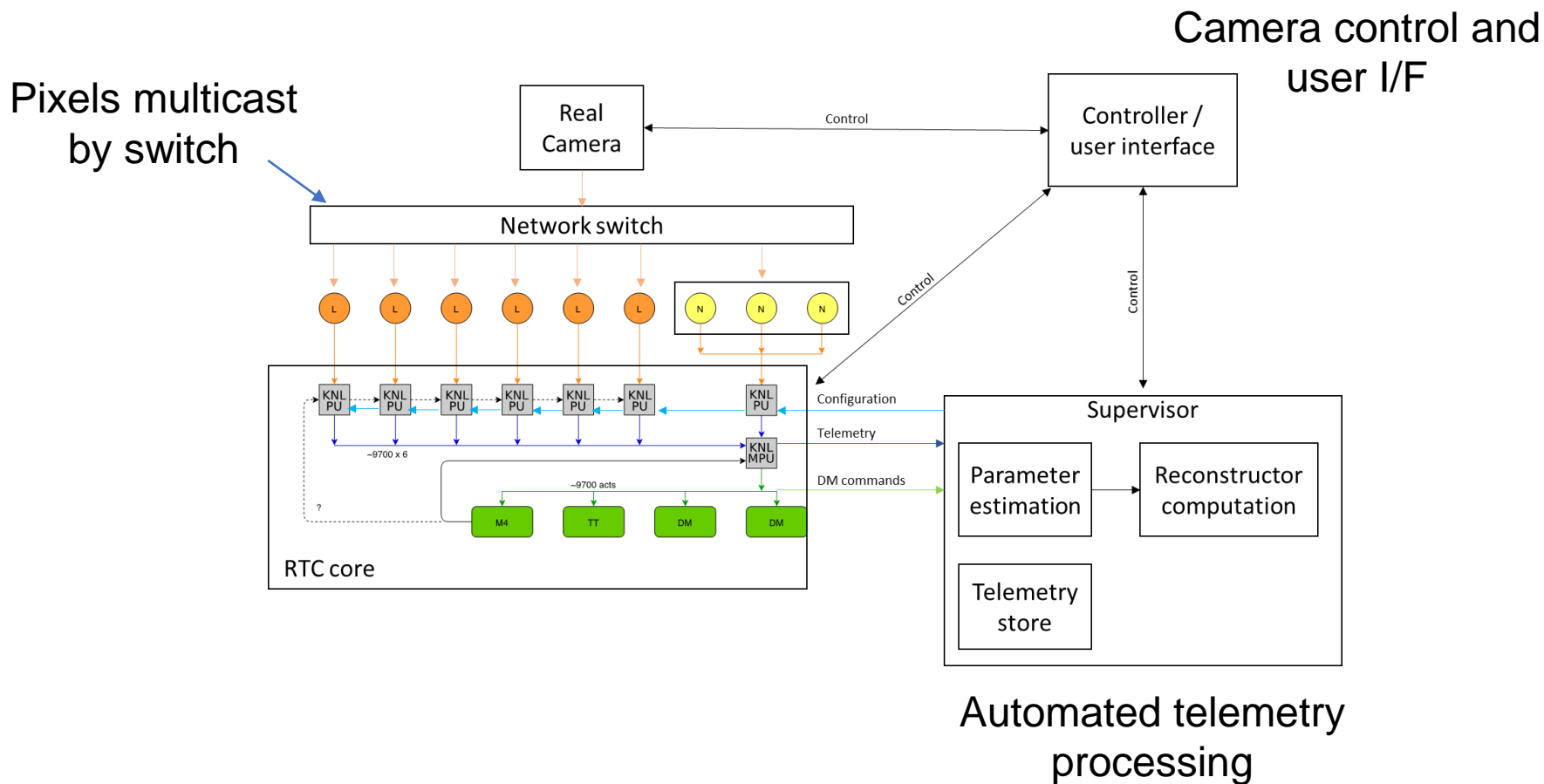