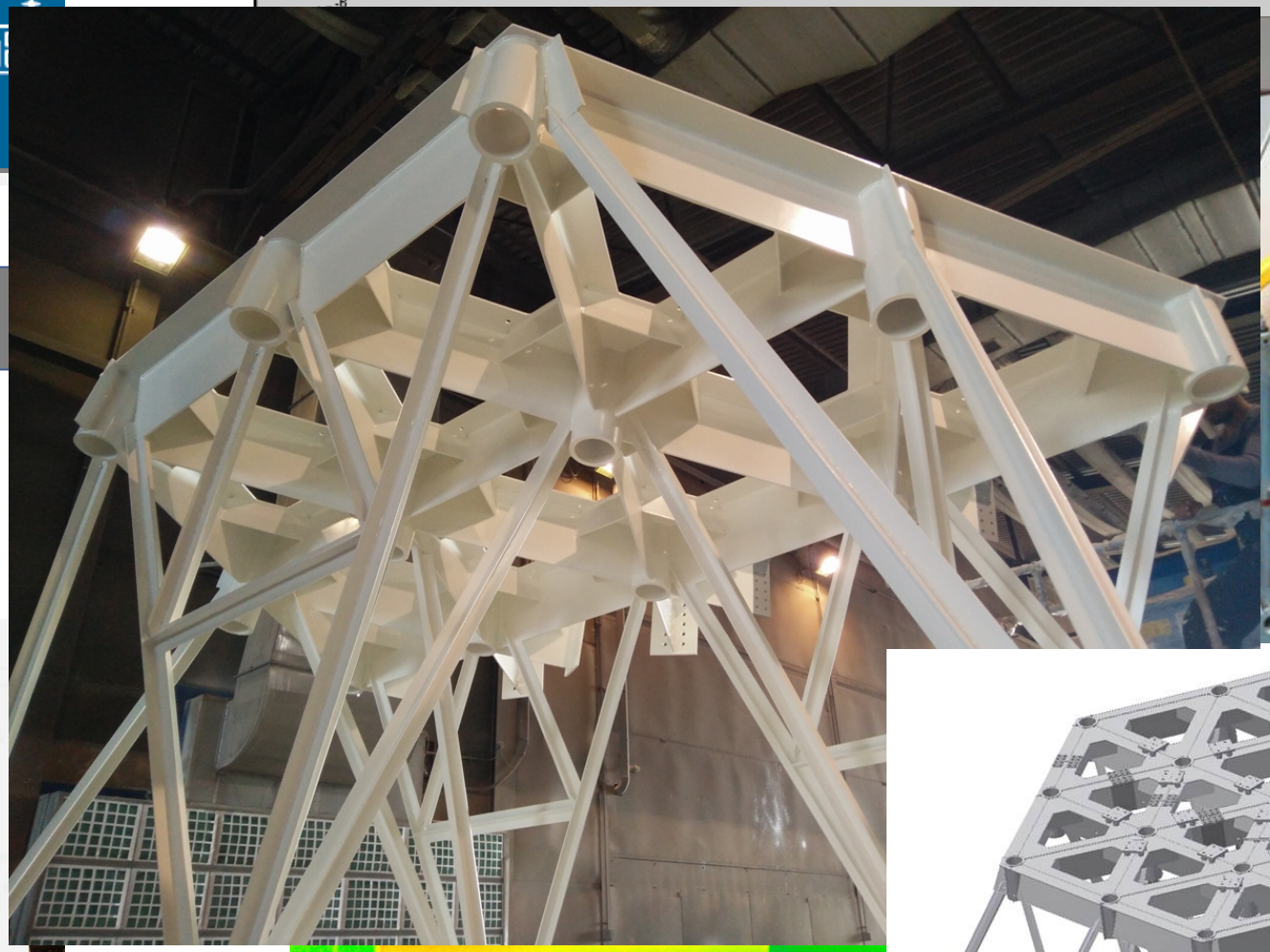
Hard PACT Before ES loop

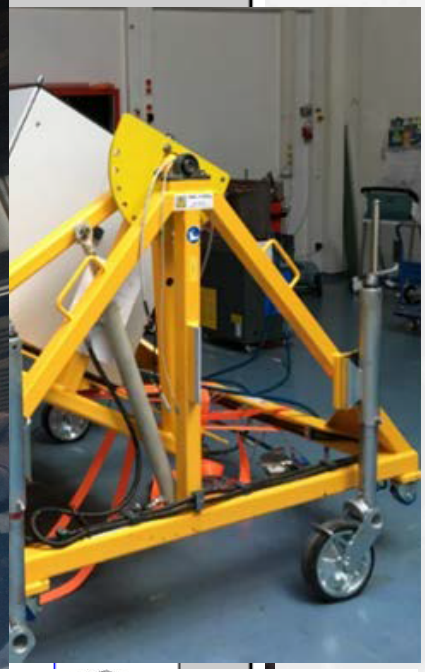


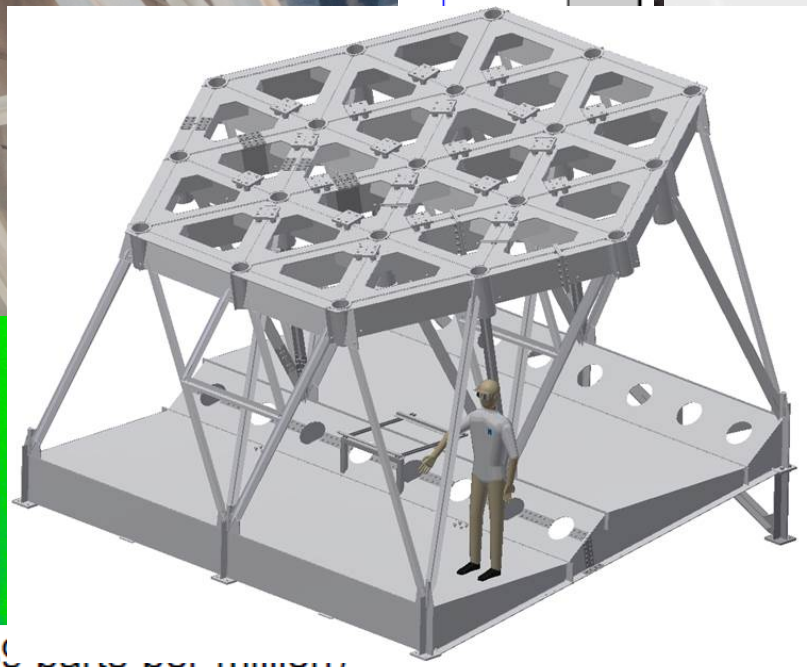
Hard PACT After ES loop

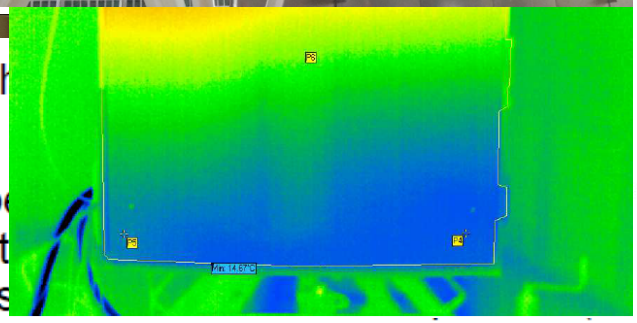
M1 Local Control System Progress











Recording th :

Total time

Total loop pe

Max absolut

Above 10 us

Above 15 us count 79 periods (0.3[illegible] parts per million)

