SDP Update

SKA Engineering Meeting
Rotterdam
Version 1.0

Jeremy Coles
SDP Project Manager

12th June 2017

Design of the computing hardware platforms, software, and implementation of algorithms needed to process science data from the correlator or non-imaging processor into science data products.

UNIVERSITY OF CAMBRIDGE

SKA SQUARE KILOMETRE ARRAY

SDP

SCIENCE DATA PROCESSOR

Acknowledgements

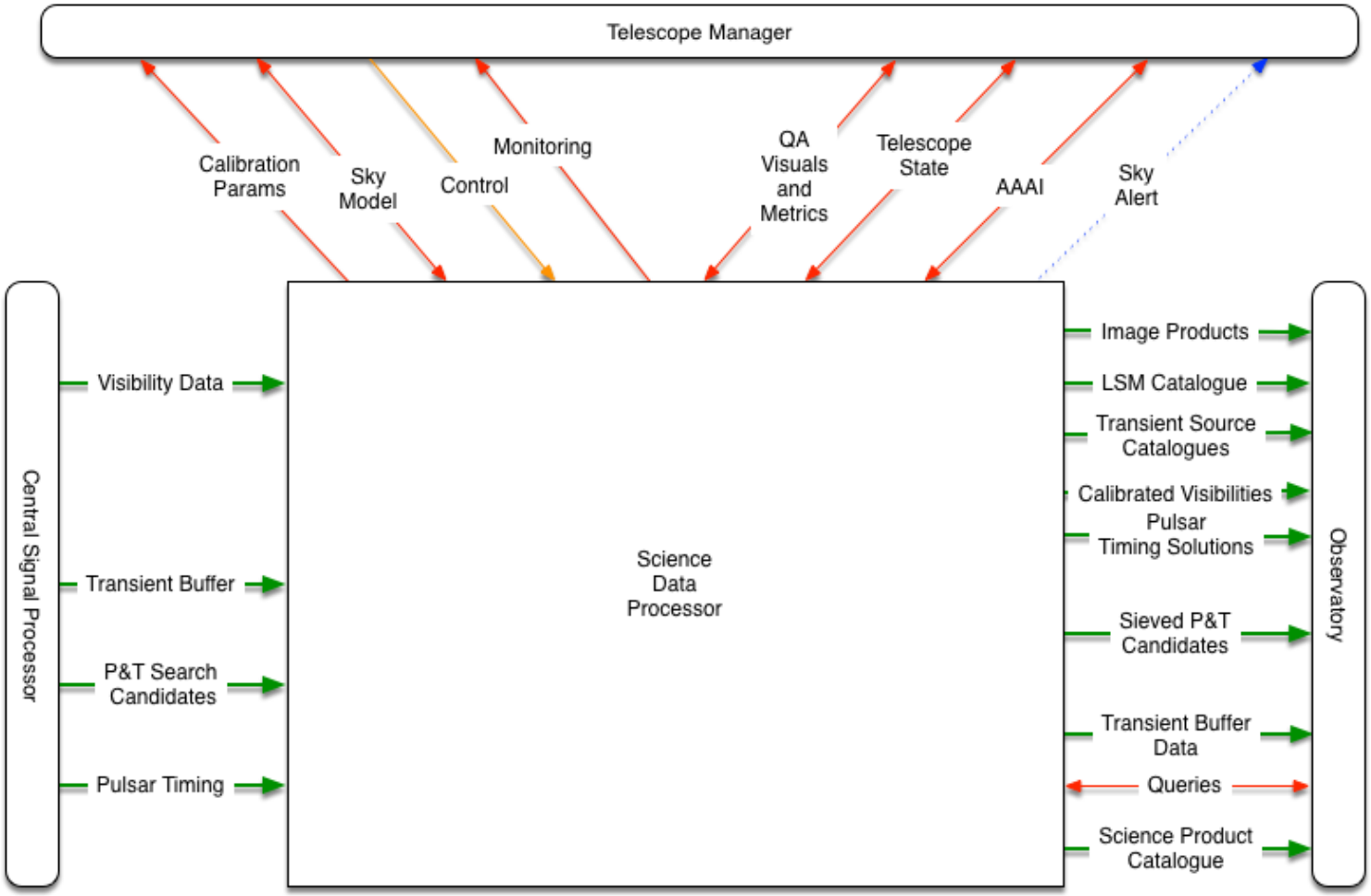
The slides presented draw directly on suggestions/material provided by:

- Paul Alexander
- Verity Allen
- Rosie Bolton
- Tim Cornwell
- Ferdl Graser
- Ben Mort
- Bojan Nikolic
- Stif Telfer
- Peter Wortmann

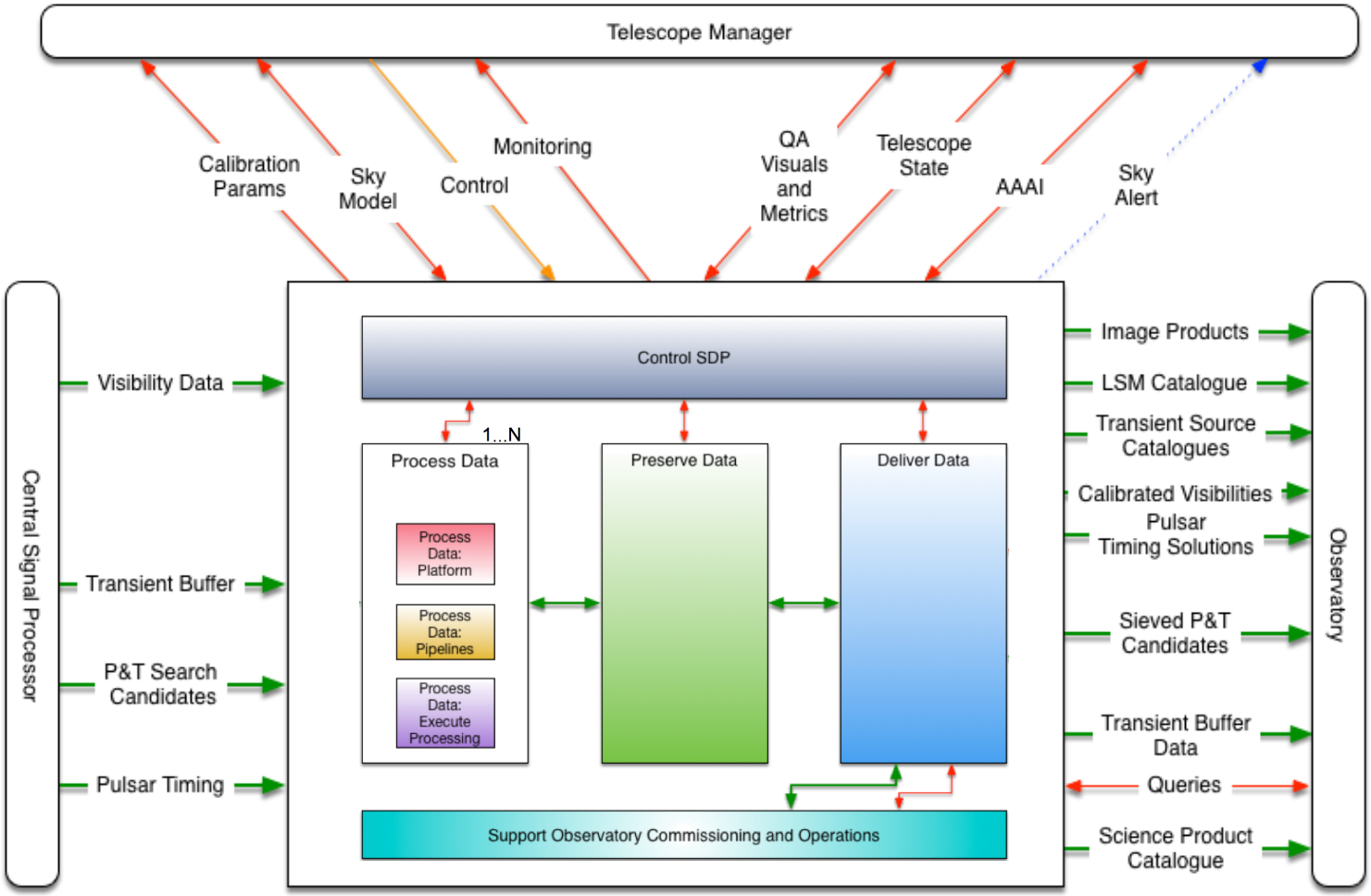
And indirectly work from across the whole of SDP.

Mistakes in translation and presentation are mine! Inevitably to fit the time available for this talk I can only present a subset of recent SDP activity. Many areas are Work In Progress.

