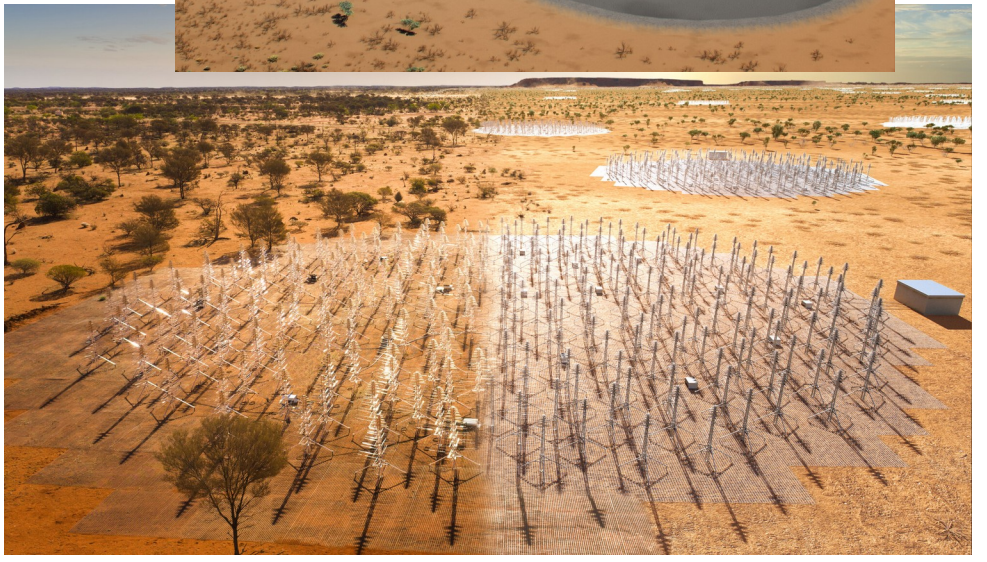
Co-design for the SKA project

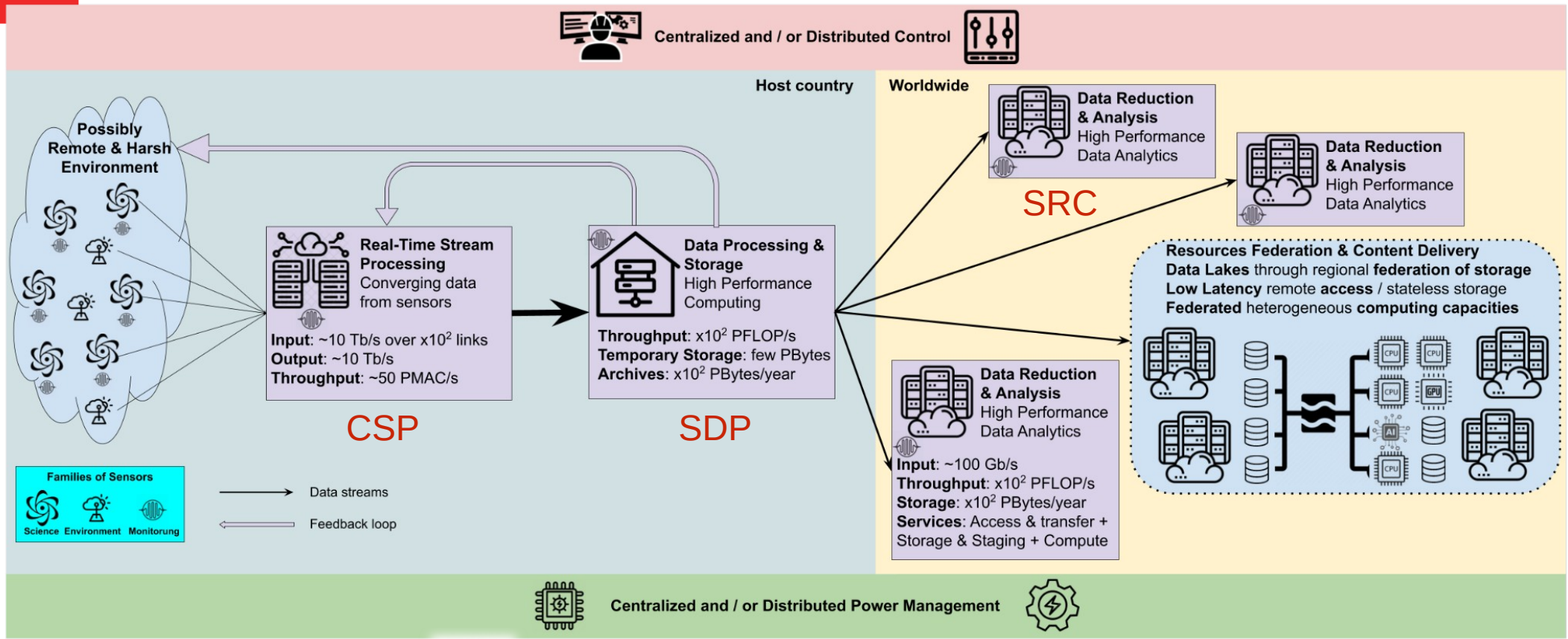
Shan Mignot
Observatoire de la Côte d'Azur