CII & MAL Progress

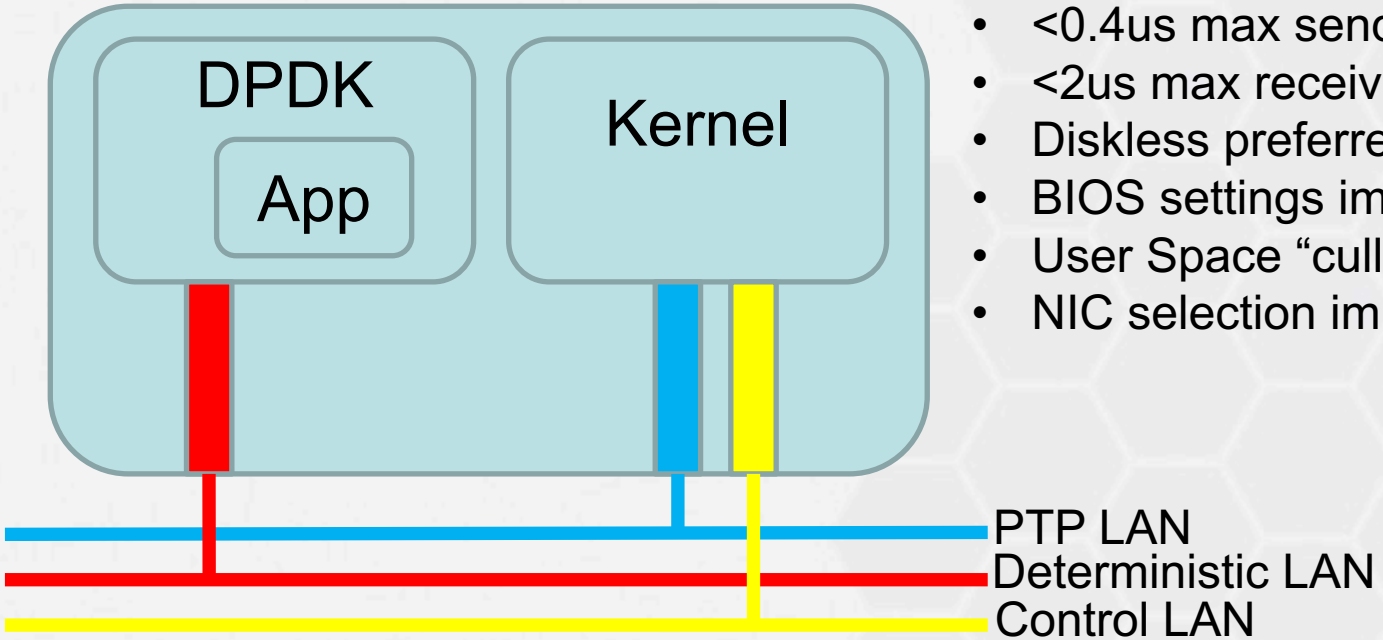
- Interface definitions in XML
- MAL APIs defined for Java/C++
- DDS MAL for C++/Java completed
- OPC MAL for C++/Java in development
- MUDPI MAL in design
- ZMQ/Protobuf MAL in design
- python and matlab stacks prototyped
- Learning to think in DCPS (!)

- OLDB: federated with scalar values only.
  - GUIs, scripts, reports.
  - E.g. Redis.

Realtime Computing Node Progress

Base RT Configuration:

- Cent OS (possibly with RT patch)
- Kernel contained and User Space “quietened down”
- User Space IP stack (DPDK)
- Precision Time Protocol
- MUDPI (multicast UDP)

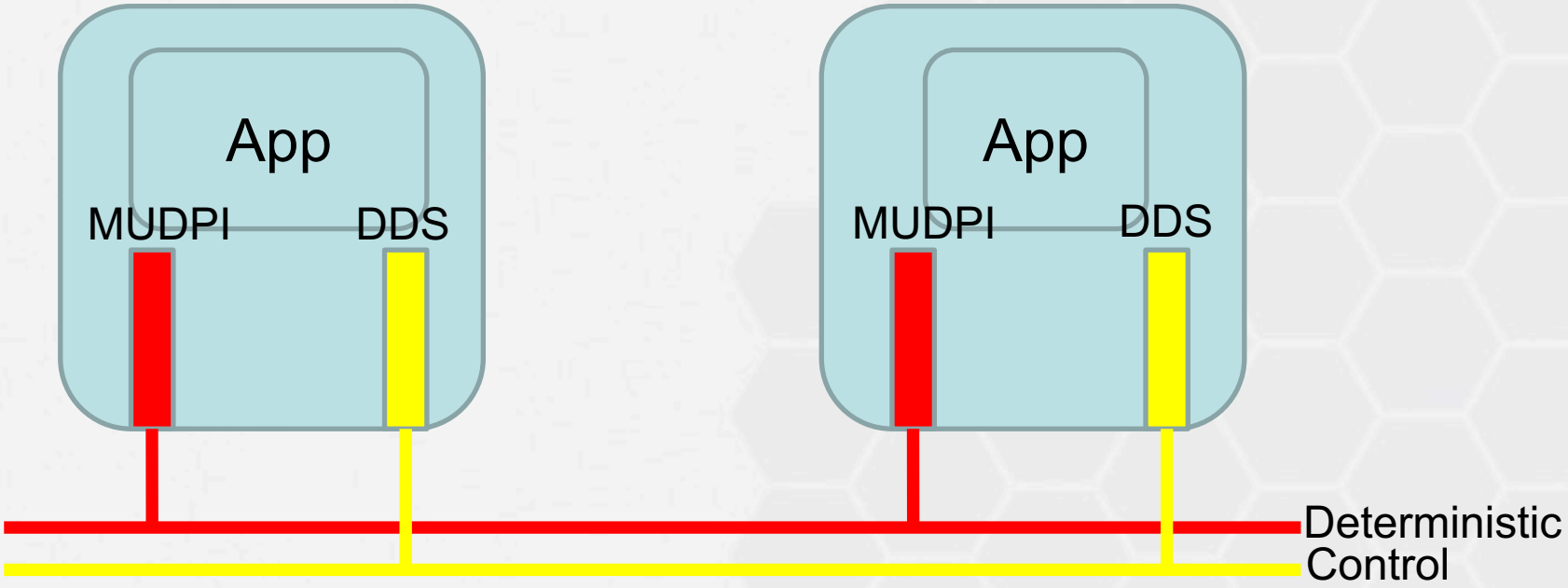


Results so far:

- Standard Dell + CISCO 10GigEth
- <0.4us max send jitter (0.2us RMS)
- <2us max receive jitter (0.6us RMS)
- Diskless preferred
- BIOS settings important
- User Space “culling” not finalized
- NIC selection important

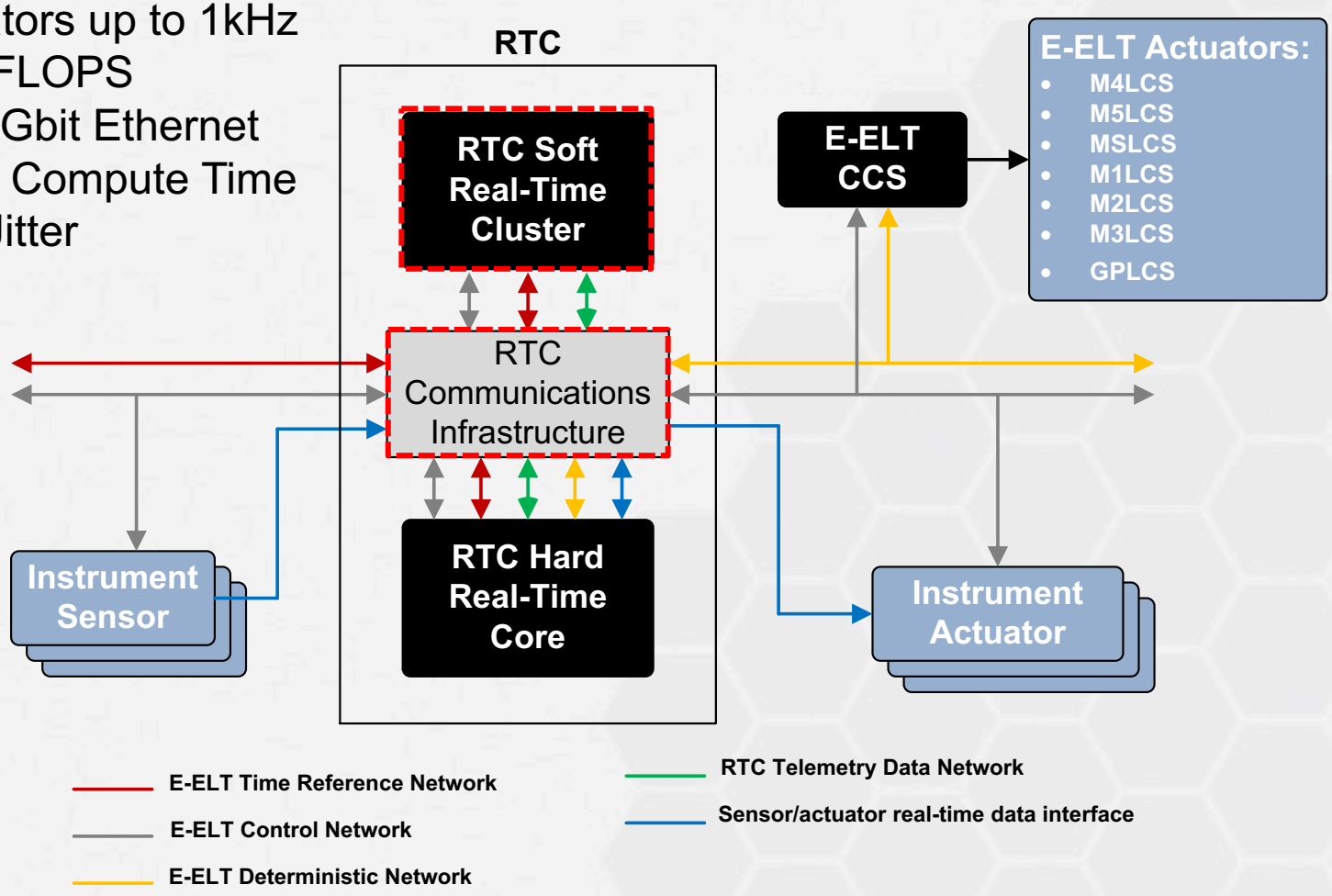
Realtime Computing Node

Are there patterns to preserve achieved communication performance, but leverage much of DDS?

