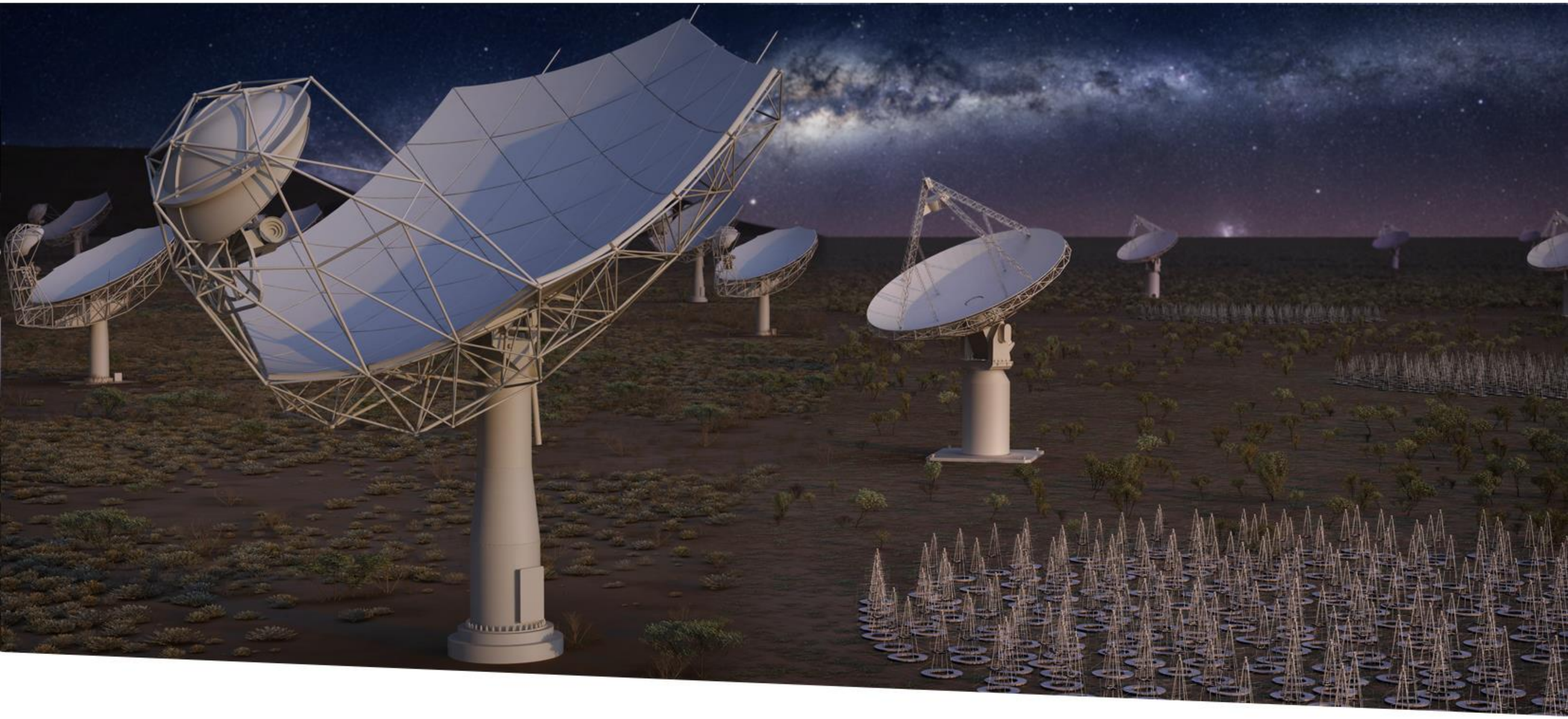
Systems Engineering at SKAO



SQUARE KILOMETRE ARRAY

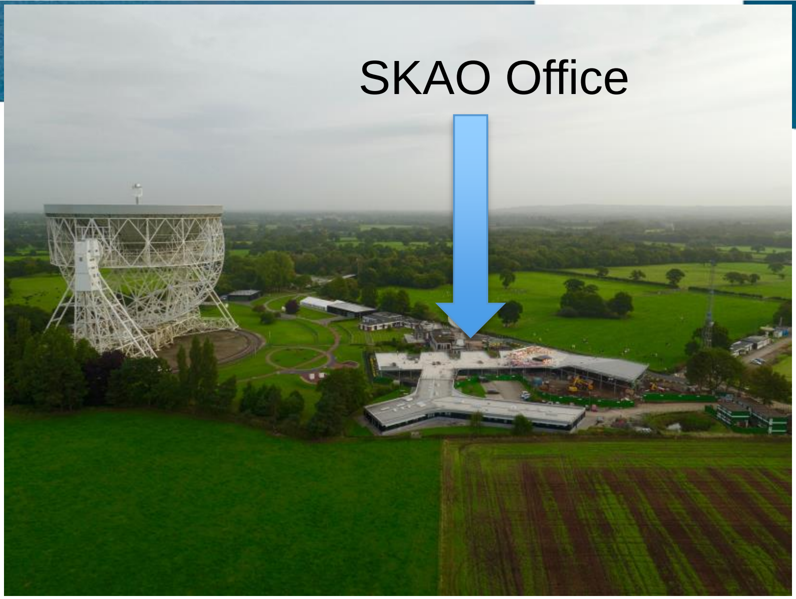
Exploring the Universe with the world's largest radio telescope

Daniel Hayden

22 March 2018

Introduction

This talk will try to give a flavour of what systems engineering at the SKAO involves.



(Dragon's Eye Filming)

The SKAO (SKA Organisation) is responsible for co-ordinating the global activities of the SKA project.

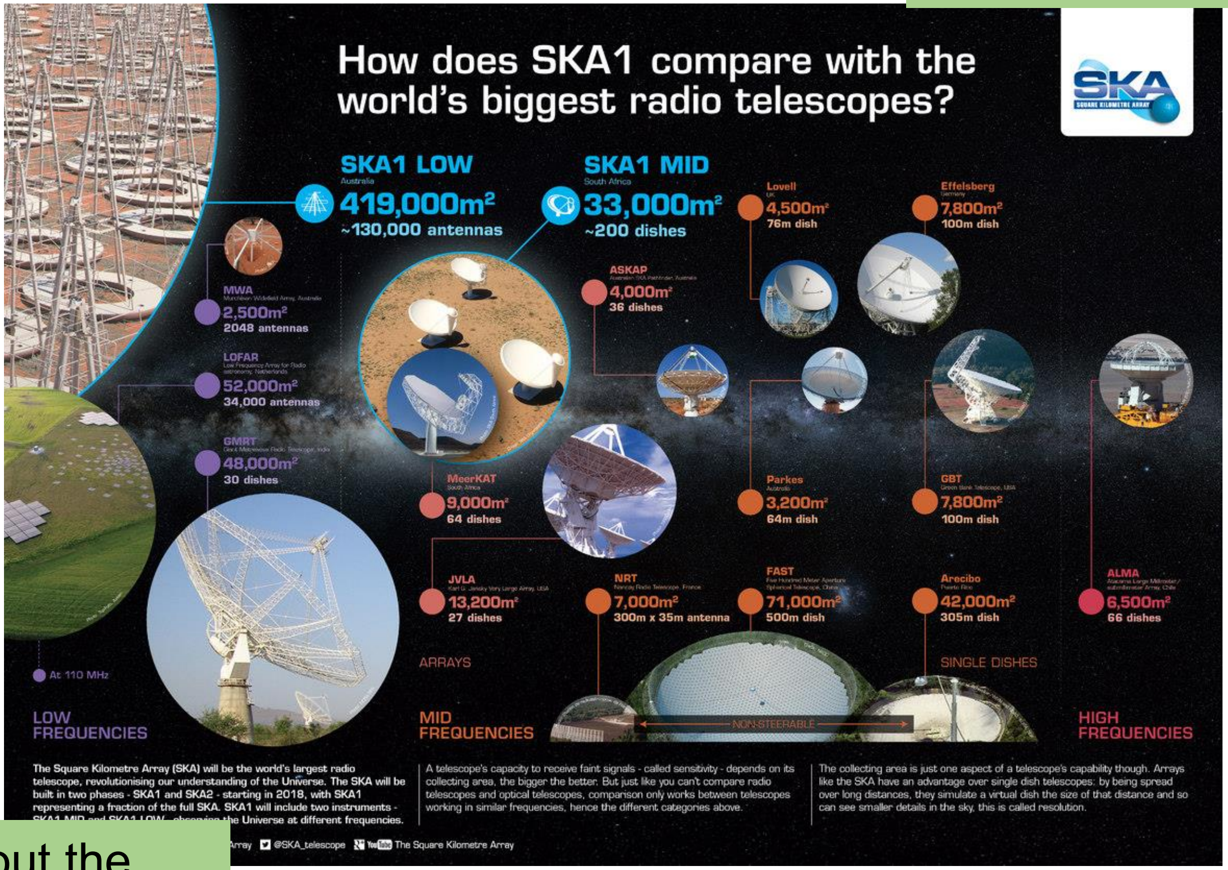
The office for the SKAO is near Manchester, England.