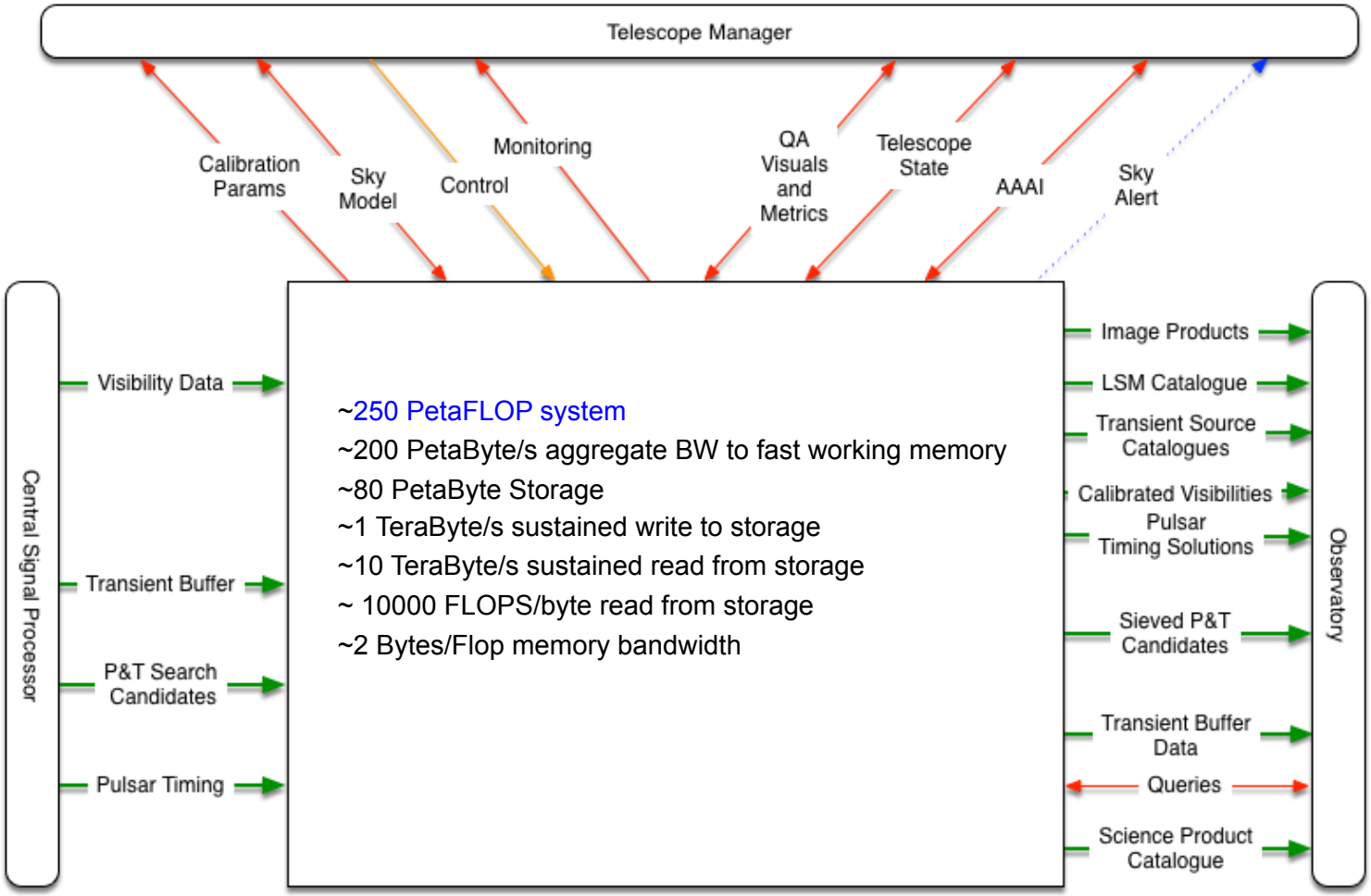
SDP Context 1



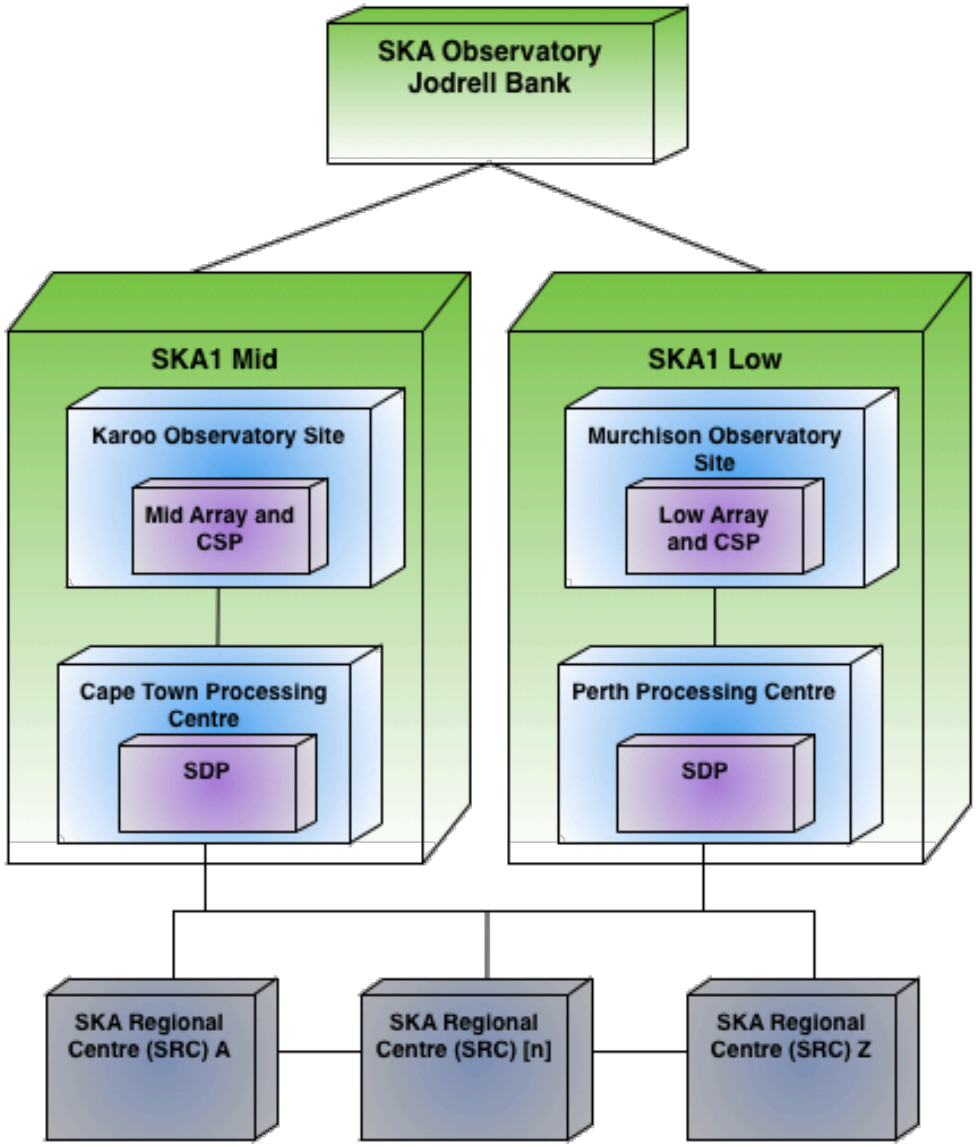
SDP Context 2



SDP Context 3



Reminder: One SDP Two Telescopes



	Ingest (GB/s)
SKA1_Low	500
SKA1_Mid	1000

In total need to deploy eventually a system which is close to 0.25 EFlop of processing

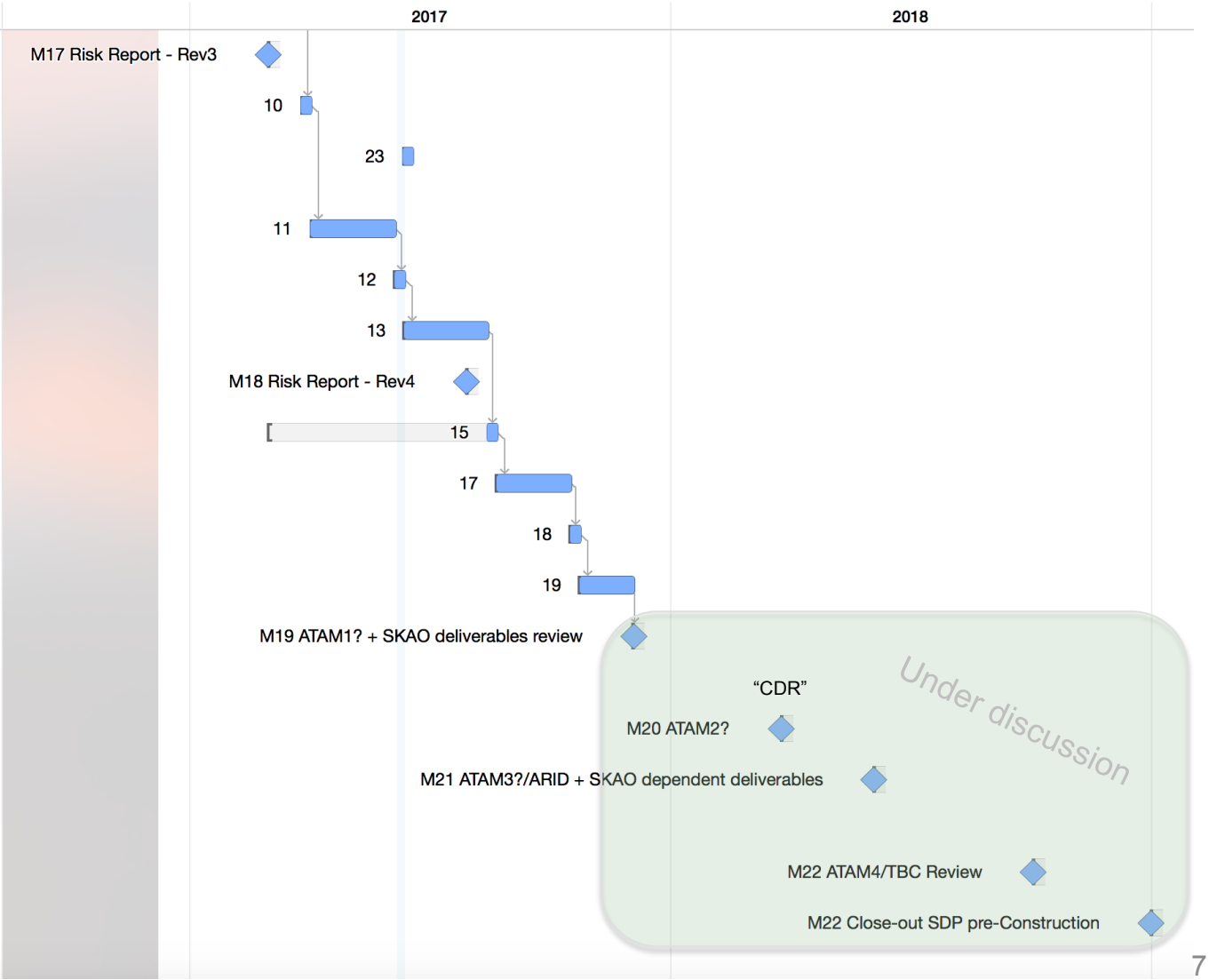
A tiered (regional) processing organisation will consume SDP outputs. Data products from SKA up to 1PB/day

Progress & planning

The understanding of how best to move forward is evolving

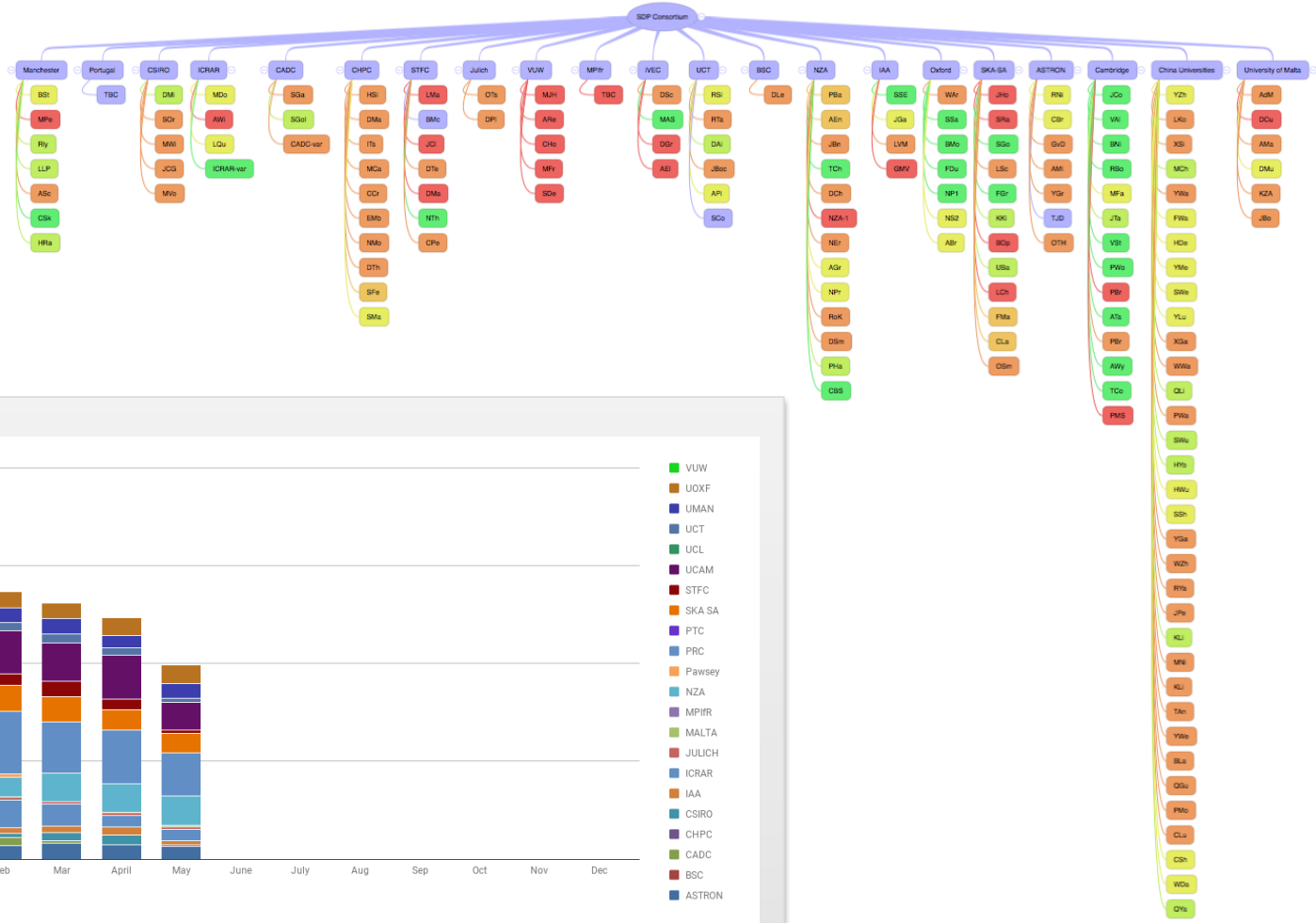
SCIENCE DATA PROCESSOR

	Title	Effort
4)	M17 Risk Report - Rev3	0h
5)	Sprint review & planning	1w
6)	SKA Engineering Meeting	1w
7)	Sprint 2	8w 4d 7h
8)	Sprint review & planning	4d 7h
9)	Sprint Jun-Aug	8w 4d 7h
10)	M18 Risk Report - Rev4	0h
11)	Sprint review & planning	4d 7h
12)	Sprint 4	7w 4d 7h
13)	Sprint review & planning	1w
14)	DMR Sprint	5w 4d
15)	M19 ATAM1? + SKAO deliverables review	0h
16)	M20 ATAM2?	0h
17)	M21 ATAM3?/ ARID + SKAO dependent deliverables	0h
18)	M22 ATAM4/ TBC Review	0h
19)	M22 Close-out SDP pre-Construction	0h