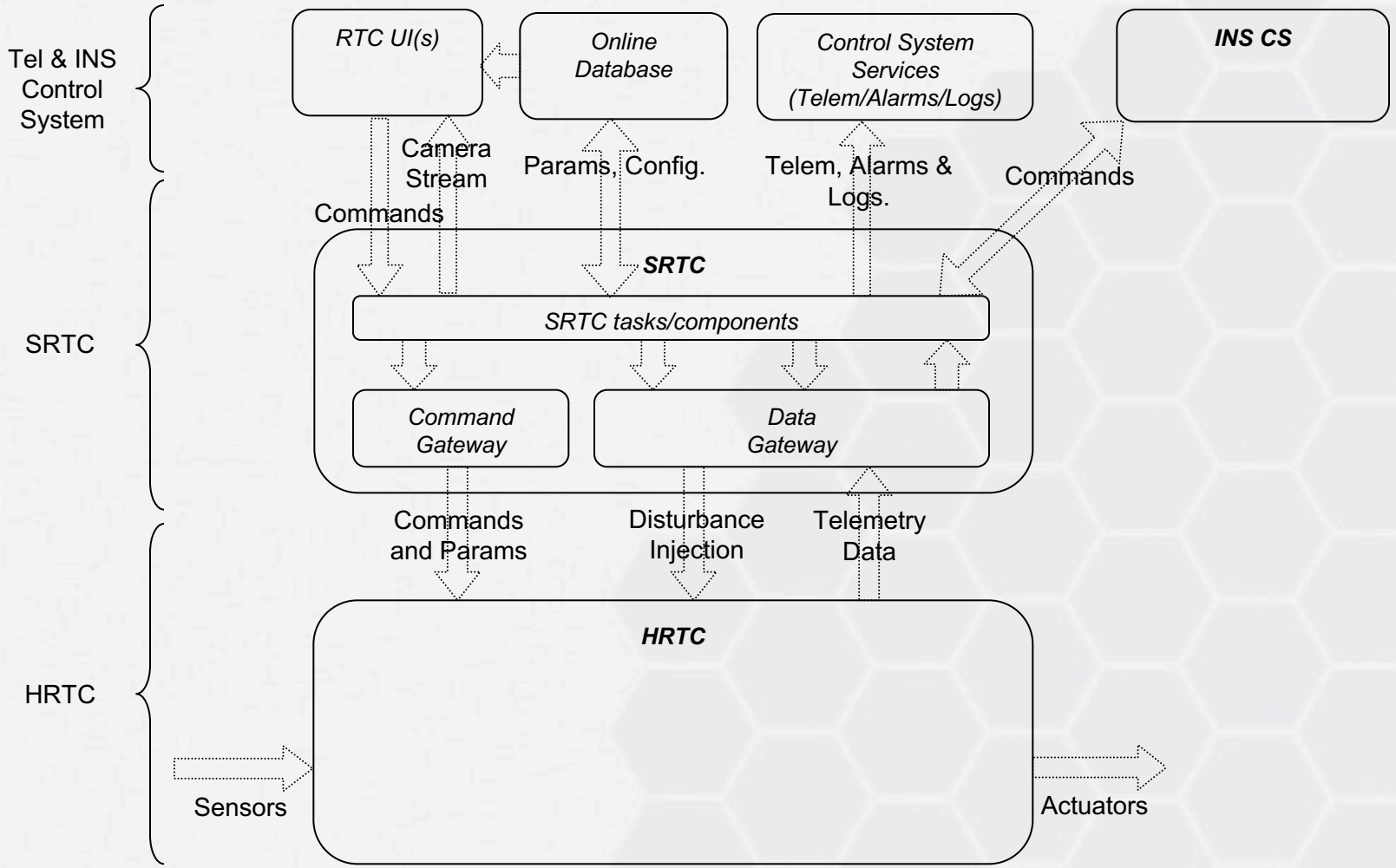


AO Real-Time Computer Progress

- Sensors up to1kHz
- Actuators up to 1kHz
- 1.4 TFLOPS
- 10/40Gbit Ethernet
- ~1ms Compute Time
- Low Jitter



Telescope RTC Pattern



Testing on the Mini ELT (MELT)