(pe.gatech.edu)

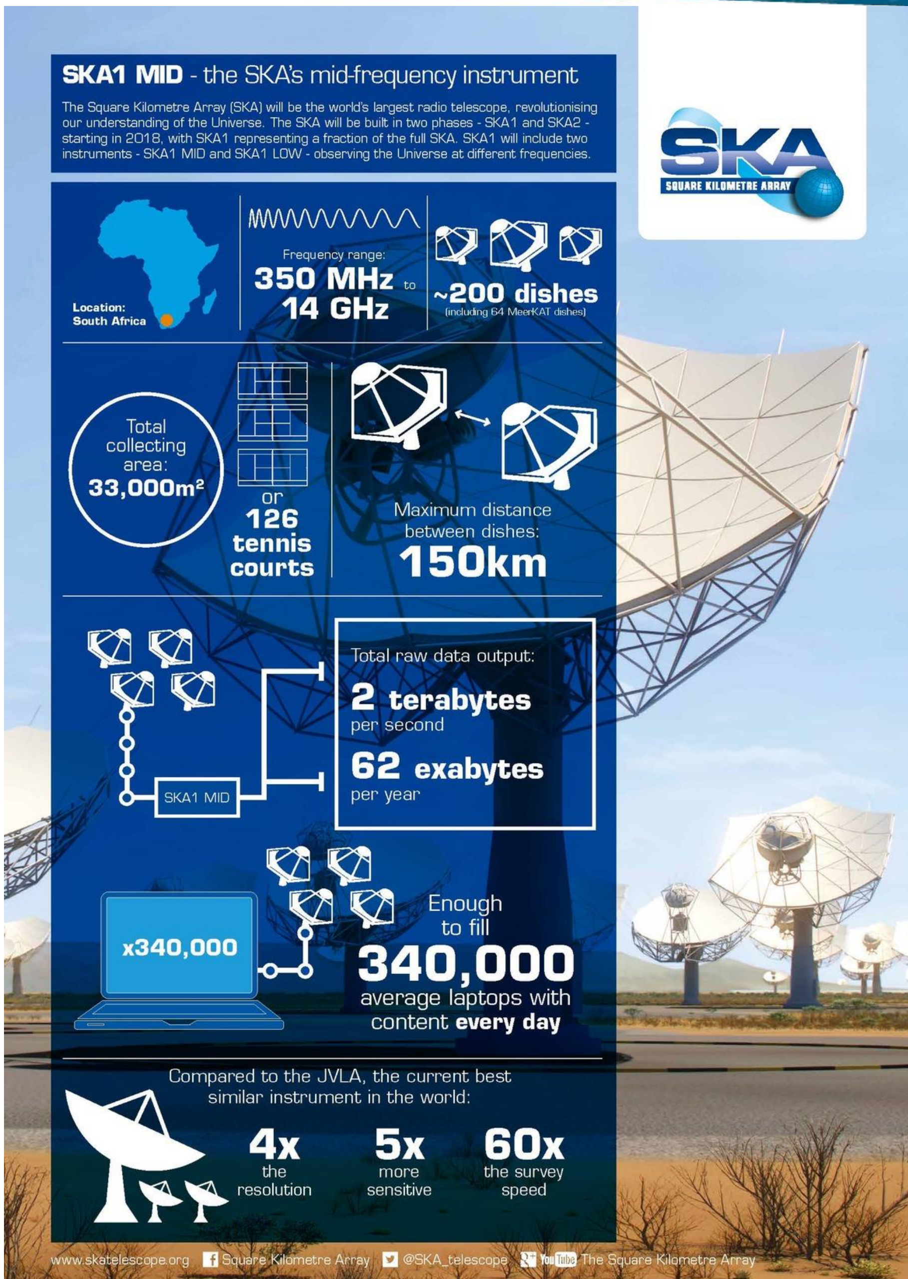
My role? I work as the Systems Engineer for the SKA1-LOW telescope.

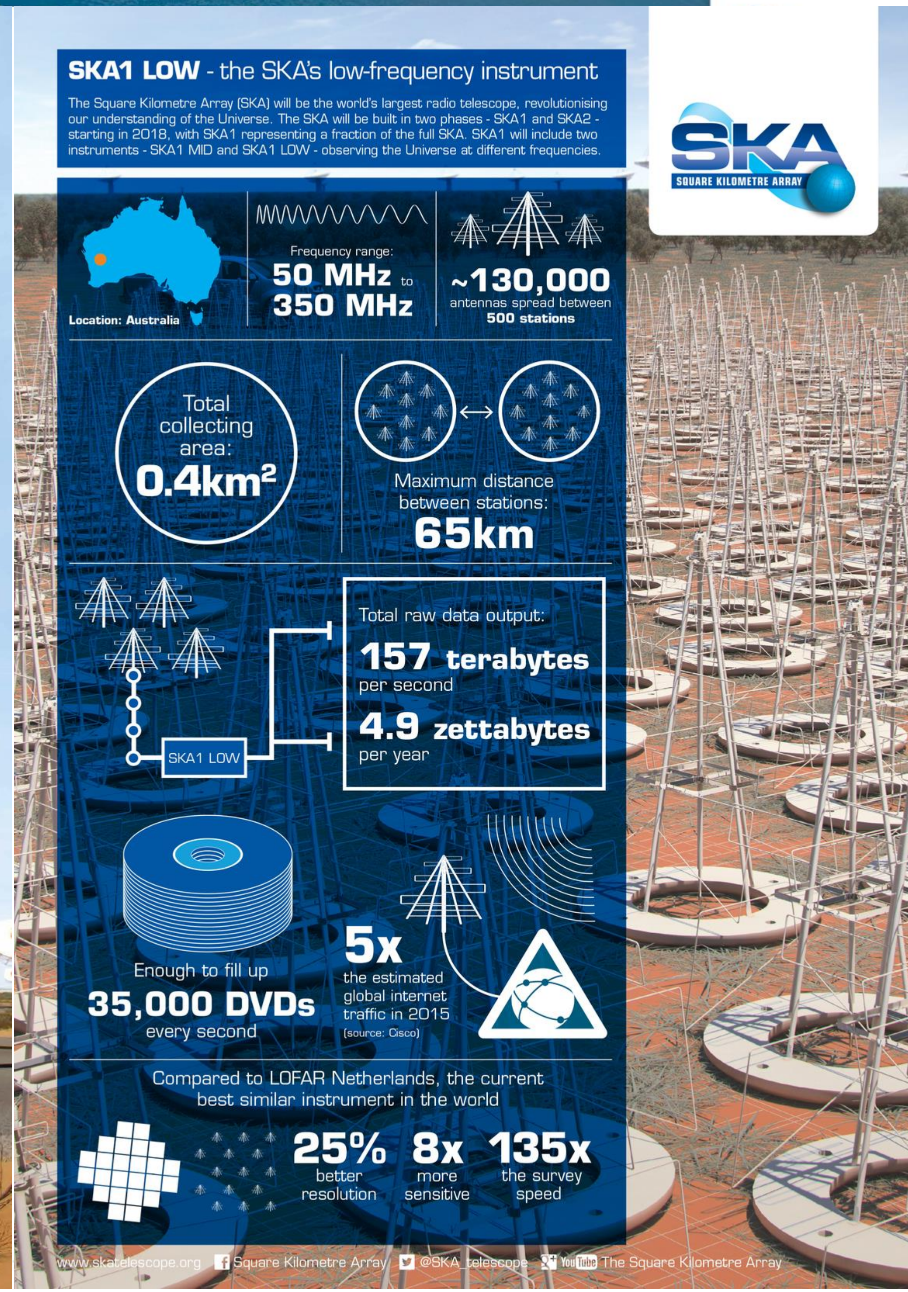
SKA1 - a quick recap

SKA1 MID – S.A
SKA1 LOW – Australia



You've heard about the SKA1 in several talks at this training, so here's just a quick recap

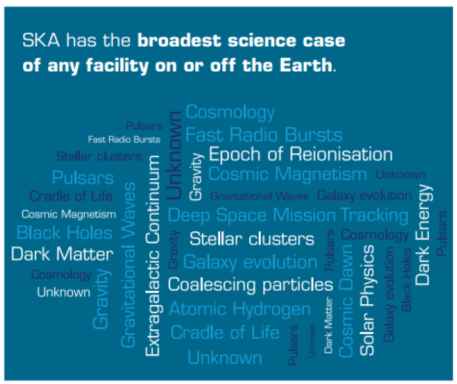


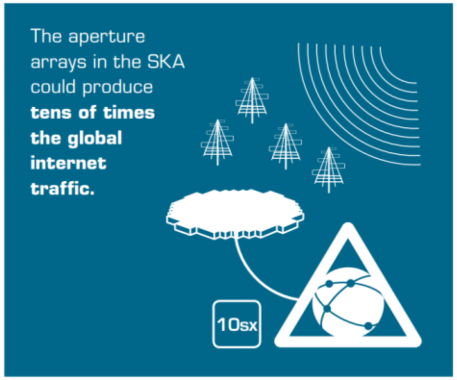


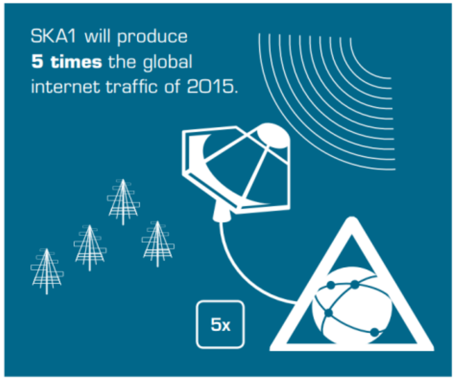
Exploring the Universe with the world's largest radio telescope

SKA1 – 2 types of challenge

- **The technical challenge** – data from 200 dishes and 500 groups of 250 antennas. That's 150,000 baselines at 65,000 different frequencies, so up to 10 Billion data streams!







SKA1 – 2 types of challenge

Global internet traffic ~360 Tb/s

(Cisco: 2016)