The SDP Consortium – Resourcing

Steady improvement in utilization requires continual monitoring



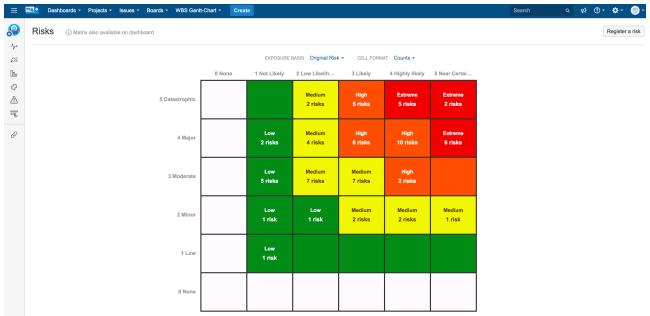
Reported effort

Sprint planning every 8 weeks

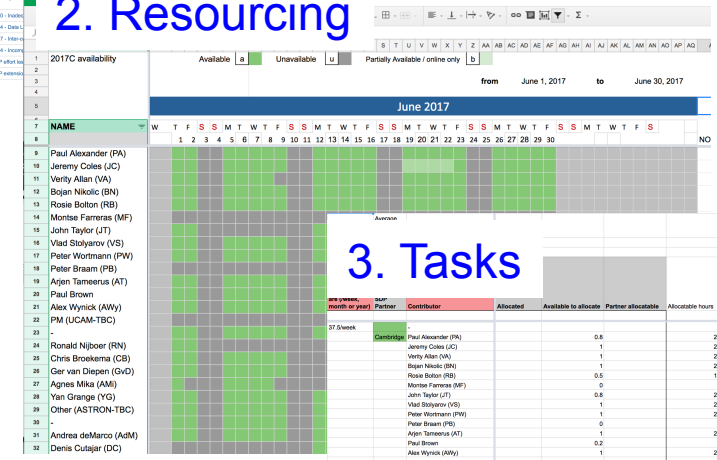
Evolving the project processes

SKA SQUARE KILOMETRE ARRAY — SDP — SCIENCE DATA PROCESSOR

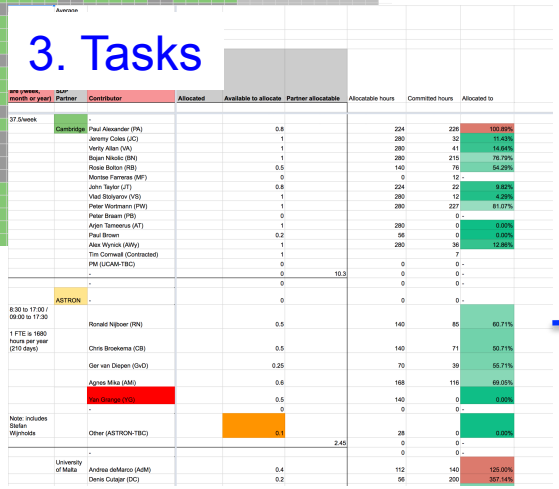
1. Risks

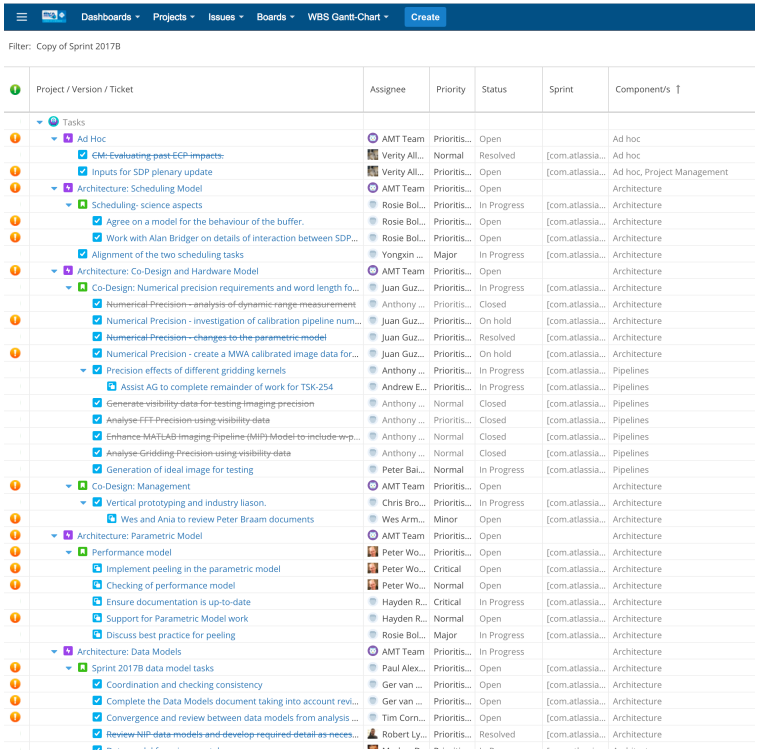


2. Resourcing



3. Tasks





Work Breakdown Structure
JIRA allocations

SDP Risks
Regular review

SDP to LTS interface

Pipelines to execution framework interface

Calibration strategy

Data model maturity

.....

Key	Summary	Status	Original Exposure	Residual Exposure	Treatment
SDPRISK-315	0000 - Inadequate System Performance due to the interface between the Data Processor and Local Teles	TREATED	EXTREME	MEDIUM	Mitigate
SDPRISK-323	0094 - Data Lifecycle Manager design not well defined	ANALYZED	EXTREME		
SDPRISK-340	0107 - Inter-compute island communications insufficient for certain workflows				itigate
SDPRISK-343	0104 - Incomplete Interface Description between pipelines components & Execution Framework				
SDPRISK-354	SDP effort less than pledged - Impact: late deliverables	TREATED	EXTREME	HIGH	Mitigate
SDPRISK-358	SDP extension funding reduced - Impact: late or reduced quality deliverables	ANALYZED	EXTREME		
SDPRISK-359	0095 Processing/Calibration Strategy is not well defined	ANALYZED	EXTREME		
SDPRISK-360	Parametric Model underestimates the computational requirements	ANALYZED	EXTREME		
SDPRISK-363	buffer software does not meet performance requirements	ANALYZED	EXTREME		
SDPRISK-367	0093 - Poor scalability due to architectural approach to the Local Sky Model and Local Telescope State	ANALYZED	EXTREME		
SDPRISK-374	0046 - Expertise lost between Pre-Con & Construction - schedule slips	ANALYZED	EXTREME		
SDPRISK-388	0074 - SDP data model is immature	TREATED	EXTREME	MEDIUM	Mitigate
SDPRISK-390	0065 - Execution Framework is immature at production (low TRL)	TREATED	EXTREME	MEDIUM	Mitigate
SDPRISK-311	0091 - Cost overrun due to complexity of pipeline components	TREATED	HIGH		Mitigate
SDPRISK-313	0089 - Uncertainty in hardware cost project for the SDP scalable unit	ANALYZED	HIGH		
SDPRISK-319	0032 - SDP does not meet SKA Availability Requirement	TREATED	HIGH	LOW	Mitigate
SDPRISK-320	0108 - Fast imaging pipeline fails to meet performance requirements	TREATED	HIGH	HIGH	Mitigate
SDPRISK-325	SDP lose effort - Impact: late or reduced quality CDR deliverables	ANALYZED	HIGH		
SDPRISK-329	0122 - Available Bandwidth insufficient at the two SDP sites.	TREATED	HIGH	LOW	Avoid
SDPRISK-333	0016 - Level 3 ICDs are poorly defined	TREATED	HIGH	MEDIUM	Mitigate
SDPRISK-341	0109 - Pulsar Timing Pipeline design has no Failure Modes	TREATED	HIGH	HIGH	Mitigate
SDPRISK-344	0023 - Data rate between nodes is underestimated	ANALYZED	HIGH		

SDP Parametric Model

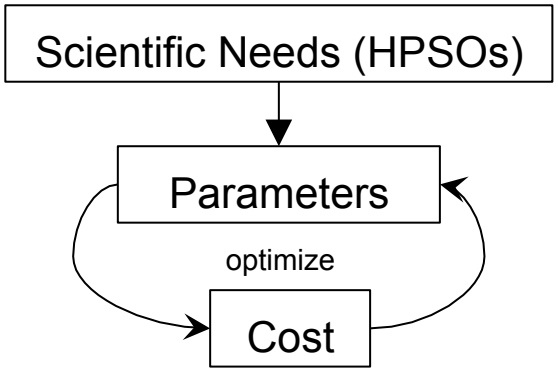
Improving our understanding of the the drivers of SDP cost

Optimises parameters for cost. Covers:

- All telescopes, bands and HPSOs
- Predict, calibration, imaging and deconvolution
- Self-calibration (RCAL, ICAL)
- Data product preparation (DPrepX)

Output for hardware costing:

- Compute rates (floating ops)
- Data rates (ingest, internal)
- Storage requirements (buffer)

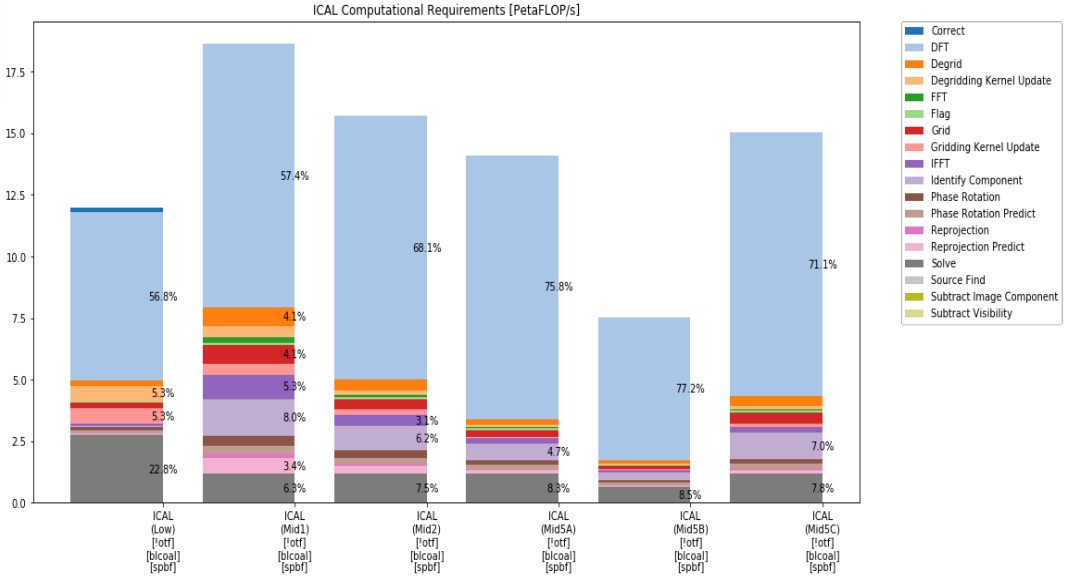


Highlights:

- Aw-imaging with snapshots and baseline-dependent averaging in time and frequency
- Facetting throughout
- Hybrid predict based on DFT + FFT;
- Stefcal calibration
- Multiscale Multifrequency Synthesis for deconvolution / major loops.

To-do:

- more advanced calibration algorithms (sagecal, peeling, facet calibration)



SDP Processing Components

Understanding bottlenecks and opportunities

High-level questions:

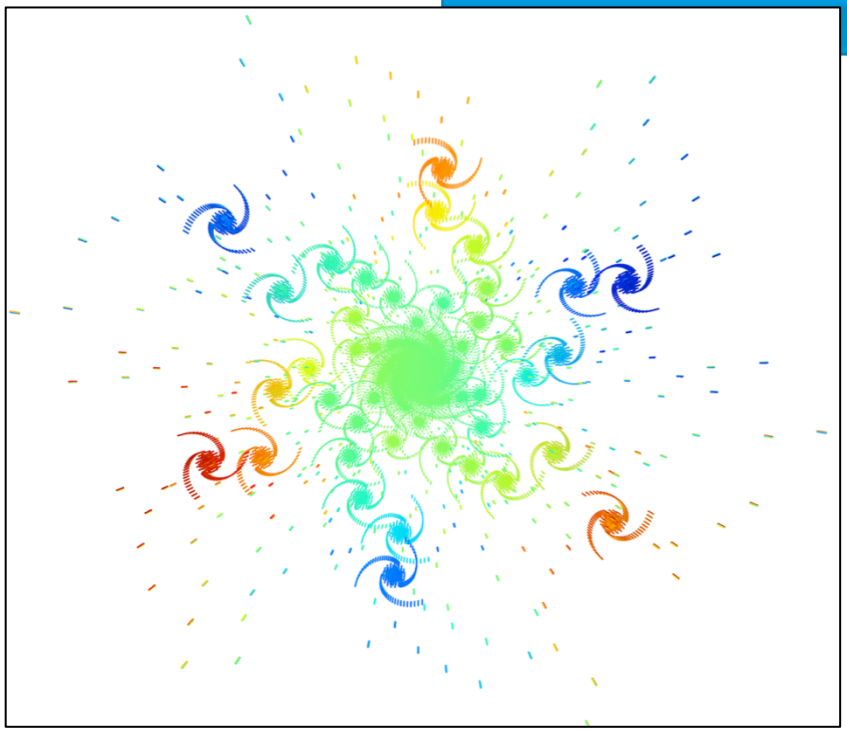
- How many processing units?
- Can memory keep up?
- Do we have right operation mix?

Expensive uncertainty:

- Assumption is 10%!
- Need to break with tradition
- Averaging decides the actual amounts of visibility data with which we are dealing
- Algorithmic approach must vary

Working (with industry) to gain knowledge:

- Focus on small benchmarks for predicted bottlenecks, with data sets generated according to PM predictions
- Modeling recommends “exotic” configurations, needs tailored tests
- Gridding, DFT likely with potential



45s snapshot grid coverage. Colour corresponds to complexity of w-kernels. There are 3 main regions to consider.

Very data intensive pipeline. Recent work was understanding the issues.

Execution frameworks

Parametric model can be used to formulate benchmarks for data rates

SCIENCE DATA PROCESSOR

Pipeline execution is high risk:

- High data & task rates, time limited
- Need flexibility:
  - Parameter changes between pipelines
  - Hardware changes (new architectures)
  - Software changes (new algorithms)
  - Real world (reliability, usability)
- But ideally not reinvent the wheel!

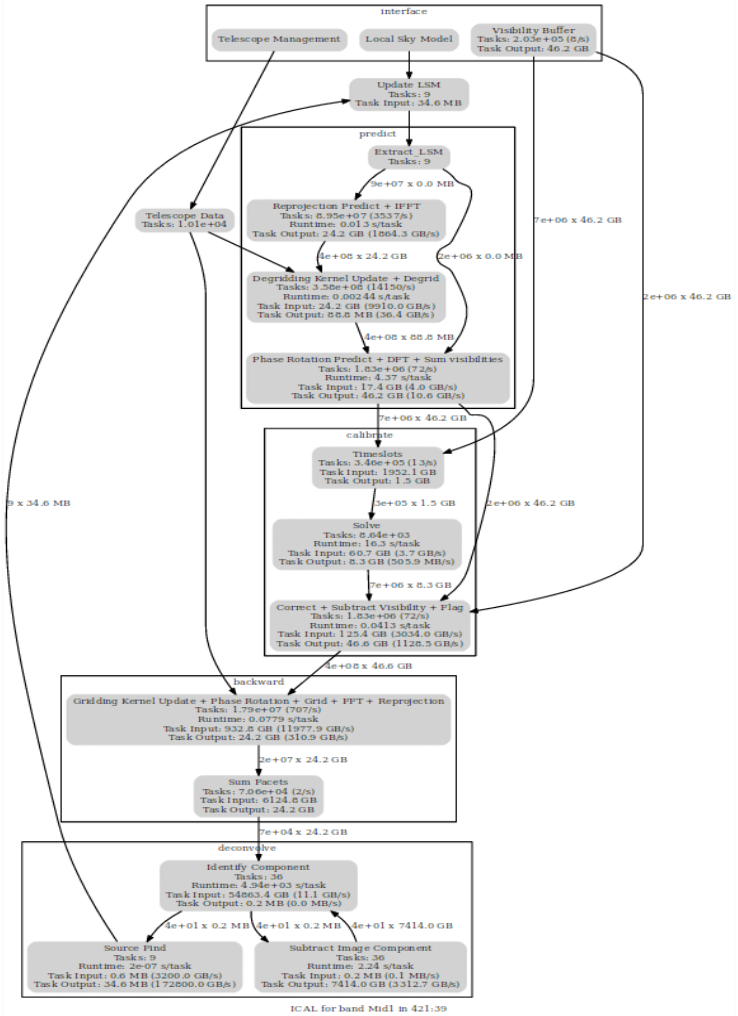
"Big data" still struggles with SKA scale.