Overview of SKA

- 2 phased-array radio-telescopes
  - Mid in South Africa:
    - 197 dishes
    - baselines up to 160 km
  - Low in Australia
    - 512*256 antennas
    - baselines up to 75 km

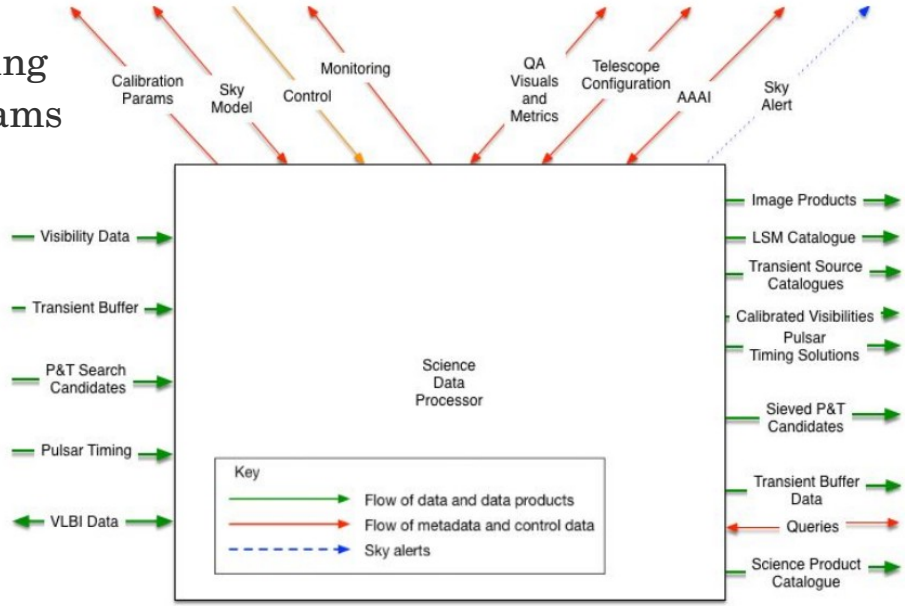


Supercomputing for SKA



Computing infrastructure (SKAO)

- Central Signal Processor (CSP)
  - Input: digitized waveforms
  - Outputs
    - Correlator: visibilities for interferometric imaging
    - Pulsar search: identify candidate pulsars in beams
    - Pulsar timing: measure pulsar timings
- Science Data Processor (SDP)
  - Input: data from CSP
  - Outputs
    - Science data products
    - Telescope/CSP feedback and calibration
    - Alert generation
    - First look



Computing infrastructure (SRCs)

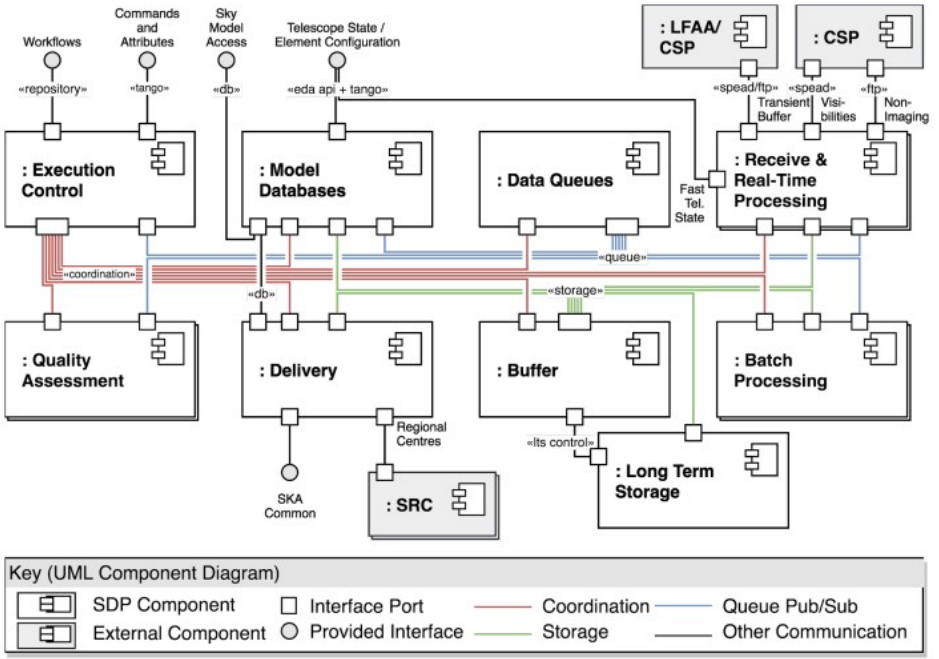
- Science Regional Centers (SRC)
  - 5 interconnected SRCs
  - Distribution of data to scientist / archive
  - Science analysis software