SKA1-LOW

~2 Pb/ s

LOW-FREQUENCY APERTURE ARRAY

7.2 Tb/ s

SKA1-MID

8.8 Tb/ s

SIGNAL AND DATA TRANSPORT

CENTRAL SIGNAL PROCESSOR

~50 PFlops

~5 Tb/ s

SCIENCE DATA PROCESSOR

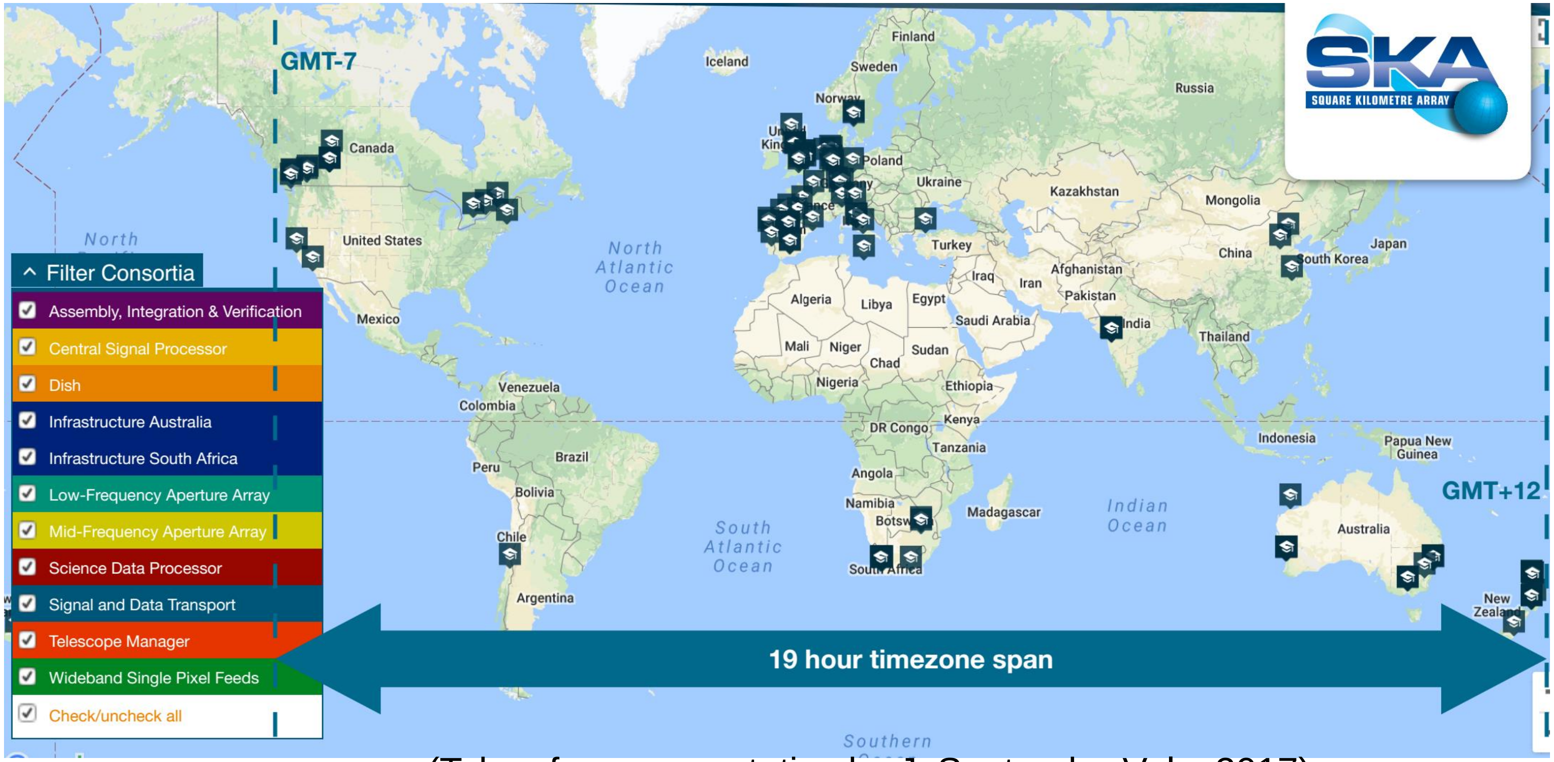
~250 PFlops

~300 PB/ yr

 (Slide taken from presentation by P. Diamond 2018)

SKA1 – 2 types of challenge

- **The organisational challenge** – The SKA is an international project, currently funded by **10** countries, bringing together over **1,000** engineers and scientists from **270** institutions in **20** countries across **20** time zones.



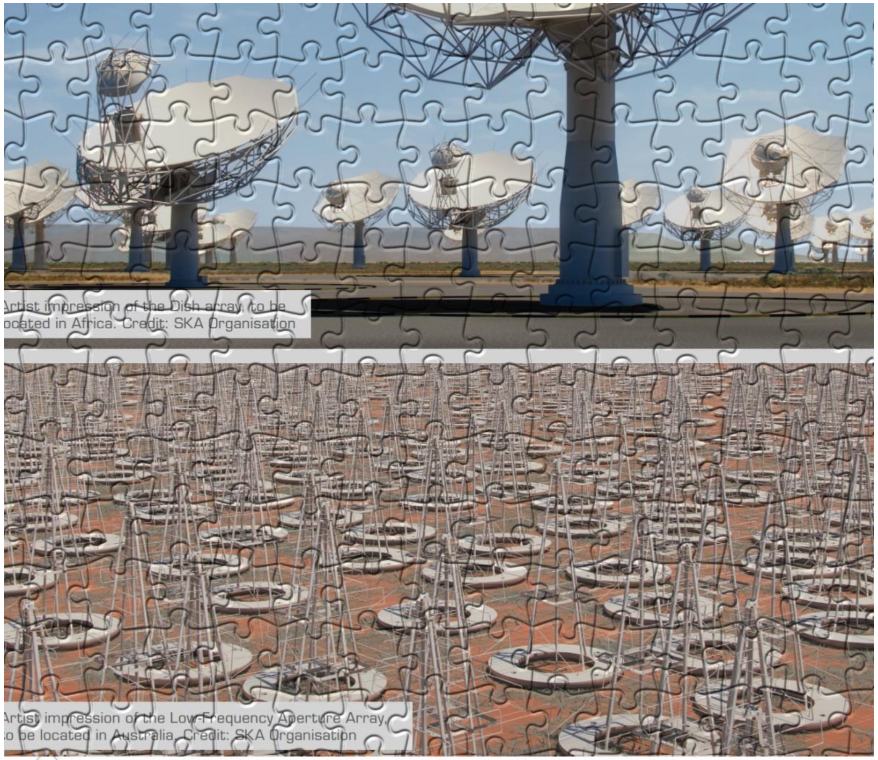
(Taken from presentation by J. Santander-Vela, 2017)

The need for systems engineering

Technical challenge
(millions of components handling billions of data streams)

+

Organisational challenge
(work being split between 270 institutions)

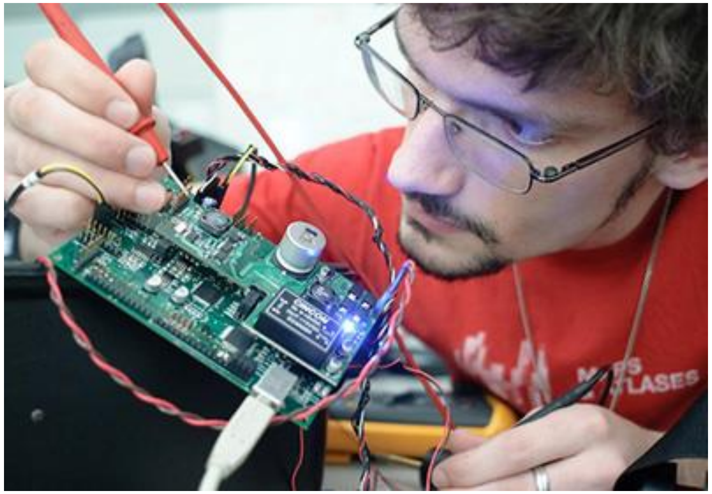


=

A serious need to make sure all the bits fit together into a working unit!

The need for systems engineering

- In addition to designing the detailed parts of the telescope, you also need a system perspective to consider things like:
  - Will all the parts fit together properly? **(interfaces)**
  - Will the end result be what you originally wanted? **(requirements)**
  - Even if all the parts fit and do want you want, is this the most **cost effective solution**?



(rhventures.org)

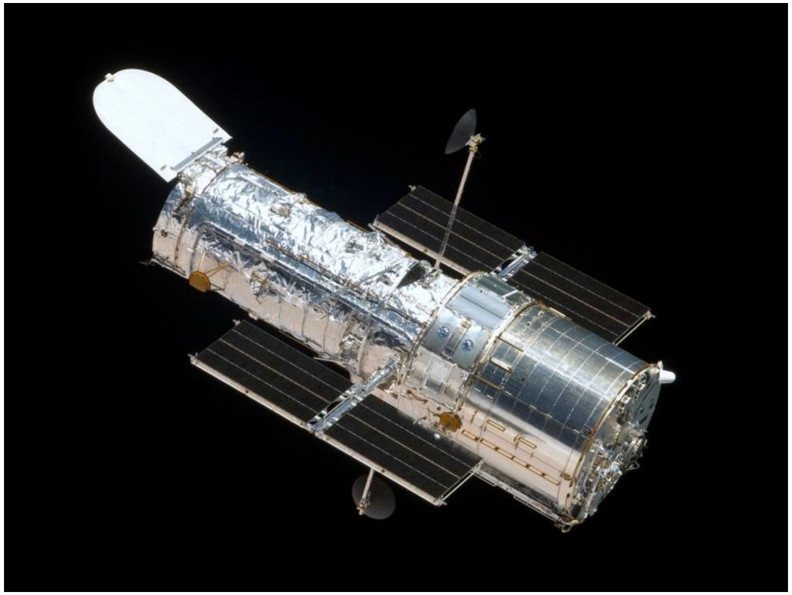
- This is not the kind of project where one person can hold the entire system perspective in their head. And all the designers are in the same building.

- Therefore the system perspective needs to be created and managed using formal techniques. This is what systems engineering does.

The need for systems engineering

- What if you don't do systems engineering?

- **A lesson from Hubble**. A review found that "People who were working the design of the solar arrays were not coordinating with people who worked the design of the control system. Therefore, as the solar arrays would swing in and out of the sunlight, they would irrevocably excite satellite motion in return and there was no image motion compensation or effective correction inside the control loop."