- Have to keep options open (e.g. RDMA)
- Benchmark predicted data flows
- Study issues and trends
- Options not well aligned to our static pipeline configurations & raw throughput.

Some frameworks we have explored with a challenging test pipeline

Swift/T, Legion/Regent, DaLiuGE, Apache Spark (+ Alluxio), StarPU, COMPSs, Dask

SDP architecture must be such that execution framework question can be reviewed later.



Example: ICAL for Mid. Some areas need further development

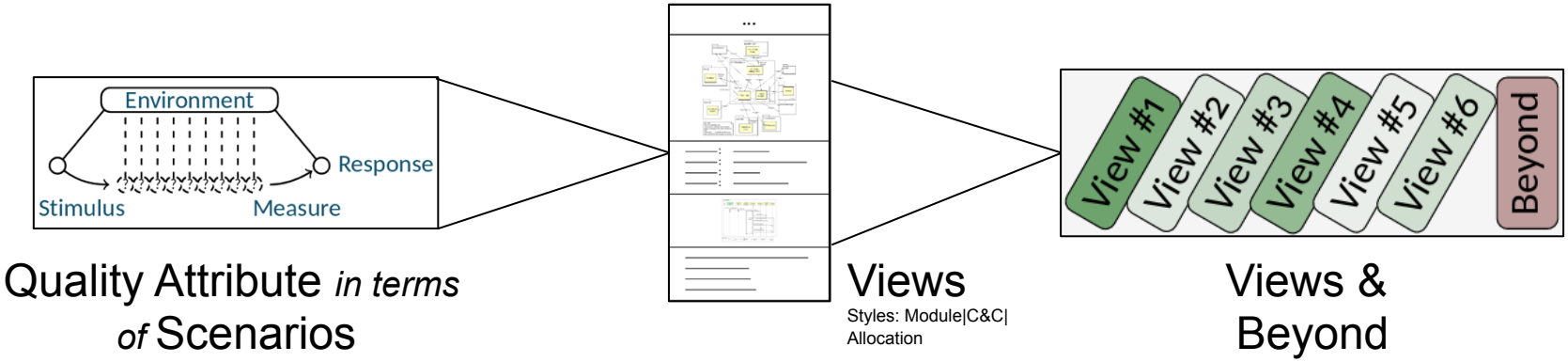
Software Engineering Institute (SEI) approach

Light on formalism but heavy on constructive methods

ECP-170001: "*This document should include [⋯] details of Quality Attribute Scenarios, views, prototypes, interfaces and use cases.*"

1. Formulate quality attribute scenarios (concrete & measurable)
2. Document solution in views (tailored to audience, test & iterate)
3. Collect and tie together into software architecture documentation.

Structured progress despite large design space and many constraints



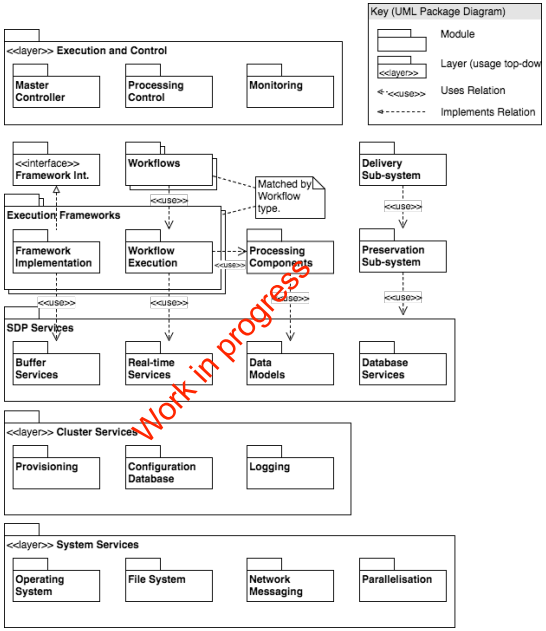
Recent document: Architecture-Centric Development Plan for the Square Kilometre Array Science Data Processor

SEI approach - view examples

Reworking the SDP architecture and documentation

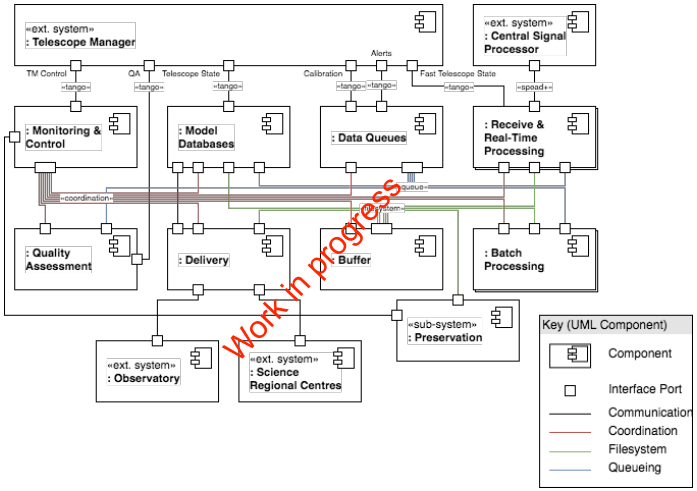
SCIENCE DATA PROCESSOR

Module views



- Focus on static structure
- Highlights coupling (e.g. execution engine)
- Reflects work packages

Component & Connector views



- Focus on runtime structure
- Highlights communication (e.g. ingest, buffer access)
- Suggests hardware requirements

Data models

Understanding expected data volumes and access patterns

- The SDP data processing pipelines will use and produce various data products.
- We have been analysing their data models as well as the expected data volumes and access patterns.
- As data processing is distributed and parallelised this needs to be supported by the data models and access to the data.
- Visibility and image data and calibration solutions will form the bulk of the data and can be distributed in a natural way.
- Other data (such as LTM and LSM) will be relatively small and kept by LMC.
- Note they can potentially be accessed simultaneously by possibly hundreds or thousands of distributed pipeline components in a bursty manner.

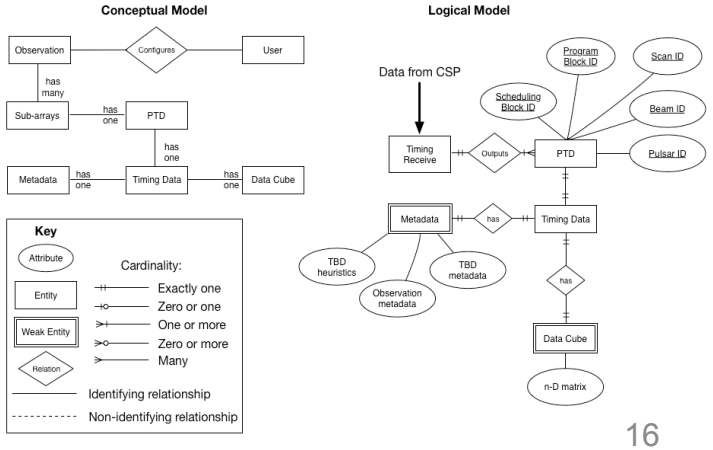
Inputs:
Visibilities
UVW coordinates
Visibility weights
Flags

Intermediate Data Products:
Multiscale clean scale images
Multiscale clean residual images
W-kernels
A-kernels
Anti-aliasing kernel
Oversampled kernel
Imaging weights

Outputs:
Dirty image
PSF
Residual image
Clean image
Clean components
Processing log

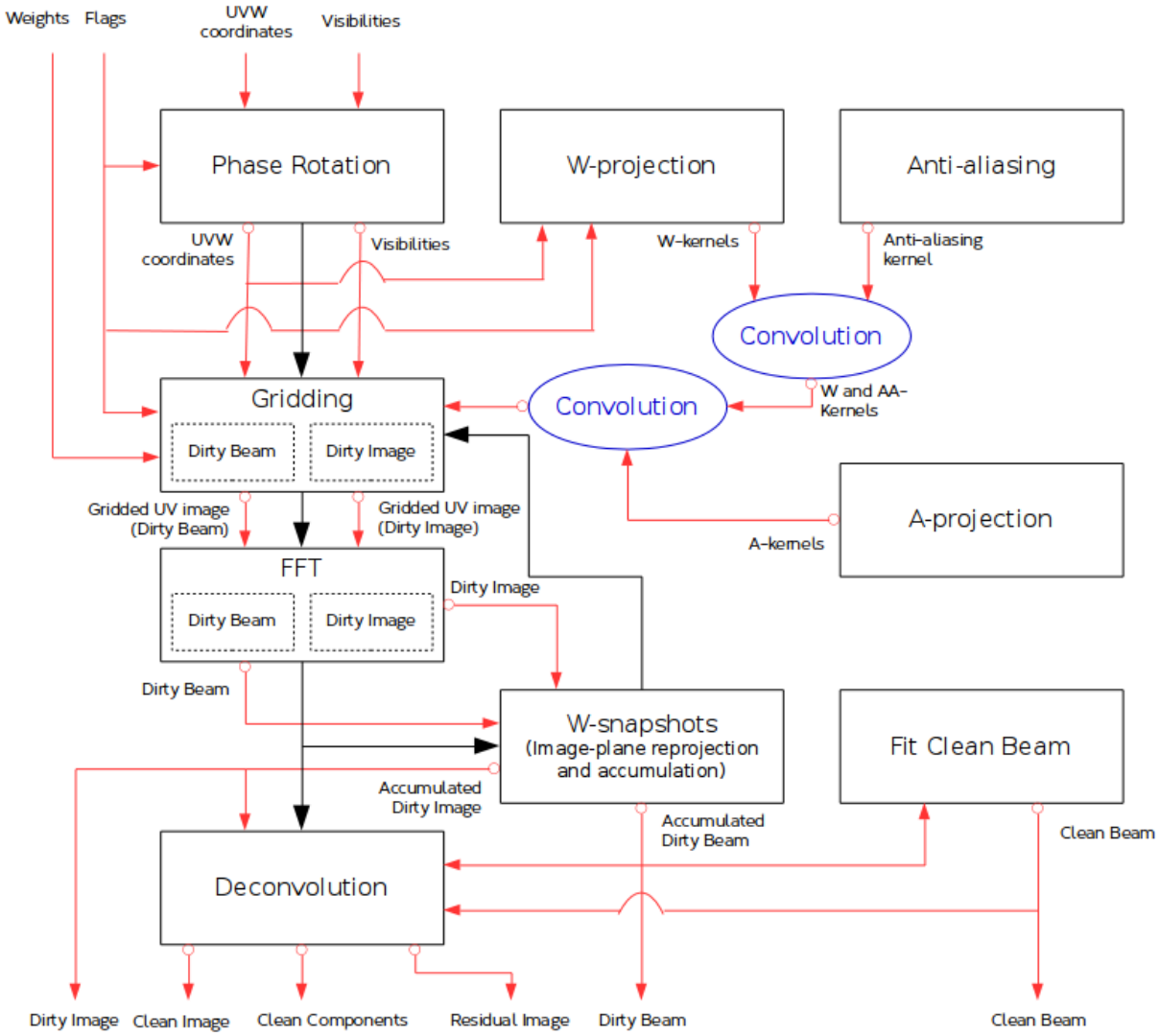
Components:
Phase rotation
Direction-dependent corrections (A-projection)
W-projection
W-snapshots
Anti-aliasing
Gridding
FFT
De-gridding
Deconvolution
...