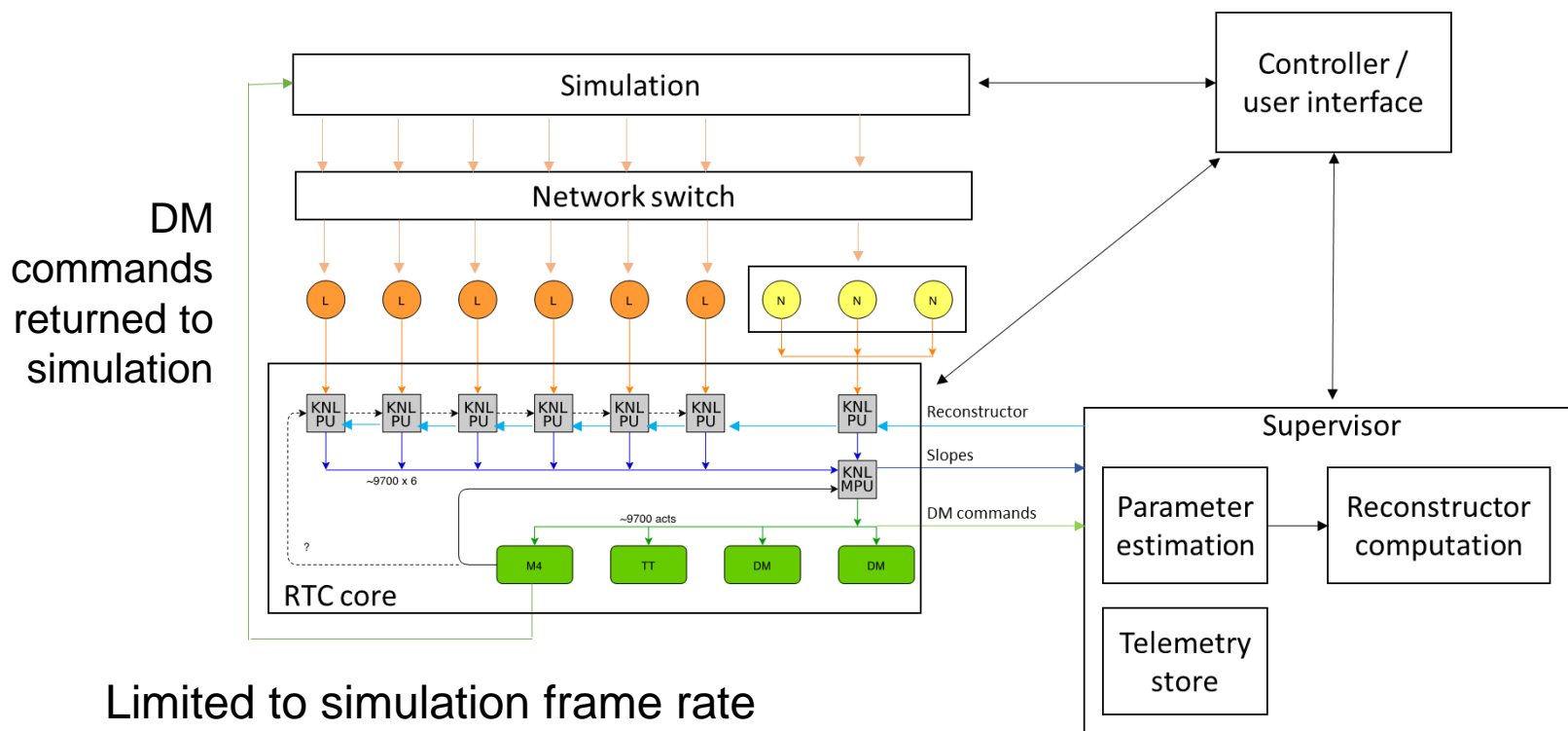# Final Architecture