(NJ Slegers, 2012 – inspired by a presentation by Simon Wright)

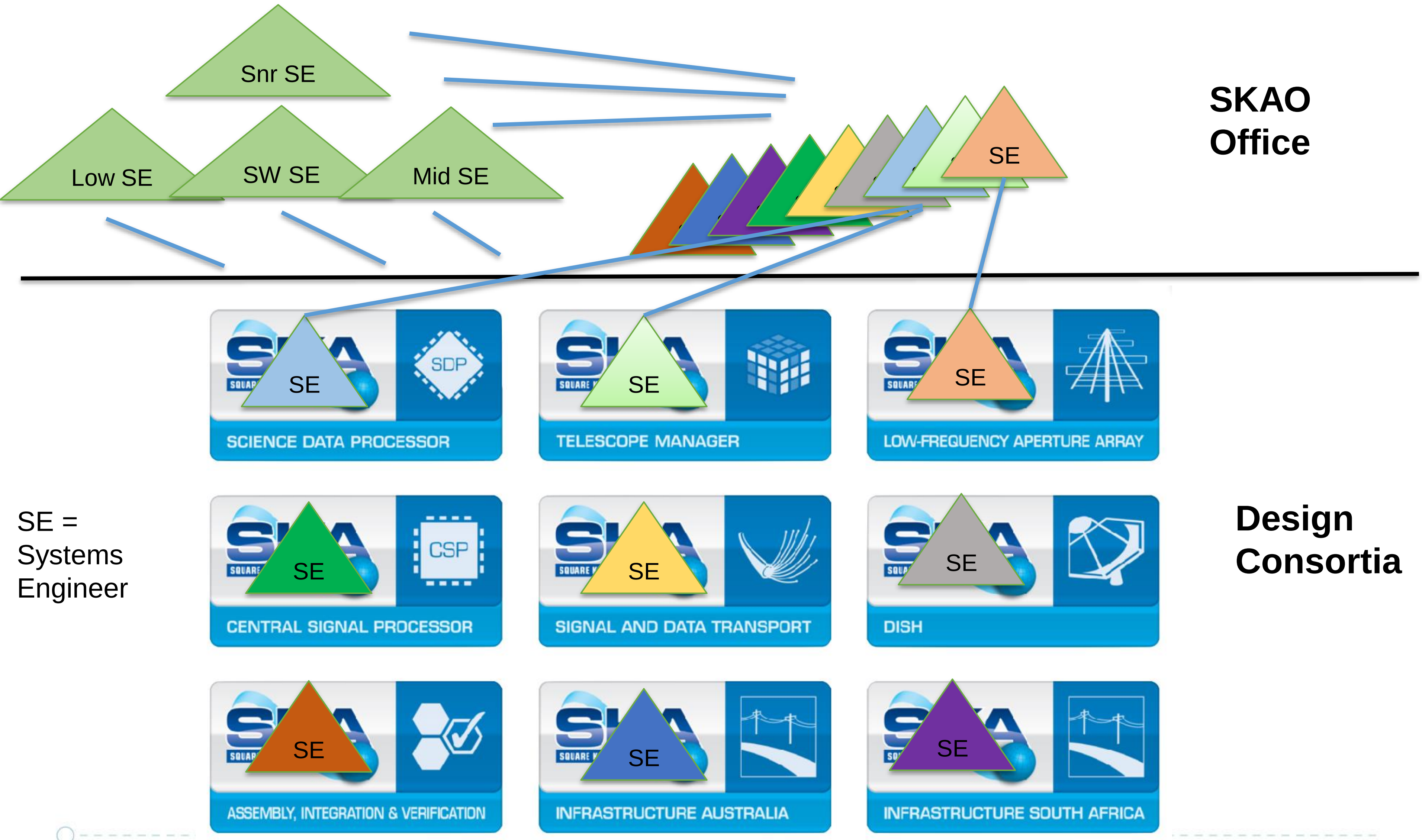
What is systems engineering?

Systems engineering is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect. (FAA, 2006)

(INCOSE SE handbook)

This is one of several possible definitions. But one that I like.

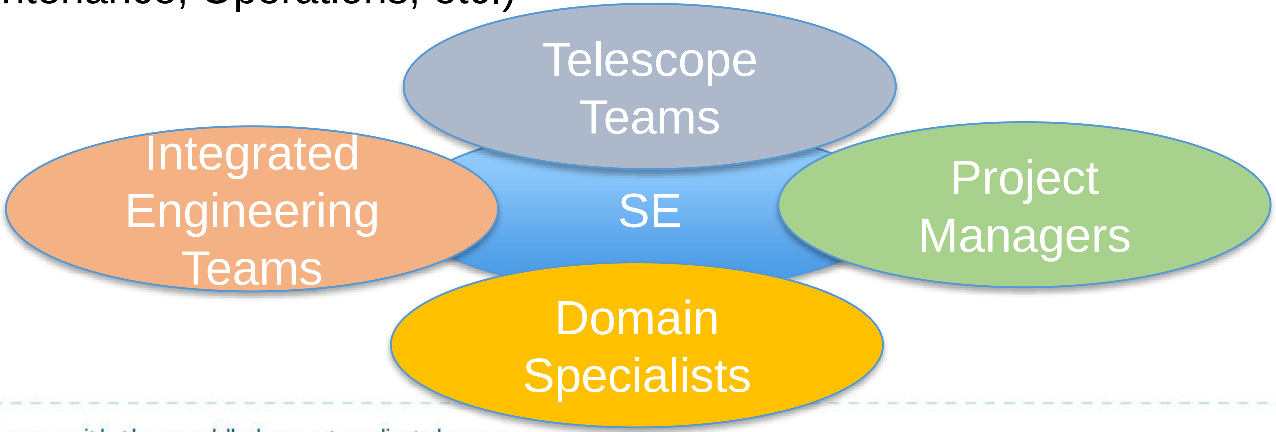
Communication channels



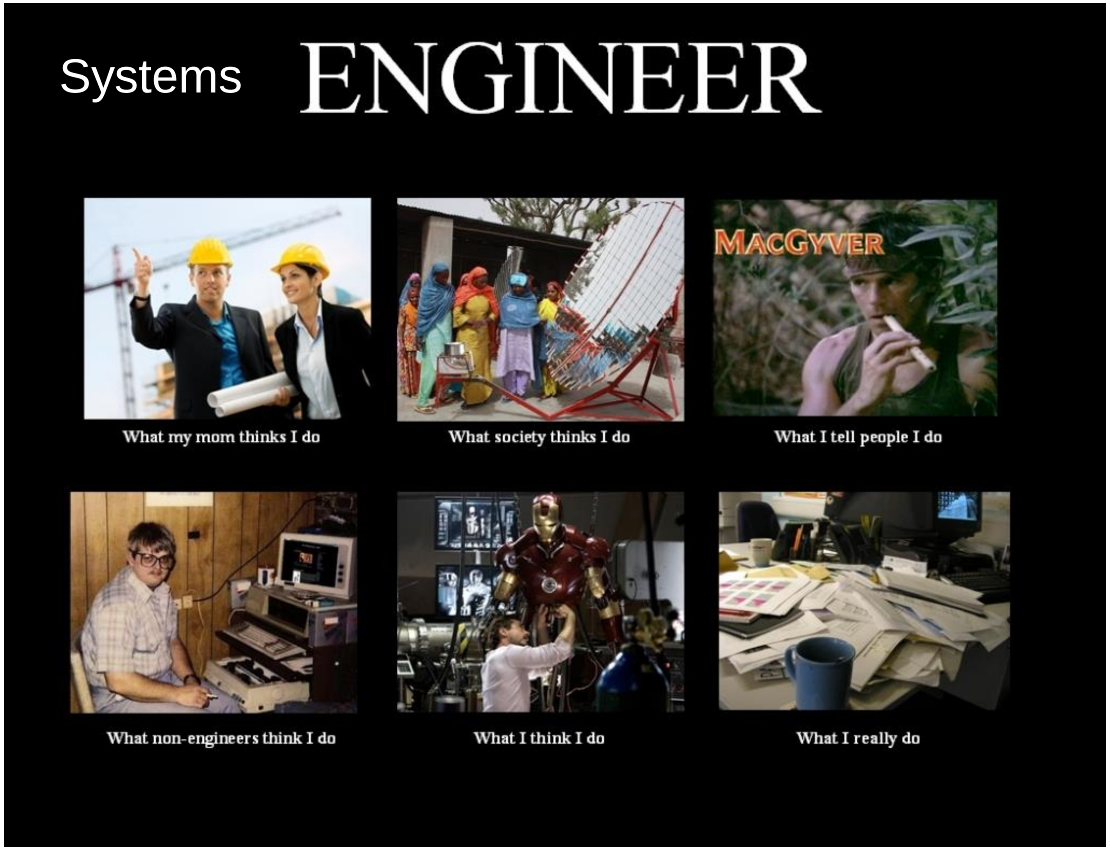
SE = Systems Engineer

Communication channels

- As the previous slide shows, nominally there is an IET (Integrated Engineering Team) SE corresponding to each consortium SE. The IET SEs are in most cases dual-roles.
- There are also 4 dedicated SEs that work with the IET SEs and the consortia SEs.
- To ensure having the right communication channels and knowledge share, the SKAO SEs also work together with:
  - IET Project Managers.
  - Technical Domain Specialists.
  - Telescope Teams (one per telescope).
  - Other multi-disciplinary members of the IETs (Science, Reliability and Maintenance, Operations, etc.)



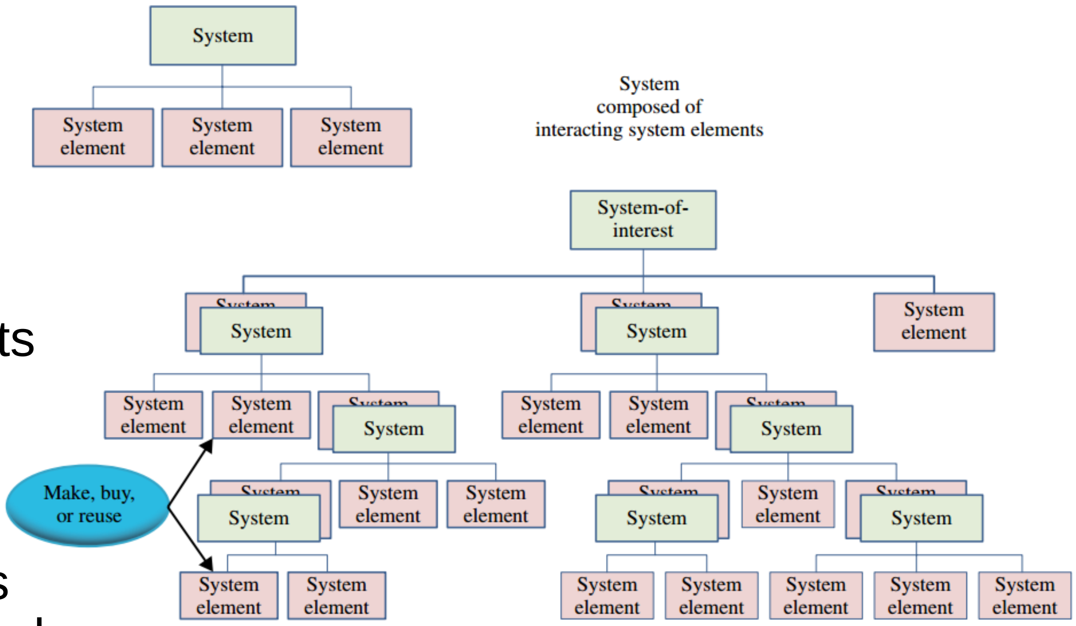
So what are some of the Systems Engineering activities we do at the SKA Organisation?