NIP example



Illustrative flow chart

Describing the data relationships between components

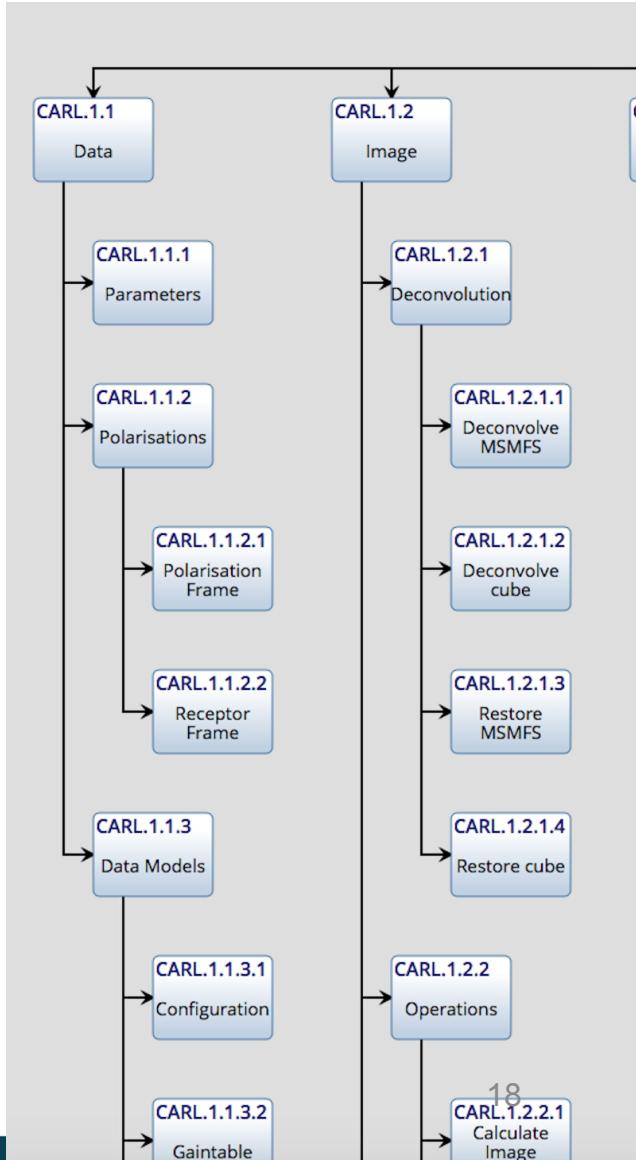


DRAFT

Algorithm Reference Library (ARL)

A Python prototype of the algorithms

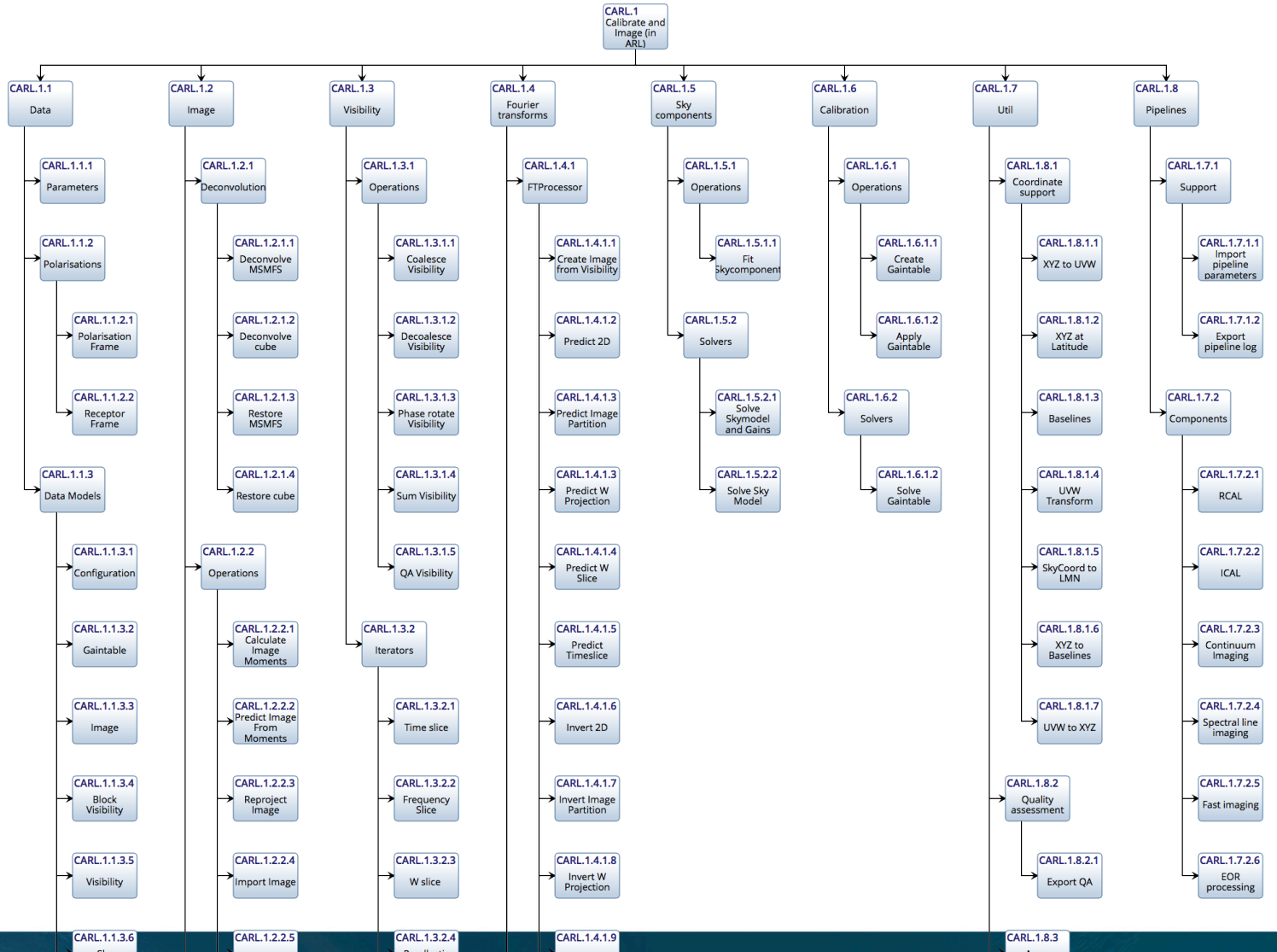
- All major calibration and imaging algorithms expressed in Python unencumbered by practicalities of I/O, memory management, optimisation
- Estabishes reference e.g. for later implementation by non-experts
- Prototypes of functional components (referentially transparent)
- Synthesis components: Fourier transforms (predict, invert) using 2D, wprojection, faceting, wslicing, snapshots
- Model for LOW sky(S3) and LOW station beam (OSKAR)
- Testing is memory-limited: largest image made so far is 25K by 25K pixels
- Largely complete
- 97% coverage test suite



Algorithm Reference Library

Synthesis framework allows experimenting with components

SKA SQUARE KILOMETRE ARRAY
SDP
SCIENCE DATA PROCESSOR



Graph processing using ARL and Dask

Demo in data flow environment. Platform for experiments.

- ARL forms a suitable basis of exploration of graph processing for SDP
- Graphs built and executed with Dask python package

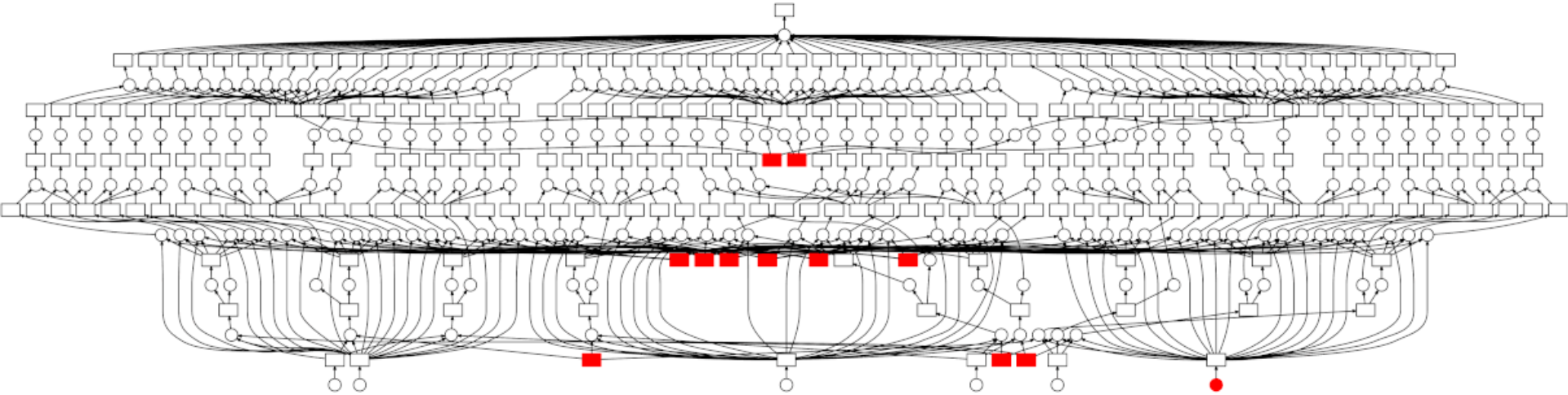
http://dask.pydata.org

- Express pipelines as graphs
- ICAL - the core selfcalibration/continuum imaging pipeline - done
- Evaluation of performance, memory overhead, locality control, schedulers
- Easy scaling from laptop (4 cores) to Darwin cluster

Animation of graph processing

To help convey the approach....

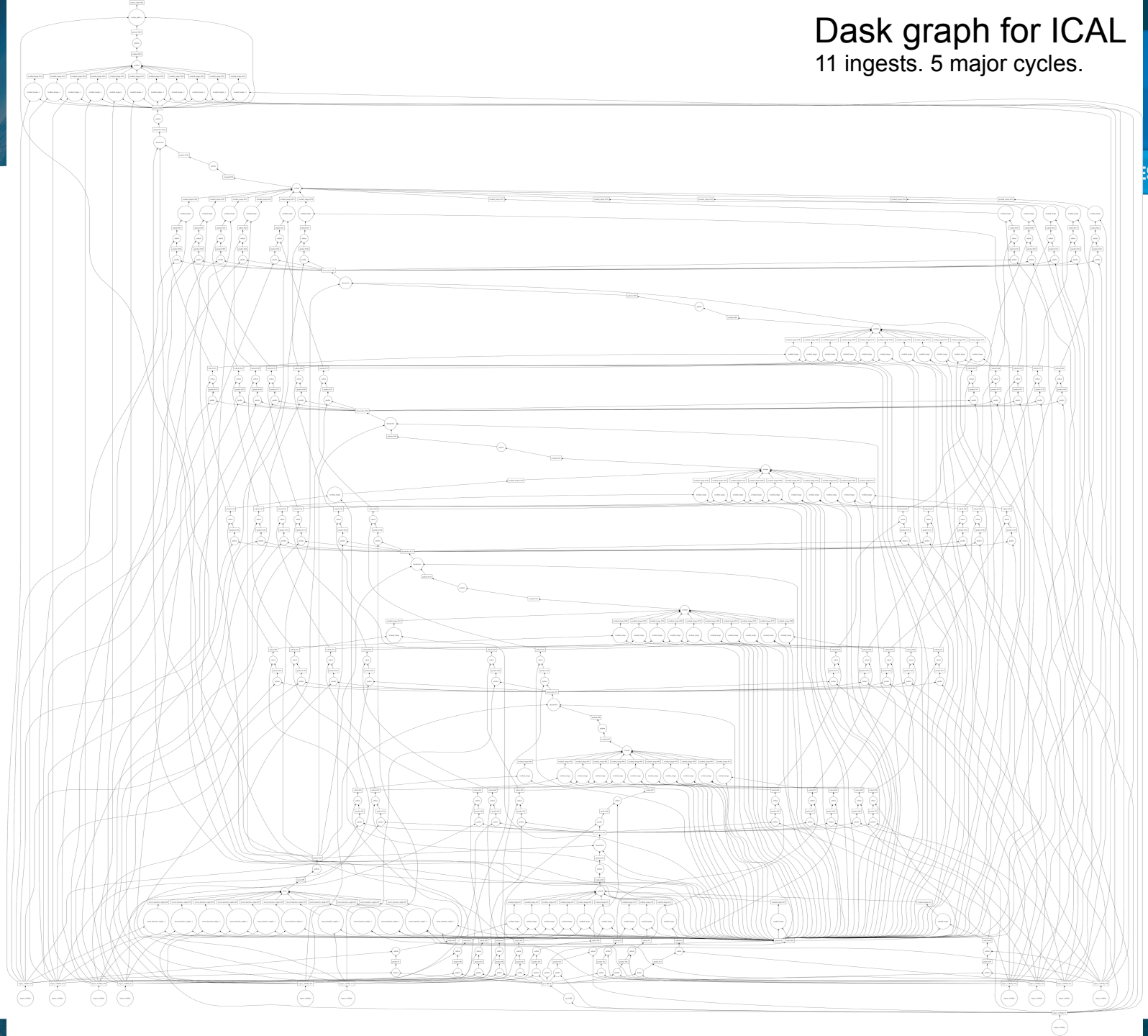
- Not an SDP example (yet) but one from Dask tutorial.



- Multiple inputs (bottom) flow to single output (top)
- Approach goes for depth first processing (vs MPI breadth first)

Dask graph for ICAL

11 ingests. 5 major cycles.