CPU/MIC Prototype

Outline

- MIC/CPU prototype architecture
  - Several related configurations required for full test procedure

- Hard Real Time has three potential configurations
  - XeonPhi SCAO
  - XeonPhi Tomographic
  - AMD Epyc SCAO

- Here we concentrate on the Phi Tomographic System
  - Requires some network reconfiguration to switch between them

- Show example

Real Camera Configuration

Camera control and user I/F

Pixels multicast by switch

Automated telemetry processing

End-to-end simulation

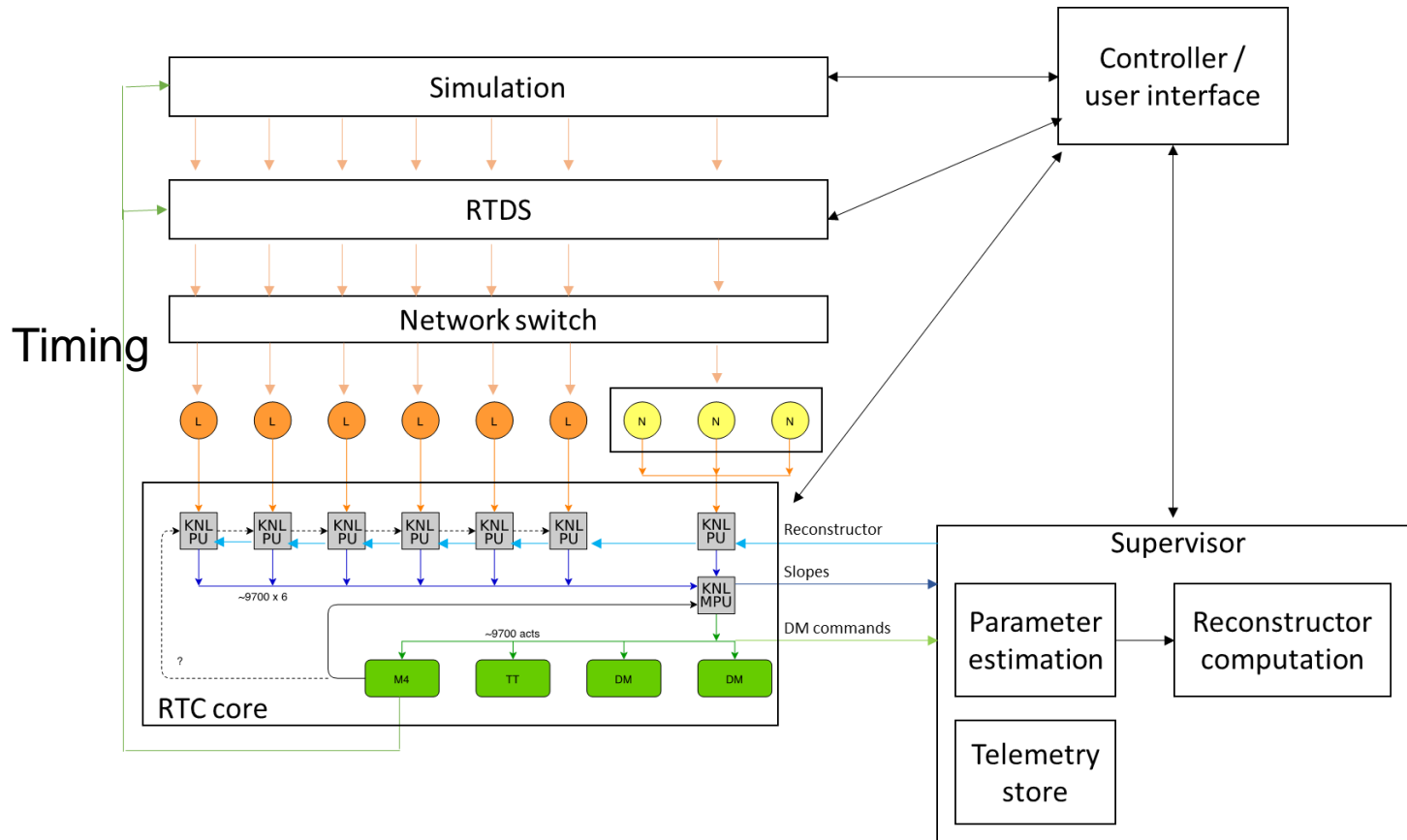DM commands returned to simulation

Limited to simulation frame rate
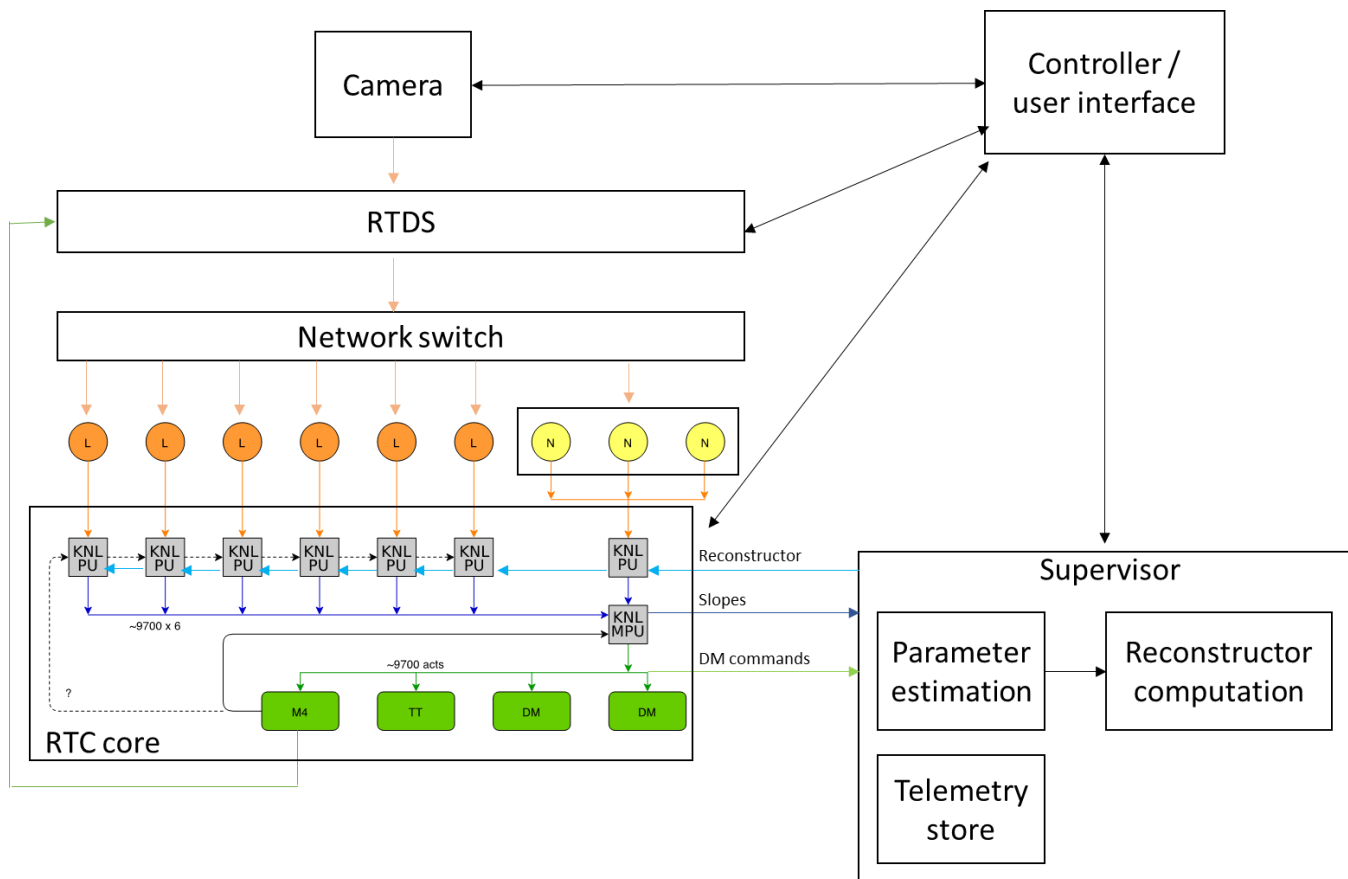Large jitter
Test feedback and control

Real-time Simulation

RTDS: Real-time data shaper buffers data with fixed jitter (500Hz)

Also monitors RTC input/output for external latency/jitter measurements

Latency with Cameras

RTDS used to determine start of frame and DM update times.
Stabilises input jitter if you want, but only really used for measuring latency

Supervisor Architecture

- Lots of benchmarking results using both Xeon Phi and Xeon
  - No full/optimised implementation yet

- CPU implementations generally not as fast as GPU (may be optimised further) e.g.
  - CPU 60k matrix inversion in ~150s
  - GPU 60k matrix inversion in ~24s

- Part of the work we've done is to reduce problem size e.g.