(whatmyfriendsthinkido.net)

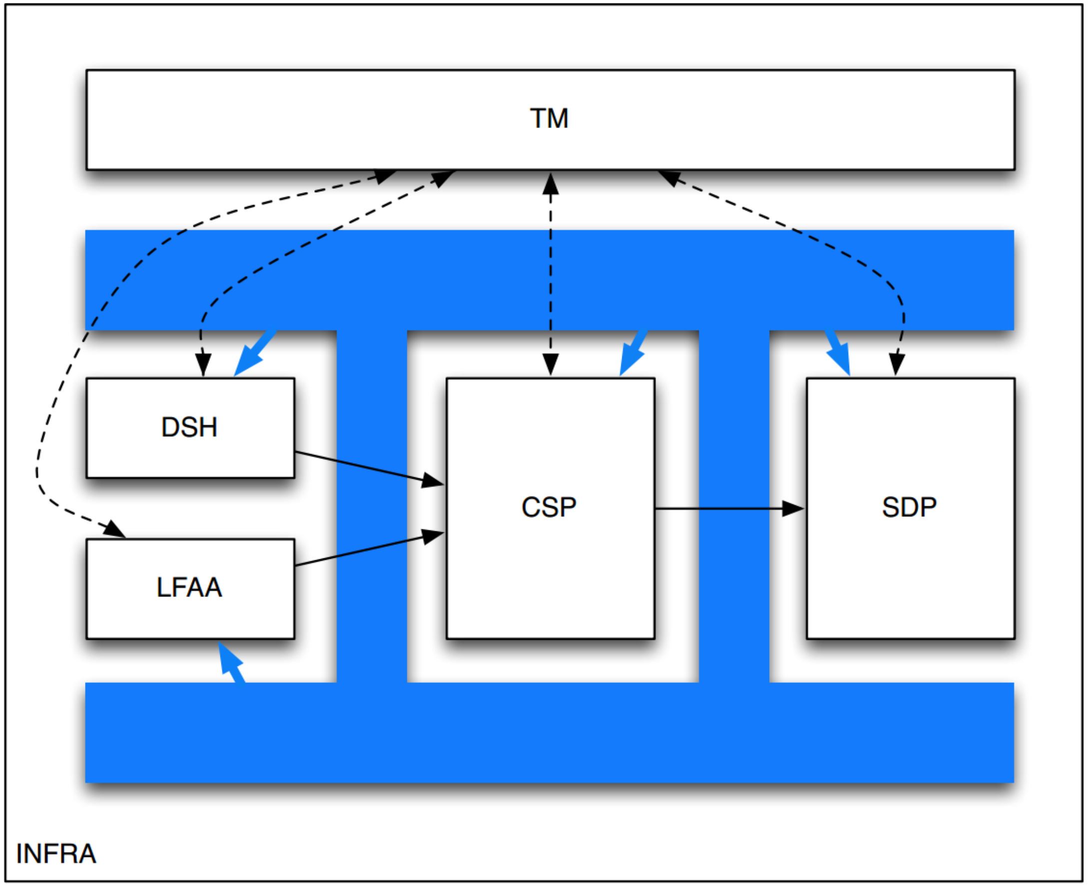
Interfaces

- Remember the 9 consortia mentioned earlier? Each of these is responsible for delivering the design of one ‘Element’.
- Therefore the system is broken up into these Elements or Sub-Systems.
- So between these Sub-Systems, there are interfaces that need careful definition and management.



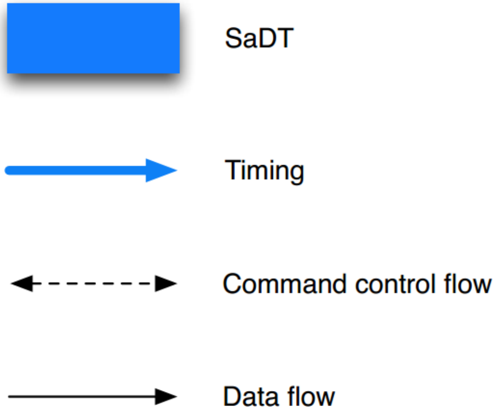
(INCOSE SE handbook)

SKA Elements



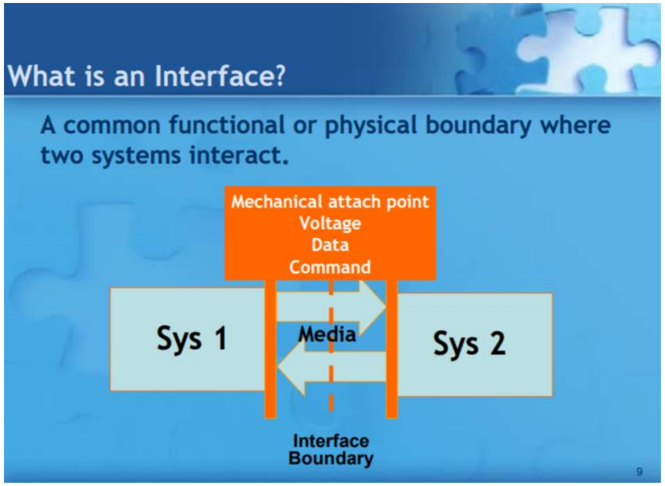
(Taken from presentation by J. Santander-Vela, 2014)

Conceptual figure: it should be split by telescope



Interfaces – what are they?

- System 1 and system 2 might be self-consistent, but they also have to be consistent with each other at the boundary between them.
- This boundary is not fully under either System's control.
- A single definition for this boundary needs to be developed and negotiated by the designers of both systems.
- There is an old saying "If you want to sabotage someone's system, do it at an interface." (Wheatcraft, 2010).



(Wheatcraft, 2010)

- "An interface is a boundary where, or across which, two or more parts interact."
- "An interface is that design feature of a piece of equipment that affects or is affected by a design feature of another system." (Wheatcraft, 2010).

Interfaces – external and internal

DISH	DISH						
CSP	-	*LFAA*	*DISH*	*SDP*			
SADT	*AIVMEERKAT*	*SADT*	*SADT*	*SADT*	*SADT*		
TM	*TM*	*TM*	*TM*	*SDP*	*CSP*	*TM*	
INFRA SA	*AIV MEERKAT*	-	*DISH*	*SDP*	*CSP*	*SADT*	*TM*
INFRA AUS	-	*LFAA*	-	*SDP*	*CSP*	*SADT*	*TM*
	AIV MEERKAT	**LFAA**	**DISH**	**SDP**	**CSP**	**SADT**	**TM**

SKA1 LOW SDP - CSP INTERFACE CONTROL DOCUMENT

Document Number 100-000000-002
Document Type .... ICD
Revision 03
Author .... F. Graser, U. Badenhorst
Date 2017-11-01
Document Classification .... FOR PROJECT USE ONLY
Status .... Released

Name	Designation	Affiliation	Signature	
Owned by:				
D. Hayden	LOW System Engineer	SKAO	Daniel Hayden	
Approved by:				
F. Graser	SDP Systems Engineer	SDP	Ferdi Graser	
J. Bunton	CSP Low Systems Engineer	CSP	John Bunton	
R. Gabrielczyk	SADT Systems Engineer	SADT	Robert Gabrielczyk	
J. Santander-Vela	SDP Element System Engineer	SKAO	[illegible]	
W. Turner	CSP Element System Engineer	SKAO	W.Turner	
R. Olguin	SADT Element System Engineer	SKAO	Rodrigo Olguin M.	
M. Caiazzo	Senior System Engineer	SKAO	Marco Caiazzo	
Released by:				
A. McPherson	Head of Project	SKAO	[illegible]	
			Date:	Nov 15, 2017

- This N Squared diagram shows all the External interfaces that exist (~**35** in number because there is usually one interface for each telescope). The SKAO is responsible for these.

- Each of these 35 interfaces is defined and detailed in an Interface Control Document (ICD).

- Since each Element consists of many sub-systems in turn, there are also a number of Internal interfaces for each Element. The consortia are responsible for these.

Interfaces – external and internal

- The distinction between internal and external interfaces is based on organisational boundaries during the design phase.
- For construction the boundaries might be different. If they are, the content of the external and internal ICDs will need to be re-packaged along these new boundaries.
- Thus the distinction between external and internal interfaces will fall away during construction.

(fbaforward.com

The moral: always be aware that distinctions and categories in SE are not 'intrinsic' but are human-made to serve a particular use at a particular time and this can change.

An aside – how to split up a system


(springbok-puzzles.com)

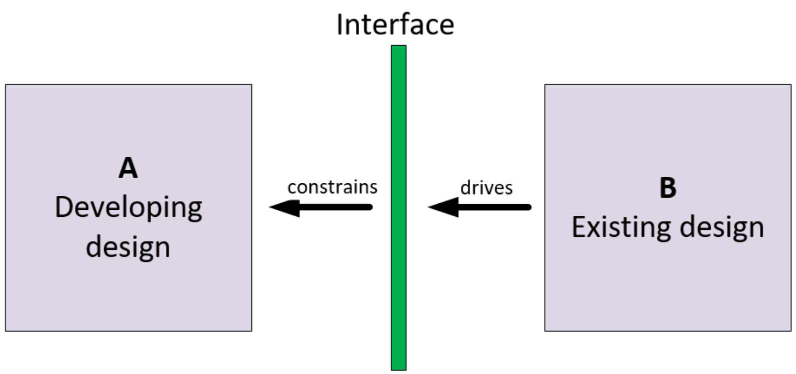
- You may have wondered by now – what determines how the SKA is divided into its Elements?
- There are many ways to break up a system. Systems engineering encounters many of these.
- A system can be decomposed along many boundaries, such as:
  - Functional
  - Physical - Line Replaceable Unit
  - Organisational
  - Contracts
- Often it is necessary to use different 'breakdowns' and be able to map from one to the other.