Stressing the architecture

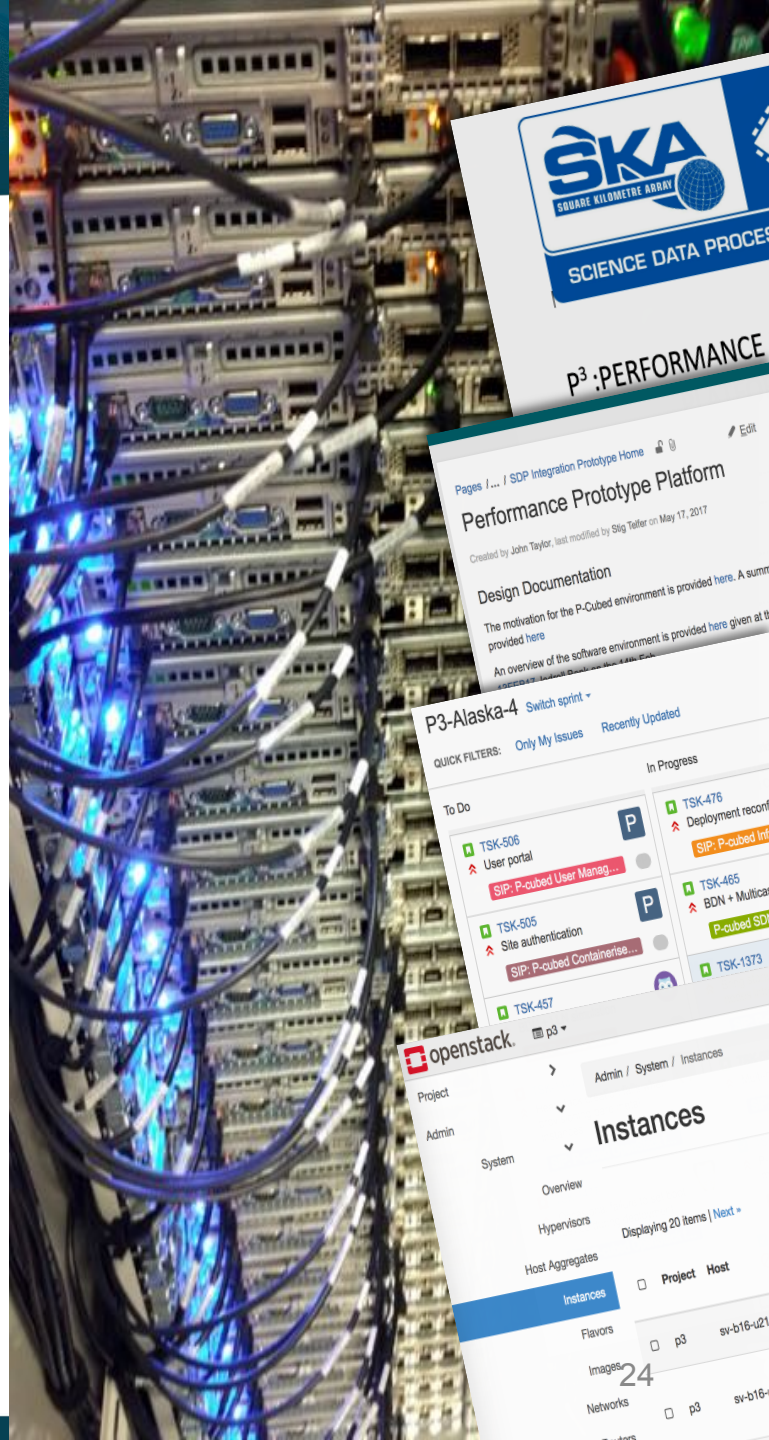
Exploring the limits of the island based nature of data flow.

- Can the SDP architecture support all the algorithms we expect to need?
- SDP models processing as parallel streams of data flow undergoing processing with limited information exchanged between streams
- Algorithms that stress this model: global calibration, MSMFS, SageCAL-CO
- Can restate this question in terms of the telescope: are there physical effects that are coupled across data streams? How important are they?
- Preliminary answers: yes, there are coupled effects, and yet, they have the potential to limit the science
- A possible and inexpensive modification is to use an all-to-all non-blocking switch. The software implications would need examining.
- Still a work in progress but expect to settle very soon

ALASKA – Advanced SDP Infrastructure

A single tenancy platform to support our prototyping

- Designed to support diversity and flexibility
  - Heterogeneous hardware technologies
  - Advanced OpenStack control plane
  - (but without sacrificing performance)
- Designed for prototyping and comparing
  - Software defined infrastructure
  - Rapid deployment of execution frameworks
  - Support for profiling, monitoring and analysis



Supporting Performance Prototyping

For studying all levels of the SDP software environment

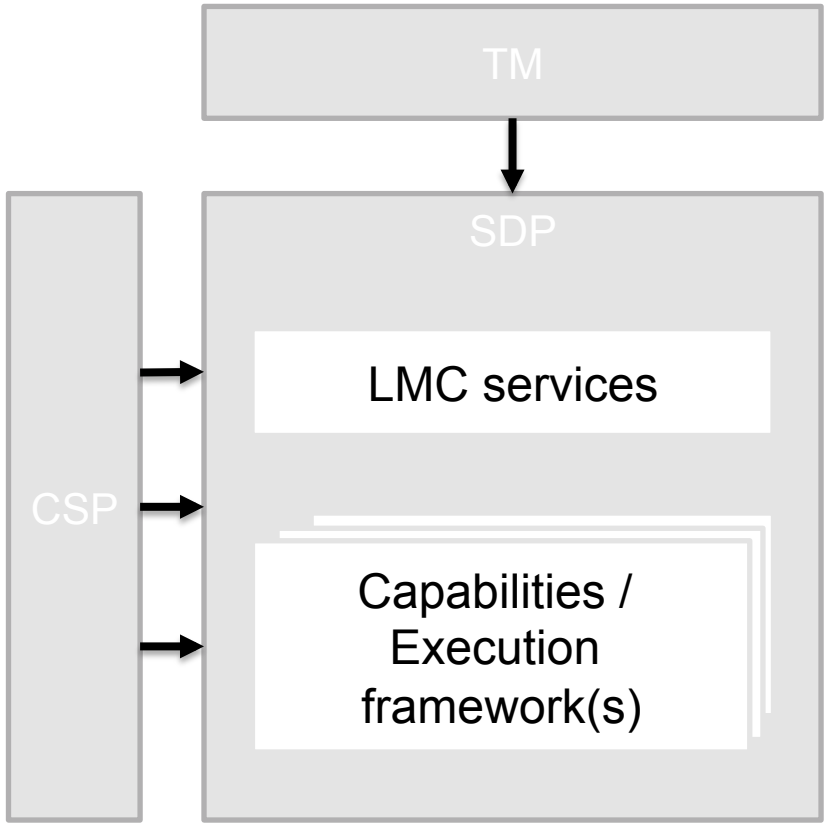
- Providing Execution Frameworks (as a service)
  - SLURM and MPI
  - Docker Swarm, Kubernetes
  - Mesos
  - Spark
- Models for Stimulus and Simulation
  - Bulk Data Network
  - Local Monitoring & Control
- Providing Tools for Performance Analysis

Alaska's OpenStack provides logging and monitoring tools for execution frameworks. This enables application performance telemetry to be seen alongside telemetry from infrastructure components - creating a holistic picture.

SDP System Integration Prototype (SIP) - context

Demonstrating an end-to-end SDP system prototype

- Prototype of all major external and internal interfaces
- Verification and testing of SDP architecture
- Focused testing and analysis of technology choices
- Link to hardware prototyping (AlaSKA-P3)
- Link to vertical / execution framework prototyping



SDP System Integration Prototype (SIP) - activities

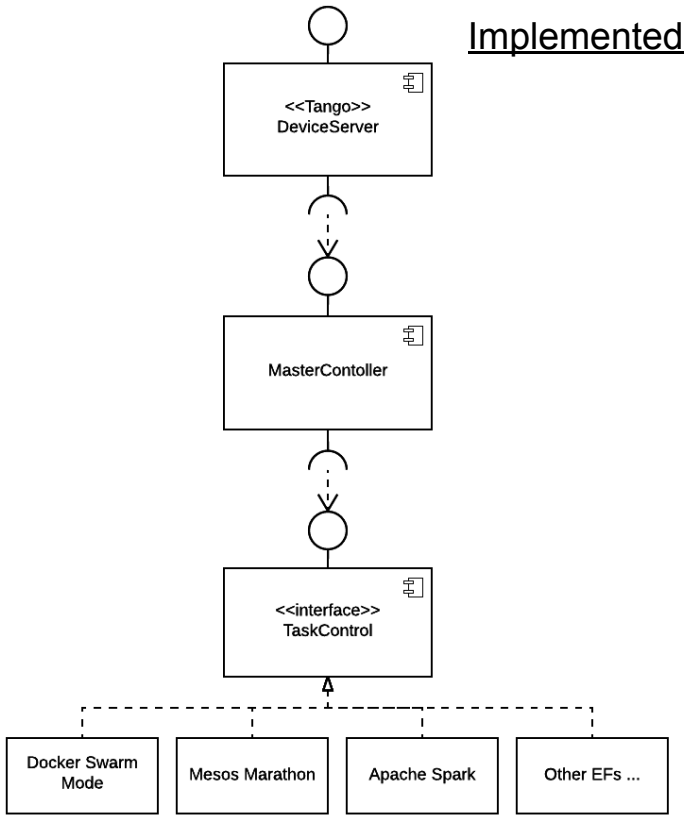
Exploring high-risk areas

Current activities

- TANGO interfaces to TM
- CSP interfaces (SPEAD, Pulsar search)
- Master Controller interface to tasks and execution frameworks
- Prototyping a number of test capabilities (pipelines)
- LMC services such as LTS using distributed Redis
- Deployment of SIP code onto AlaSKA-P3

Next Steps

- Prototyping of the SDP buffer
- Integration of LMC services (eg. LTS, Sky model, QA) with execution frameworks



Example outcome: resilience in running LMC services. By moving towards a micro-services like architecture using container orchestration the risk of single key service failure for our control interfaces is largely mitigated

SDP System sizing methodology
Exploring cost saving options

HPSO	Total (PFLOPS)	Hours of telescope time	Fraction of time
U.HPSO-4a Pulsar Search MID SPF1	~0	800	0.01
U.HPSO-4b Pulsar Search MID SPF2	~0	2400	0.04
U.HPSO-5a Pulsar Timing MID SPF2	~0	1600	0.02
U.HPSO-5b Pulsar Timing MID SPF3	~0	1600	0.02
U.HPSO-13 Hi Kinematics and Morphology	25.6	5000	0.07
U.HPSO-14 Hi MID	32.7	2000	0.03
U.HPSO-15 Studies of the ISM in our Galaxy	26.2	12600	0.19
U.HPSO-18 Transients MID	~0	10000	0.15
U.HPSO-22 Cradle of Life MID Band 5	25.4	6000	0.09
U.HPSO-27 All Sky Magnetism	26.3	10000	0.15
U.HPSO-37a Continuum Survey MID band 2	28.1	2000	0.03
U.HPSO-37b Continuum Survey MID band 2 (deep)	28.1	2000	0.03
U.HPSO-37c Continuum survey, band 2 wide	28.1	10000	0.15
U.HPSO-38a Continuum Survey MID band 5	26.1	1000	0.01
U.HPSO-38b Continuum Survey MID band 5	26.1	1000	0.01
Weighted average FLOPS value for MID HPSOs			20 PFLOPS
Approximate AVERAGE Apparent power requirement[2]			~2.7 MVA

Calculate required number of operations for each experiment (total).

Use fractions of time spent doing each experiment, calculate average SDP compute load

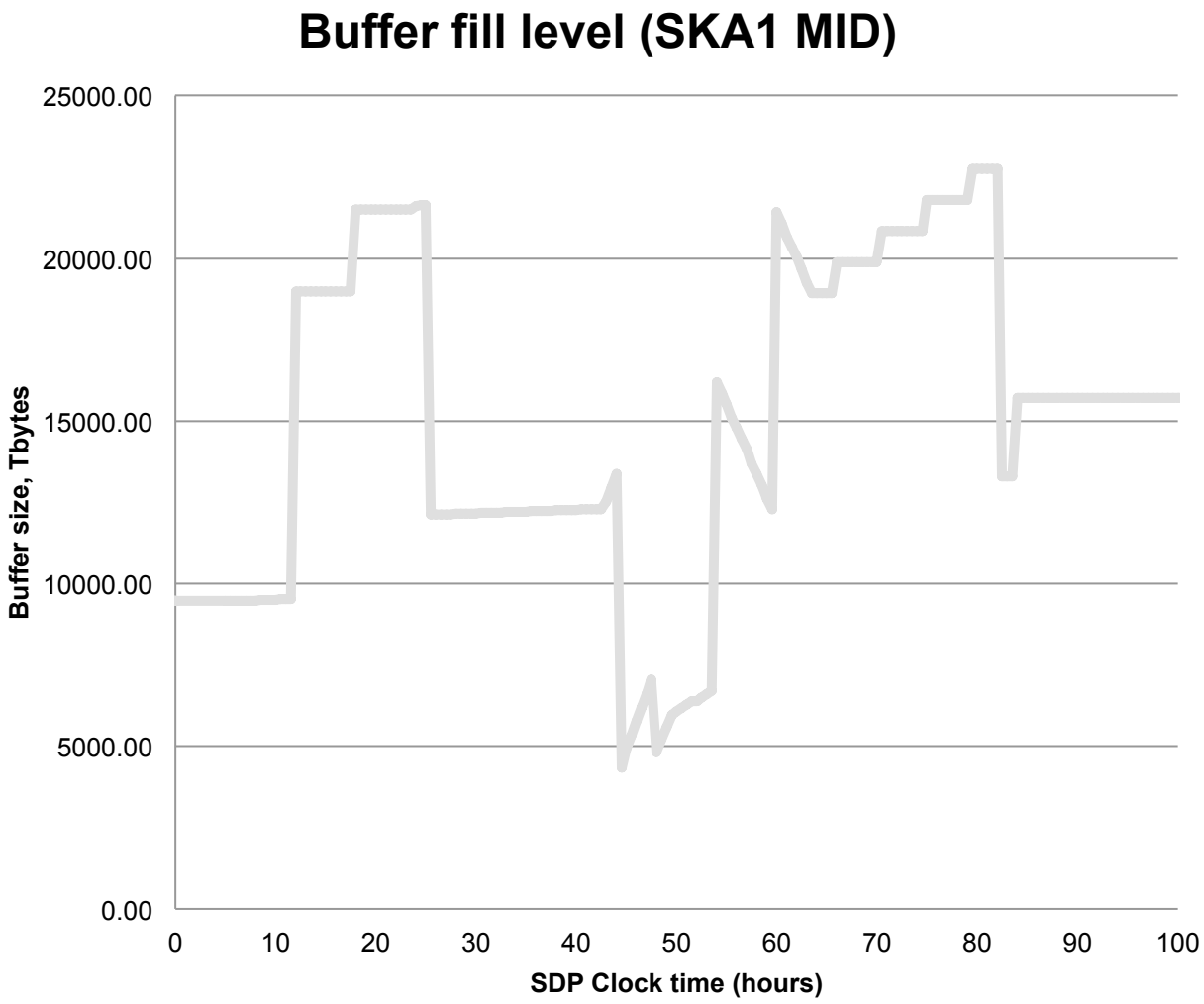
Average FLOPS values 3.5x lower for MID than maximal case.