The SDP challenge

- 2018 assessment at the Critical Design Review (per SDP)
  - Compute: ~125 Pflops peak > 10 Pflops average
  - IO
    - Ingestion: ~1 TB/s
    - Storage: ~40 PB
    - Data accesses: ~10 TB/s
  - Budget: 35 M€
  - Extrapolation to procurement deadline based on Moore's law
  - 50-year lifetime
- Comparable to Sunway Taihulight (6th rank in Top500) but SKA is NOT an HPC project
- Evolution since
  - Power
    - Mid — Peak (< 5s) — Average (> 30 min)
      - Servers: 2 MW — 1.3 MW
      - Building and cooling: 0.56 MW — 0.37 MW
    - Low
      - Servers: 2.2 MW — 1.6 MW
      - Building and cooling: 0.45 MW — 0.32 MW
  - Staged deployment (procurement schedule not final)
  - Cost and availability of HW

SDP baseline

- 4 high level tasks
  - Receive and condition instrument data, including real-time processing
  - Buffer data
  - Batch processing of conditioned data
  - System management tasks
- Node architecture deriving from a single server model
  - Cores: latency, throughput
  - Storage: capacity, performance
  - NIC: throughput, latency, management
- Nodes configured to assume 4 “personalities” predefined or defined at run time
- Assumption
  - HW architecture defined by supplier after a call for tender
  - But… risks regarding performance, cost, feasibility (interfaces being pre-defined)



Co-design for the SDP

- Risk mitigation activity
  - De-risk procurement of SDP
  - Improve investment: performance/cost ratio
- SCOOP team in SKAO's SAFe framework (contact Shan Mignot)
- ECLAT "laboratoire commun" (contact Damien Gratadour)
  - CNRS/Inria/Atos
  - R&D to feed co-design work
- What about FPGAs?
  - Baseline is a mix of CPU/GPU (ratio to be determined)
  - FPGAs considered as a possibility should need arise and benefit be established
  - Concern regarding maintainability

FPGAs for the SDP

- Accelerators
  - Specialised logic for kernels
    - Pipelining and parallelisation: efficient use of resources
    - Adjust precision: resources
    - Fine-tune clock domains: energy
  - Reconfigurable depending on task
- “Real-time” processing
  - Not hard real-time or strong latency constraints
  - Process the stream of data
    - Pre-processing prior to buffering (reduce volume)
    - Feedback loops to telescope and CSP
- Data management
  - High-speed IOs
  - Offload management from computing units
  - Hide data movements

FPGAs for the CSP

- CSP has FPGAs
  - Key component for correlators
  - Also central for pulsar search (PSS)
    - Baseline architecture is a mix of CPU/FPGA/GPU
    - Set-based development strategy:
      - Developing multiple versions CPU/FPGA/GPU for most modules
      - Select HW as late as possible
  - CSP code assumed stable during survey (no maintenance)
- Developments for FPGAs
  - > 40 000 lines of VHDL (PSS)
  - Porting code to FPGA/GPU considered a similar effort (based on cost of subcontracting)
  - Issue with finding the risk skills
  - Evaluation of OpenCL: performance considered disappointing

Perspectives

- Performance: ability to exploit FPGAs efficiently
  - Parallelisation
  - Pipelining
  - Custom data types
  - Re-using resources
  - Clock domain management
  - Inspection of timing
  - Interfacing with VHDL (as Assembly with C)
- Integration in a HPC system
  - HW and SW interfaces
  - Re-configuring FPGAs during operations
- HLS for SDP: acceptability
  - Community of SW developers (although FPGAs require a different mindset)
  - Improve maintainability of code
  - Maturity of HLS and available expertise
- Evaluation activities to prove added value (as part of SCOOP/ECLAT)
  - Select SDP task/kernel
  - Prototype HLS port
  - Assess value: return on experience, performance, power, cost
- Contributions welcome!