Back to interfaces – process challenges

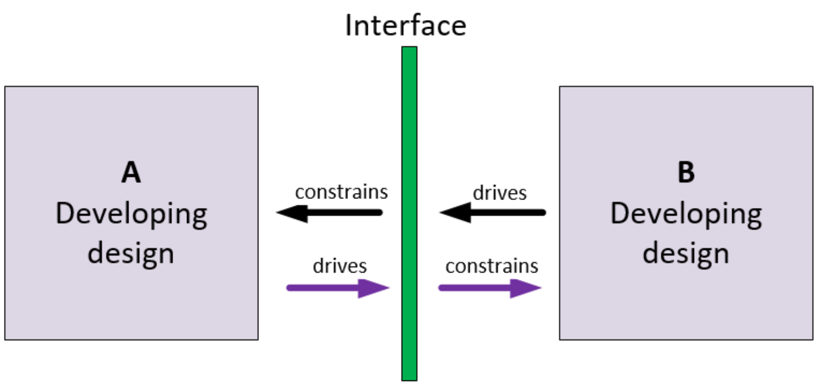
- A role of the SKAO is to:
  - review the external ICDs
  - manage their change process
  - Negotiate resolution of issues with the consortia when necessary
- There are several ‘process’ challenges to do with managing ICDs. Such as…

(ewocnj.org)

CHALLENGE 1



Scenario 1



Scenario 2

Scenario 1: System A (developing) has an interface with system B (existing). B drives the definition of the interface which constraints the design of A.

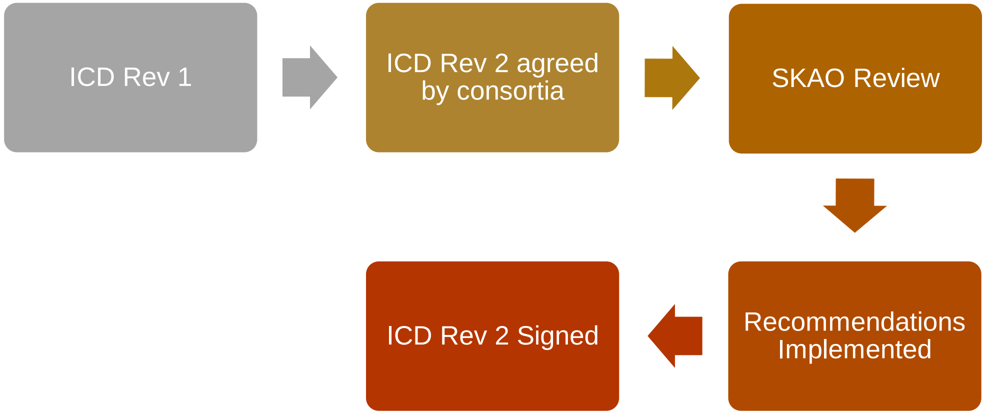
Scenario 2: Both A and B are both being developed concurrently. A drives the definition of the interface which constrains the design of B and vice versa! This is a bit of a chicken-and-egg problem. The interface definition therefore has to evolve iteratively.

CHALLENGE 2

- The definition of an interface in someone’s ‘head’ may have changed a while before this change is formally captured.
- Although the design of an interface is always changing, it has to be frozen at various times to give a baseline that can be reviewed or referenced elsewhere in the design.

Process description

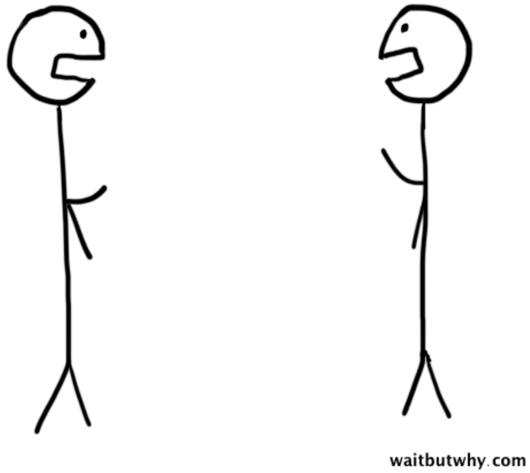
The change process for an ICD can take time. First the 2 consortia need to agree the changes among themselves. Then the SKAO has to review the new ICD. They issue recommendations to the consortia. The consortia respond to the recommendations which are then sometimes adjusted. When they are implemented, the ICD can be signed.



Baselines and the meaning of a ‘signature’

Don’t refer to that part of the design, it’s wrong!

Yes, but at least it’s baselined!

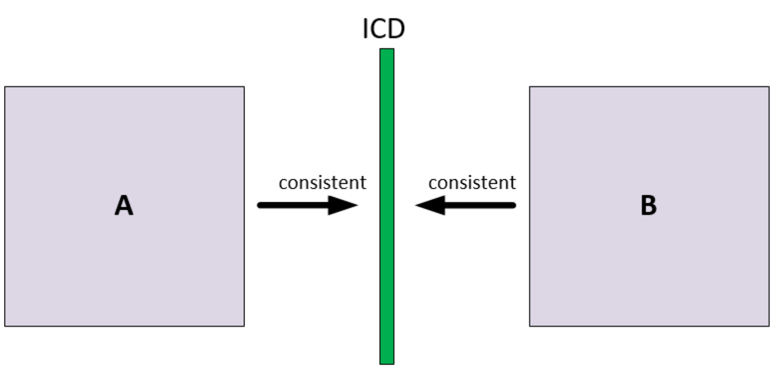


Imagine design **A** refers to design **B** refers to design **C**. But which revision/incarnation of A and B and C? One that has been agreed and frozen, even if it is not the most correct and current version.

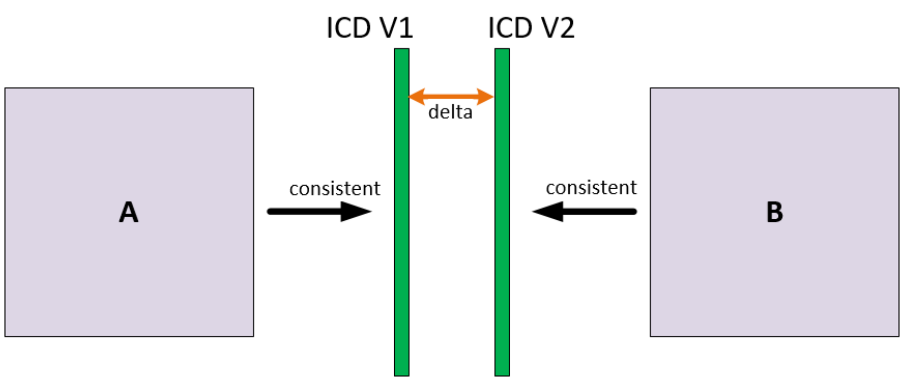
Often this is the meaning of a document signature. It doesn’t mean that the document is 100% correct. But rather that it has been agreed in order to be frozen.

CHALLENGE 3

- Critical design reviews (CDRs) for some Elements are separated by 1 year.
- This means the ICD is frozen at CDR1. But Element 2’s design continues to evolve for 1 more year. This evolution will likely drive changes to the interface, leading to a different version of the ICD to be frozen at CDR2.
- But then you have 2 different ICDs describing 1 interface! If two designs are consistent against two different definitions of the same interface, there’s no guarantee the designs will be compatible with each other.



Scenario 1



Scenario 2

Types of interface issues from reviews

- Element 1 assumes that Element 2 is providing certain equipment where this assumption is not recorded in the ICD and turns out to be wrong.
- An interface needed to enable a certain functionality is not yet defined.
- In several cases there is an ICD triangle: a logical data interface between A and B, a physical transport interface between A and C, and physical transport interface between C and B. Sometimes these are not in sync (i.e. the bandwidths described in the one are not supported by the link capacity described in the other).

(twelveenough.com)

- Data (e.g. calibration and beamforming parameters) often traverses multiple interfaces. These interface descriptions need to be checked for consistency.

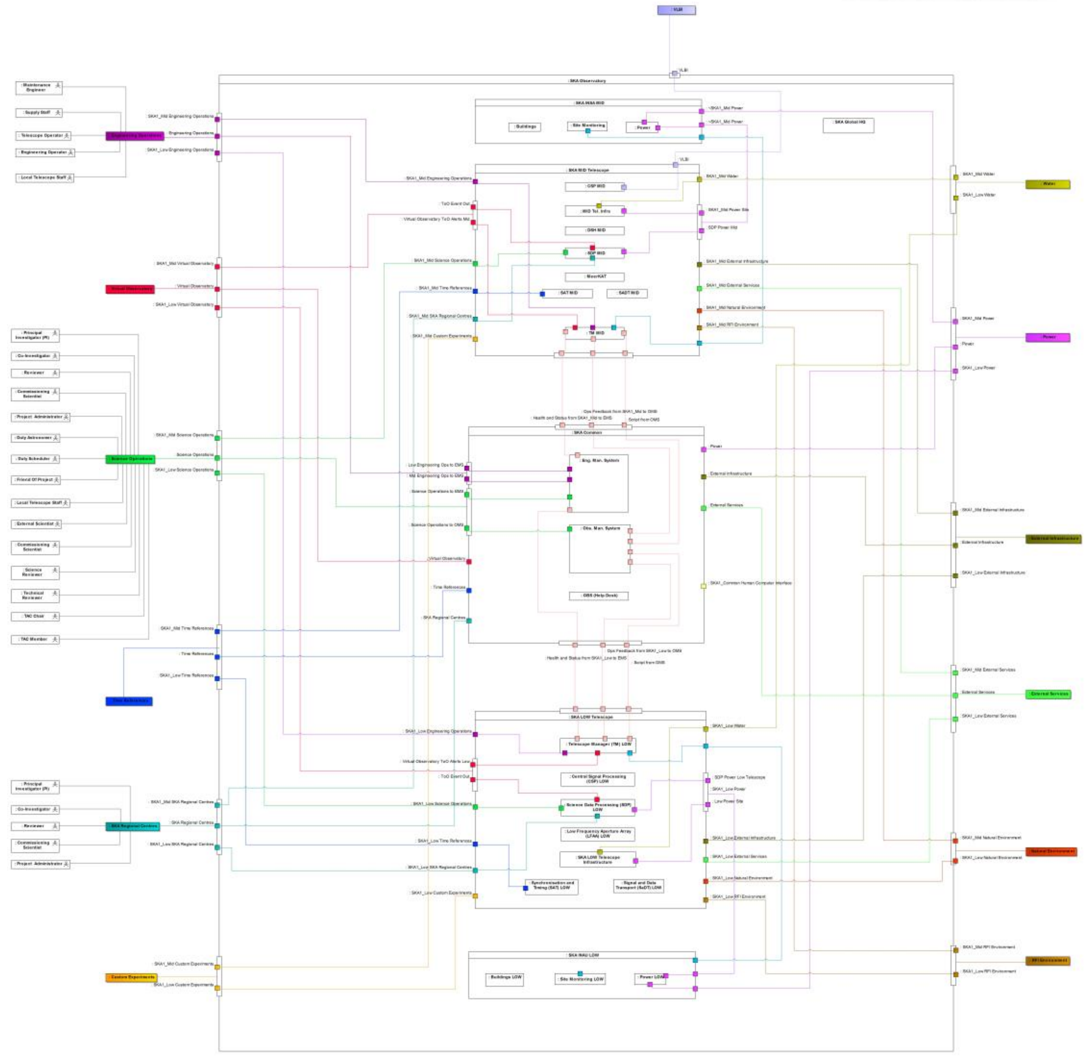
Examples of interface issues from reviews

- Is the infrastructure interface to the hydrogen masers stable enough in terms of vibration?
- Access to the TEC readings from GNSS/GPS receivers needs to be defined.