Relax latency requirement (buffer) and save both capital cost and power cost

Overall designing to a 250 PFlop peak system (average efficiency ~10% driven by likely memory bandwidth)
There are lots of assumptions!

Buffer sizing

Has provided one saving by allowing a looser coupling



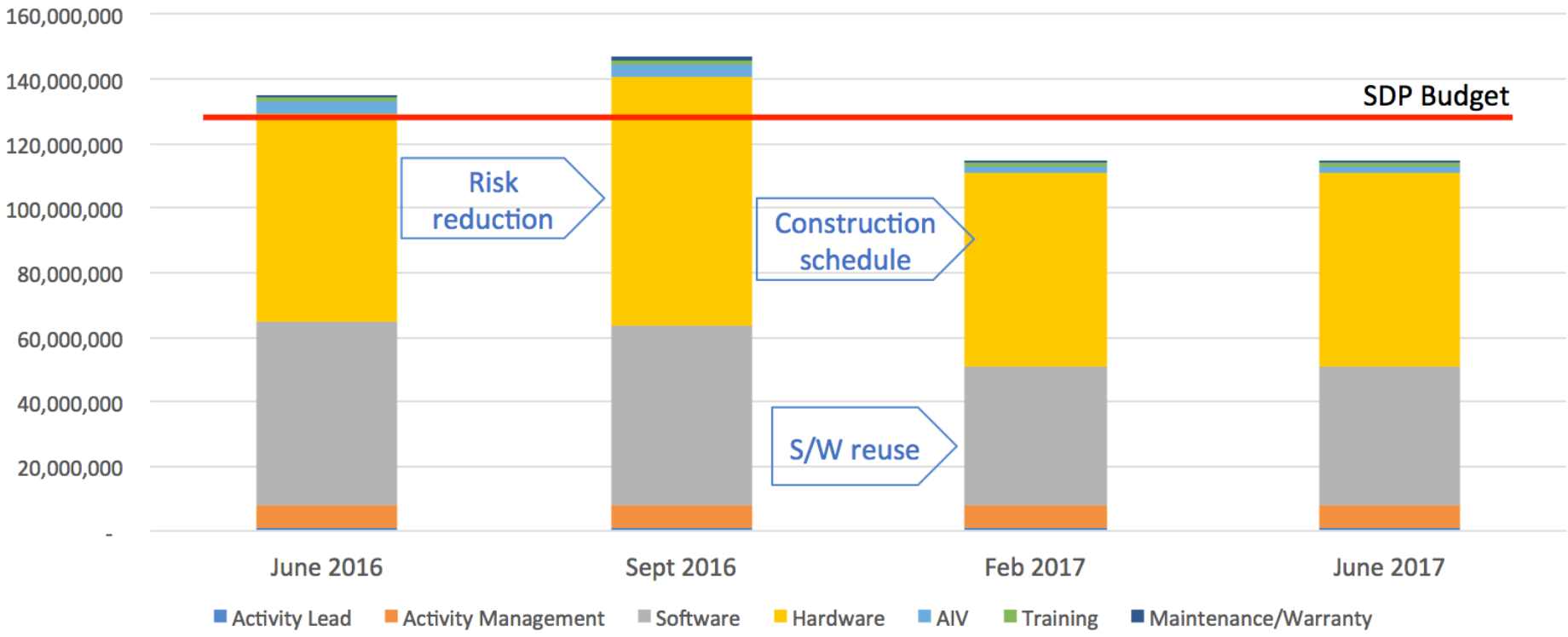
We use estimate the amount of data in the buffer at any time.

Note that one 12 hour SB for MID is about 20 Pbytes

We size the buffer using the peak value from here, with a 20% overhead, plus an additional full 12 hour imaging SB.

SDP Cost Estimate over time

SDP has a Cost Resolution Team exploring cost reductions



- SDP is now costed under the element budget allocation
- Aggressive software reuse

SDP Cost Estimate

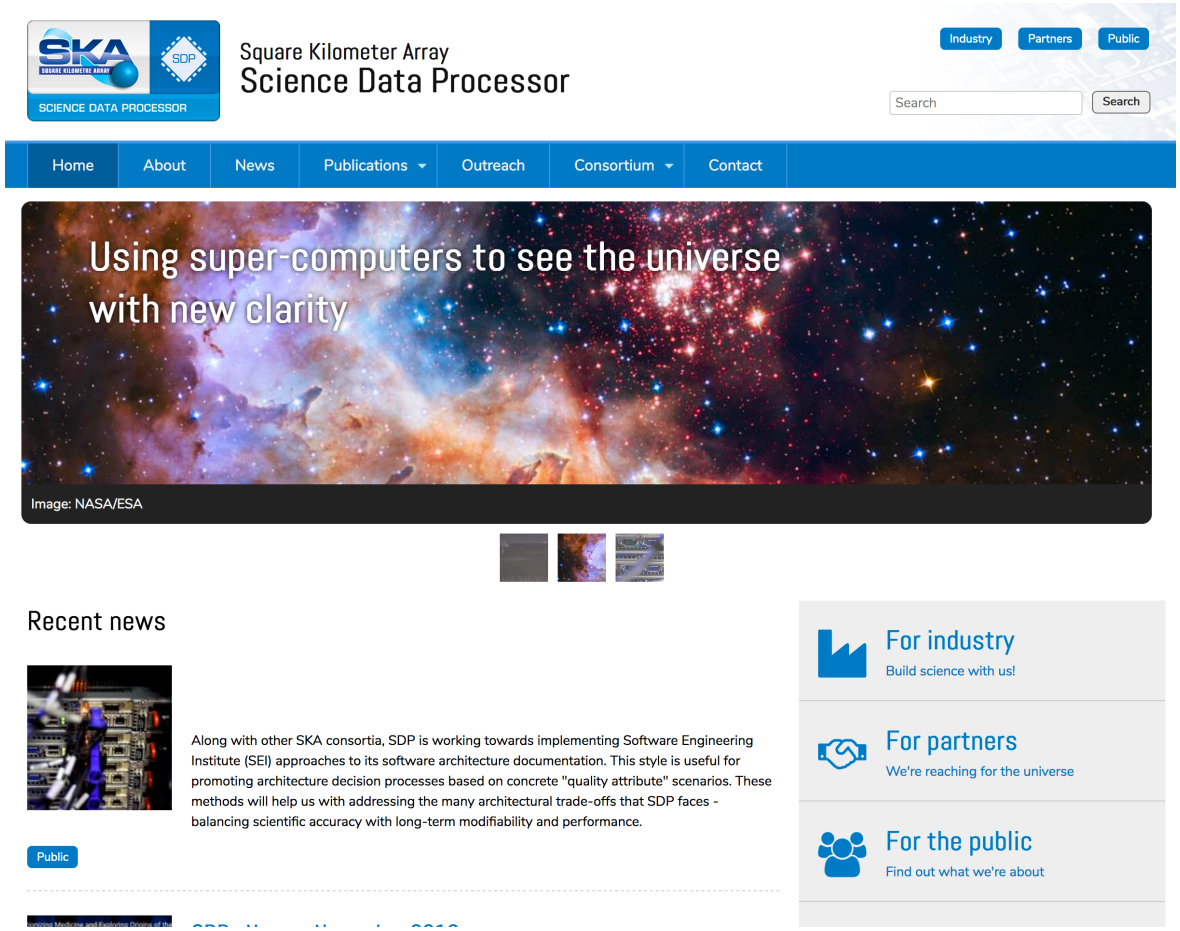
- June 2017 cost submission
  - Review of software maintenance cost (due to significant s/w reuse)
    - No change to software maintenance cost
  - Update of OPEX estimate for phased deployment scenarios
    - Submitted as part of CCP
  - Performance cost of peeling, MSMFS, etc.
    - Good progress but work is still ongoing.
    - Potential impact (risk) limited to low latency network and therefore < € 2M
  - Cost estimate same as Feb 2017.
- Other considerations:
  - Work ongoing for 2nd Tier KSPs
  - Phased deployment of hardware will give additional savings for TOC for 5-10yr period (in CCP considerations).
  - Looking at numerical precision needs and further potential cost savings

- NOTE: The SDP hardware costed concept is not a down-selection. It is a reasoned choice to provide a basis for deriving cost estimates.

Summary

- Context
- Schedule
- Sprint Planning
- Risks
- Parametric Model
- Processing components
- Execution Framework
- SEI Approach
- Data Models
- Algorithm Reference
- Completeness
- Prototyping Platform
- Integration Prototyping
- Buffer Model
- Cost Projections

http://ska-sdp.org



ADDITIONAL SLIDES

High Priority Science Objectives

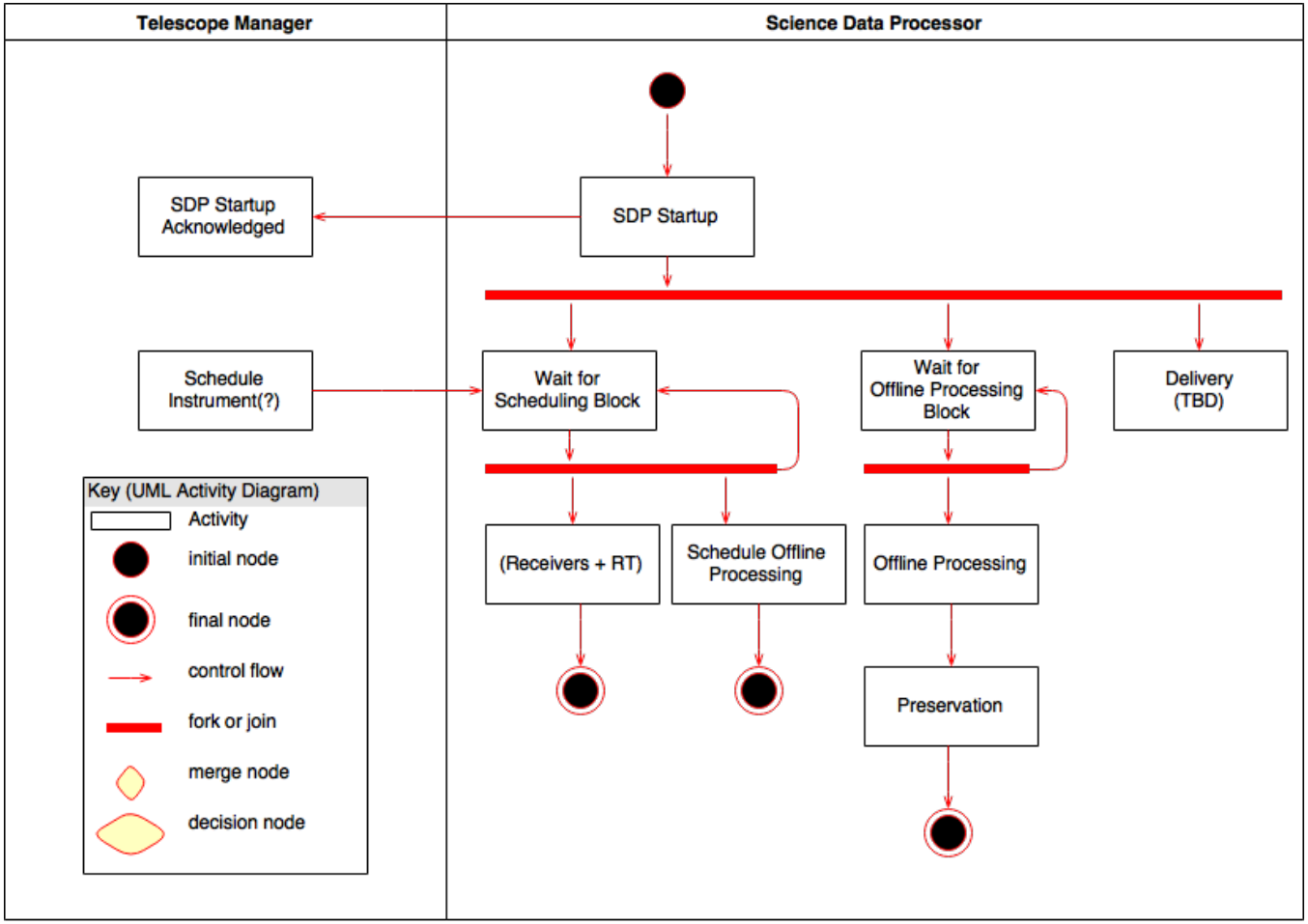
- SKAO has developed a list of HPSO experiments – programmes targeting specific scientific goals and taking long periods (~5000-16000 hours) of telescope time.
- Draft schedule for these taking 5-15 years to complete
- Just an example

Sci Goal	SWG	Science Objective	SKA1 Compone	Hours
1	CD/EoR	EoR - I. Imaging	LOW	5000 hrs
2	CD/EoR	EoR - II. Power spectrum	LOW	10000 hrs
4	Pulsars	Reveal pulsar population	LOW+MID	12750+3200 hrs
5	Pulsars	High precision timing	LOW+MID	4300+3200 hrs
13	HI	Resolved HI out to z~0.8	MID	5000 hrs
14	HI	ISM in the nearby Universe.	MID	2000 hrs
15	HI	ISM in our Galaxy	MID	12600 hrs
18	Transients	Fast Radio Bursts	MID	10000 hrs
22	Cradle of Life	Map dust grain growth	MID	6000 hrs
27	Magnetism	All-Sky magnetic fields	MID	10000 hrs
32	Cosmology	Gravity on super-horizon scales.	MID	10000 hrs
33	Cosmology	Non-Gaussianity and the matter dipole.	MID	10000 hrs
37+38	Continuum	Star formation history of the Universe	MID	16000 hrs

100,000 hours = 11 years

We can use these to generate example SDP use cases and archive growth rates.
Also could enable load balancing if we relax latency requirement of off-line processing.