  - Atmospheric/WFS parameter estimation from WFS slope data
  - Calculating only covariance elements along WFS baselines in 60k x 60k matrix increases speed by 2-3 orders of magnitude
  - GPU full matrix completes in ~25s (Damien's WP4 talk)
  - CPU (Dual Xeon 2016) goes from ~1000s to 11s

Supervisor/Soft RT Tasks

GreenFlash	Non-GreenFlash
Pseudo Open-Loop Slopes?	M4 Position Monitoring
Parameter Estimation	Projection Matrices
Covariance Matrix Generation	Pupil Position
Matrix Inversion	LGS/NGS Centroid Gains
Reference Slope Updates	PSF Reconstruction

- Internal Soft RT tasks have update rates in terms of seconds
  - Large problems with large jitter is ok – no external timing

- Interface tests between Hard/Soft(+Telemetry storage) e.g.
  - Impact of commands and additional telemetry streams on latency and jitter
  - Impact of saving pixels…

Summary

- Two architectures running the same code with a different configuration file
  - Phi for MCAO and SCAO
  - AMD Epyc for SCAO
- Several test configurations required for different tests
  - Simulation for AO performance (are we getting the right numbers?)
  - Real-time data shaper for latency/jitter measurements
  - Camera (and multicast) for end-to-end 'real-world' configuration
- Still need to precisely define algorithms/test cases for comparison