- The interface for the sequential powering up of equipment to prevent step loads needs to be defined.
- The timing accuracy provided by Element 1, consistent with their architecture, is not sufficient for the needs of Element 2.
- A change to the design removes the long range transmitters from the scope of one Element and transfers them to another Element. The interface needs to be redefined.
- Clipping is performed above a certain threshold. But who sets this threshold and how is it communicated?

Not just interfaces between Elements

- Interfaces don't just exist between Elements within a telescope. They also exist:
  - ➢ Between telescope Elements and products that are common to both Telescopes (e.g. Engineering and Observation Management Systems).
  - ➢ Between telescope Elements and products/facilities external to SKA.



All these interfaces need to be identified.

Moving on now – to requirements!

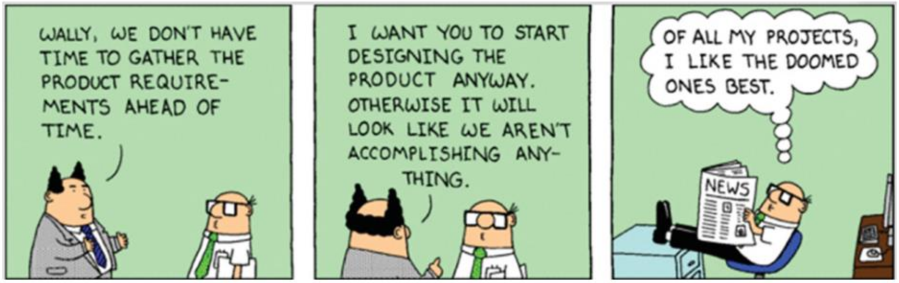
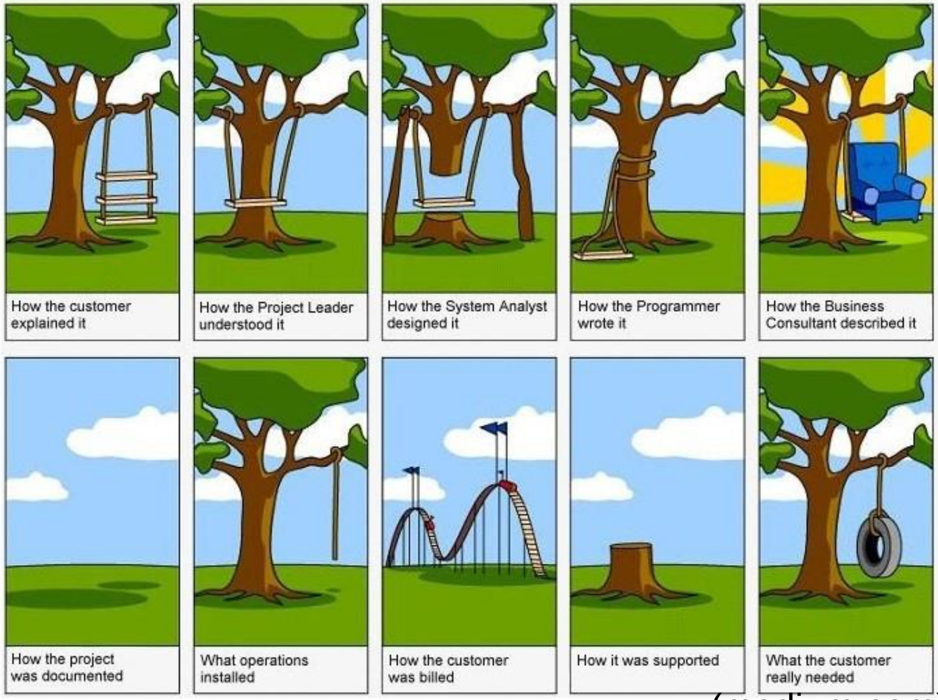


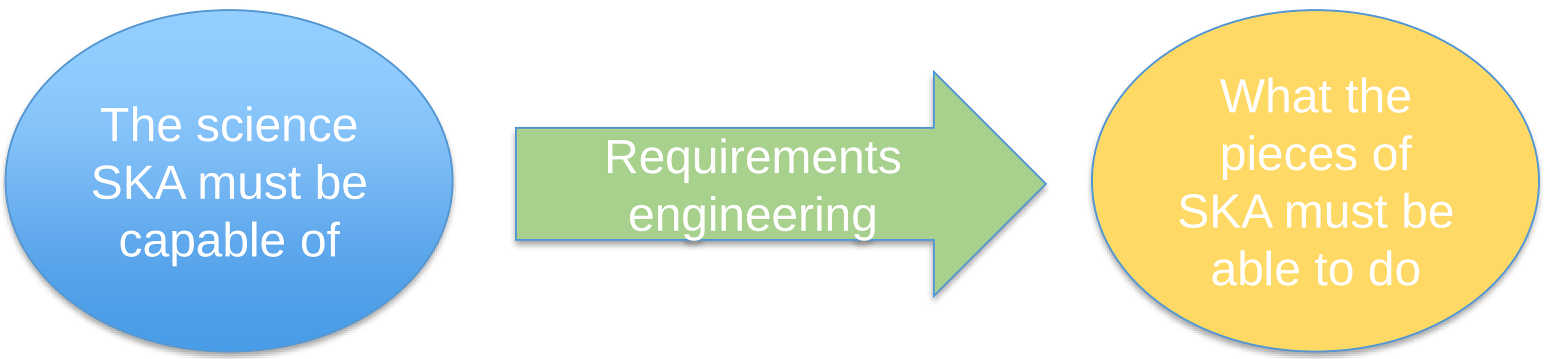
FIGURE 3.4 Importance of the concept stage. DILBERT © 1997 Scott Adams. Used with permission from UNIVERSAL UCLICK. All rights reserved.

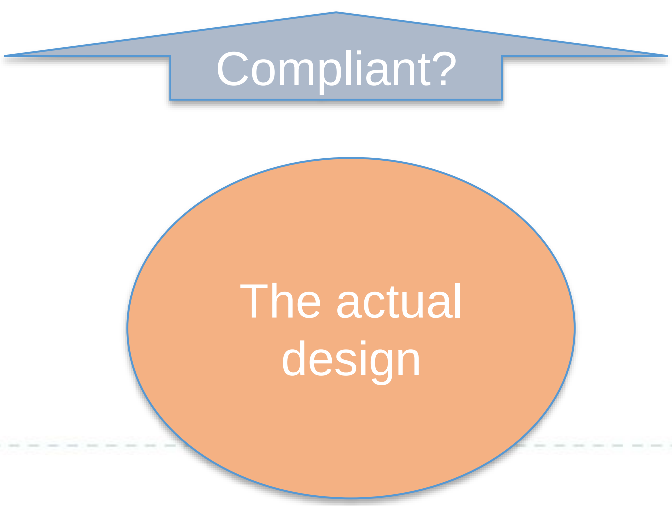


(medium.com)

Requirements

- A big part of systems engineering is requirements engineering.
- This is essentially the following process:

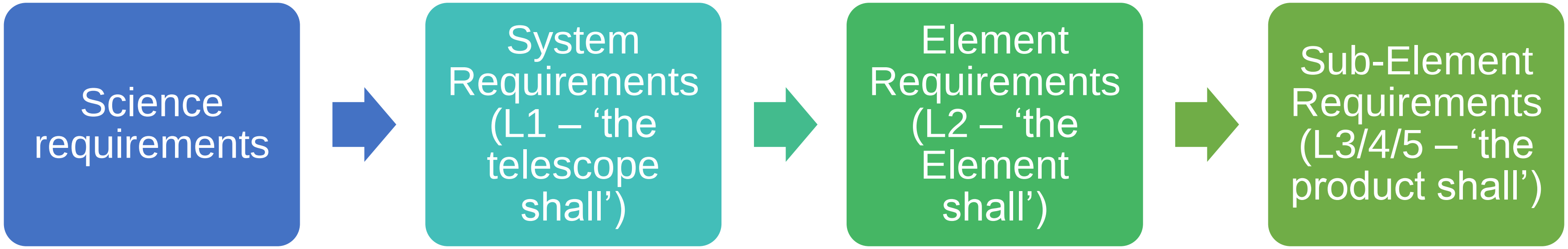




Requirements engineering translates between what the SKA must be be able to achieve and what its individual pieces must be able to do. We then check that the actual design is compliant to this.

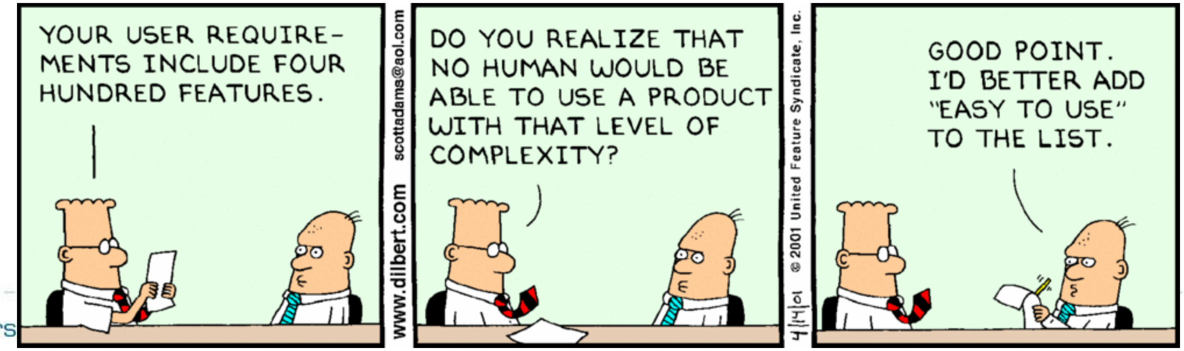
Requirements

- There are about ~ **600** SKA system L1 requirements. The SKAO allocates each of these to one or more Elements.
- The Element's consortium is then responsible for analysing these requirements to determine what role their Element needs to play in fulfilling this requirement. This leads to L2 requirements **(~2500).**
- The same derivation process is applied one level deeper in the product hierarchy to obtain L3, and in some cases L4 and L5 requirements.