What does SDP do?



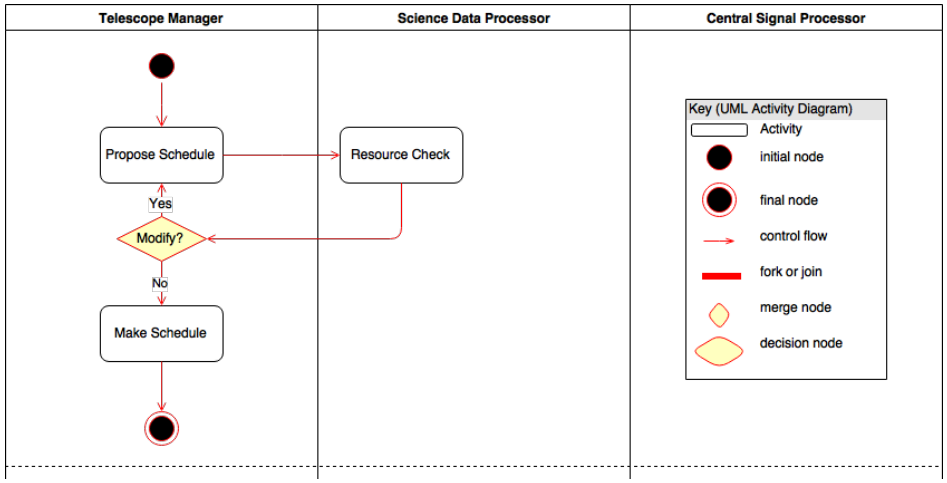
SDP is coupled to rest of the telescope

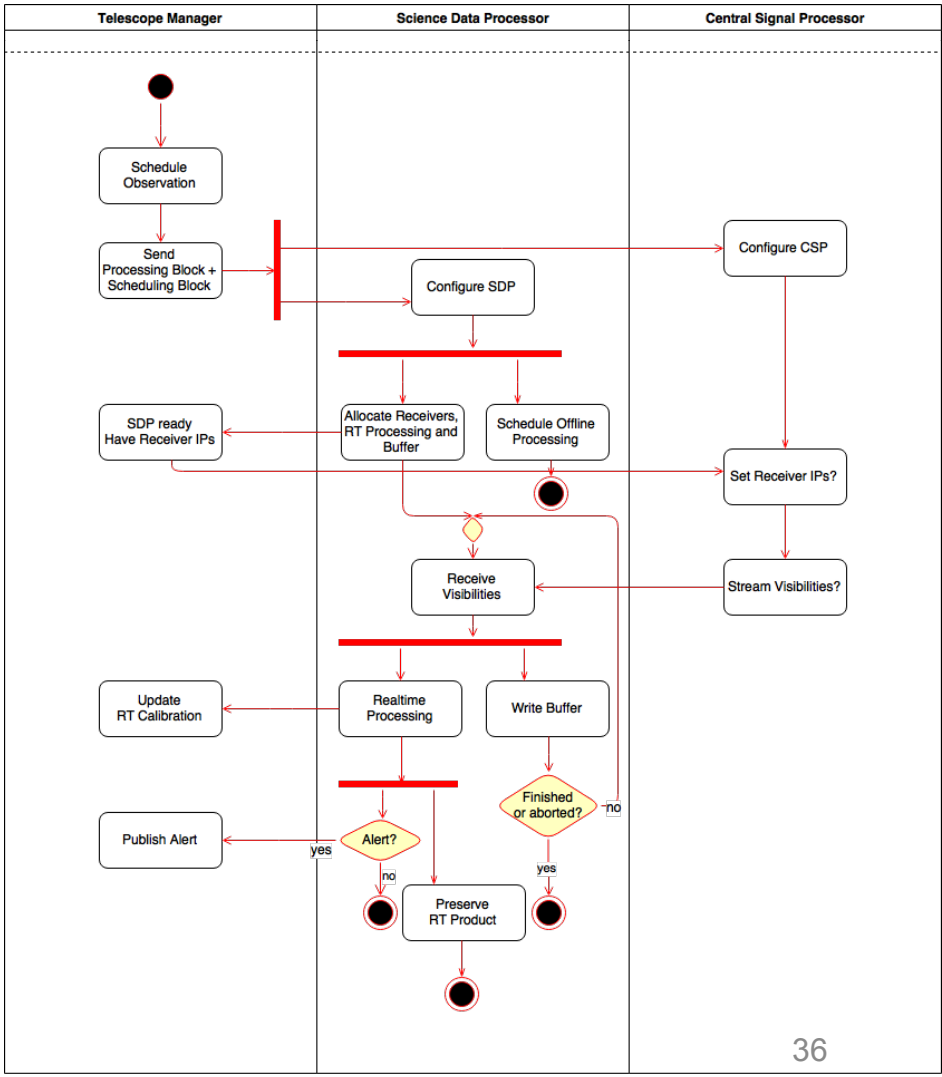
Try to make the coupling as loose as possible, but some time critical aspects

For each observation:

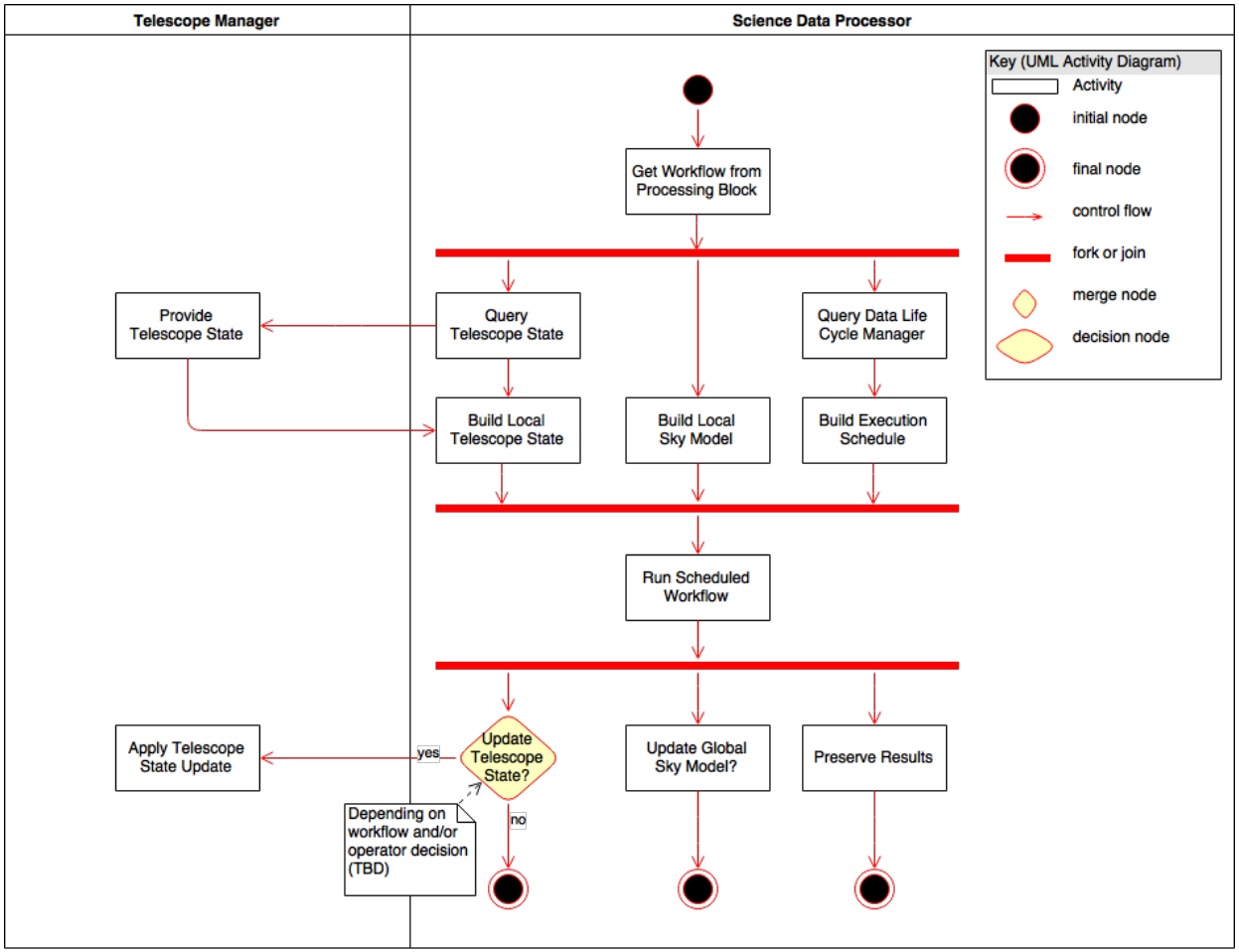
- Controlled by a scheduling block
- Run a Real time (RT) process to ingest data and feed back information
- Schedule a batch processing for later
- Must manage resources so SDP keeps up on timesacle of approximately a week

Real-time activity

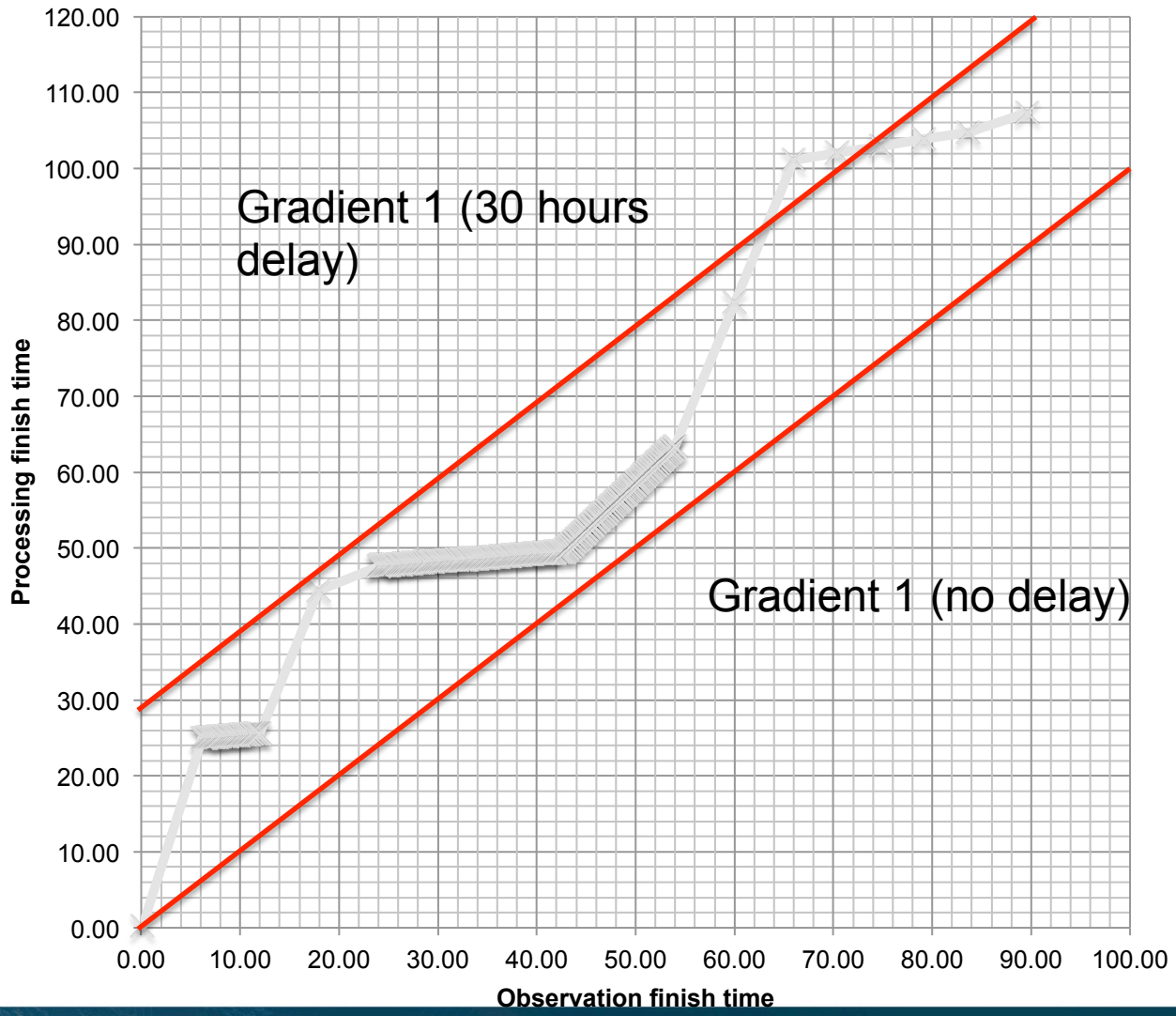




Batch activity



Throughput of Scheduling Blocks



Plot shows finish time for processing as a function of observation finish time. Each marker is a different Scheduling Block.

Our old SDP system sizing assuming a system capable of handling the maximum case would always be ready to start processing a new SB as soon as its sky-time completed. But the system would be idle for much of the time. Here instead, we aim to have a system that can keep up with the observations, on average.

Data “idle time” in buffer