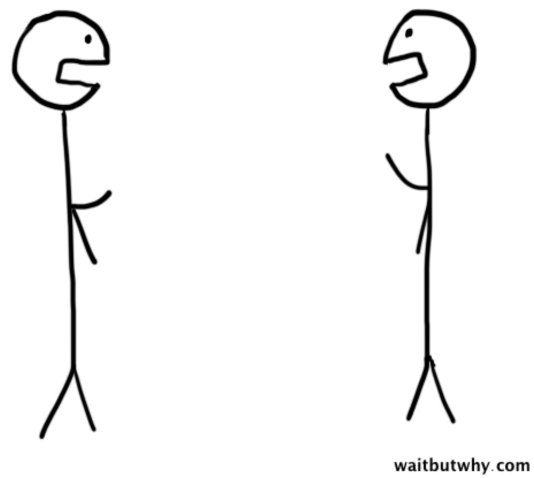
Requirements

- Some things to take into account with requirements:
  - When allocating a performance requirement (e.g. sensitivity, timing accuracy) to multiple Elements, the SKAO has to budget this performance across Elements. Each Element needs to meet their portion of this budget.
  - Interface requirements derive from ICDs rather than L1 requirements.
  - Sometimes an Element needs to create new requirements that don't yet flow down from an L1. In this case they will derive from Assumptions.
  - Requirements can be Performance, Functional, or Non-Functional.



Compliance

Jerry, why do we build a L1 compliance matrix?



We want to know if the Element designs meet the L1 requirements.

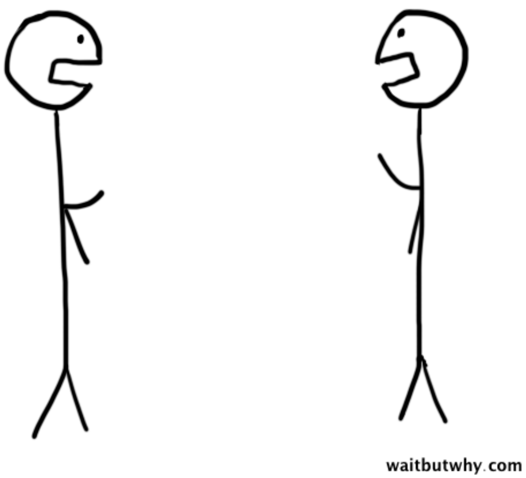
So how do we build such a matrix?

In three steps:

1) Build a matrix of compliance of Element design against L2 requirements
2) **Correctly** trace L2s to L1s
3) Roll up compliance against L2s to compliance against L1s

Compliance

But why do you say **'correctly'** trace L2s to L1s?



Well, if the **logical relationship** of the L2s to the L1s is wrong, then even if the design is compliant against the L2s, it doesn't imply compliance against the L1s.

What then is the right logical relationship of L2s to L1s?

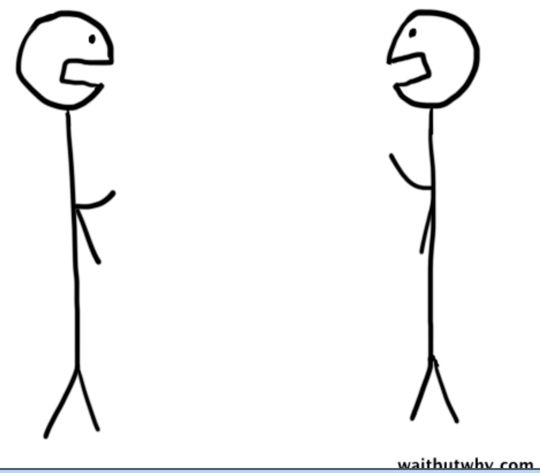
Sufficiency condition:

Achievement of all children requirements → Achievement of parent requirement.

This is diagnosed through different **patterns of traceability**

Compliance

Ah, okay. But this raises a few more questions



Such as…?

Such as:

- What are these different patterns of traceability?
- What exactly does ‘compliance’ mean? What are the possible enumerations?
- How is compliance rolled up from L2 to L1?

We won’t answer these questions now.

Requirements

- Bear in mind requirements have different purposes:
  1. To drive the design
  2. To justify the design (by proving compliance against them)
  3. To drive specifications for procurement
  4. To drive verification requirements

- For the Element critical design reviews (CDRs), the primary focus is on 2. But 4, for example, will be a focus for System CDR.

- The SKAO is responsible for the definition, allocation, and budgeting of the L1 requirements. As well as reviewing the traceability and compliance of L2+ requirements.

Back to: how to split up a system

- As I said earlier, there are many ways to split up a system. The SKA has a single Product Breakdown Structure (PBS), which is a physical division of products. These products will be grouped into 'work packages' which will be contracted out – thus work packages can't cut across product divisions.

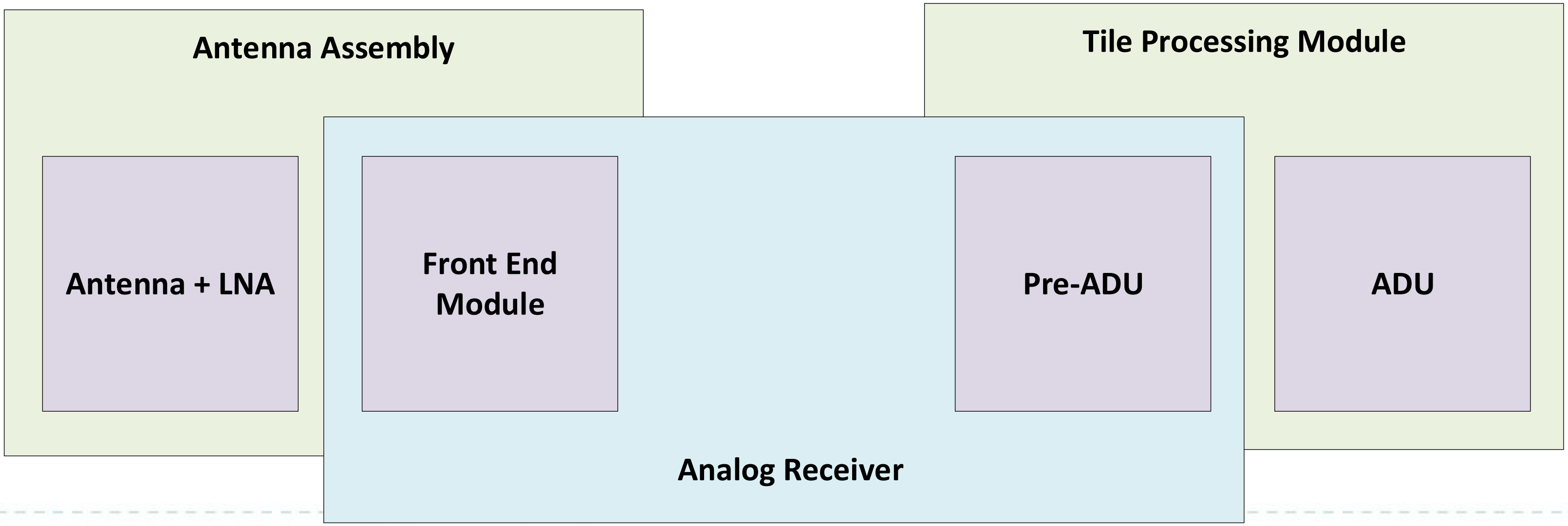
- But even deciding on a single, optimal physical PBS is not easy. Here's an example…(next page, not the cartoon below)

PBS

- For SKA1-LOW, the design of the processing module has been split up into the analog and digital parts (Pre-ADU and ADU). So the design is treating these as separate products.

- The design of the Pre-ADU is coupled to the design of the Antenna Front End Module. This coupled product is called ‘Analog Receiver’.

- So how do we divide this up as physical products? According to the blue or green groupings as shown below?



PBS – other branches in the tree

- There will be 4 array releases (AA1-AA4) of the SKA in addition to an Integration Test Facility (ITF).
- Each of these will consist of a different, increasing number of antennas.
- Final configurations for some products won't be ready till AA3 or AA4 (e.g. correlator, science processor, telescope manager, clock). Therefore emulators need to be used.
- Other simulators and temporary products will also be needed.

(lifebuzz.com)

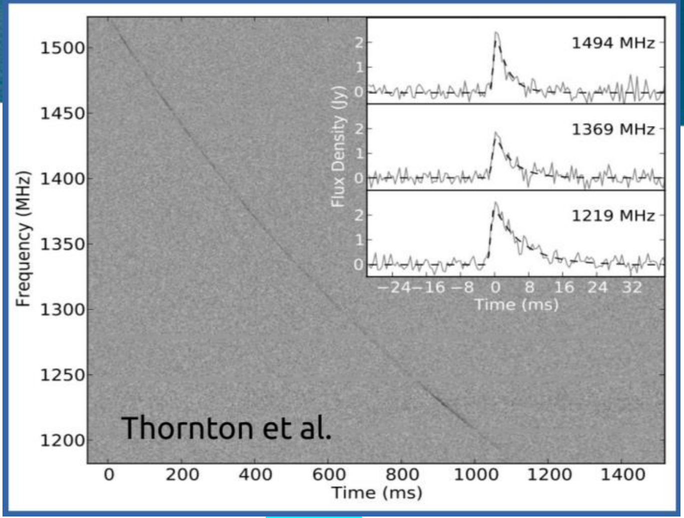
All these additional 'enabling' products need to be identified and included in the PBS, configuration managed and costed.

ECPs (Engineering Change Proposals)

- The system baseline design is not static, but evolving.
- Proposed changes can be raised with an Engineering Change Proposal (ECP).
- This proposal needs to be assessed and an impact analysis done.
- If approved, an implementation plan is written that describes what needs to change in the design and how. It is important that:
  - all affected documents are correctly identified and then updated. Otherwise the design won't be self-consistent after the implementation.
  - the scope of the ECP doesn't change between the assessment and implementation stages, as it is sometimes can do.

(smartsheet.com)

ECPs – example 1



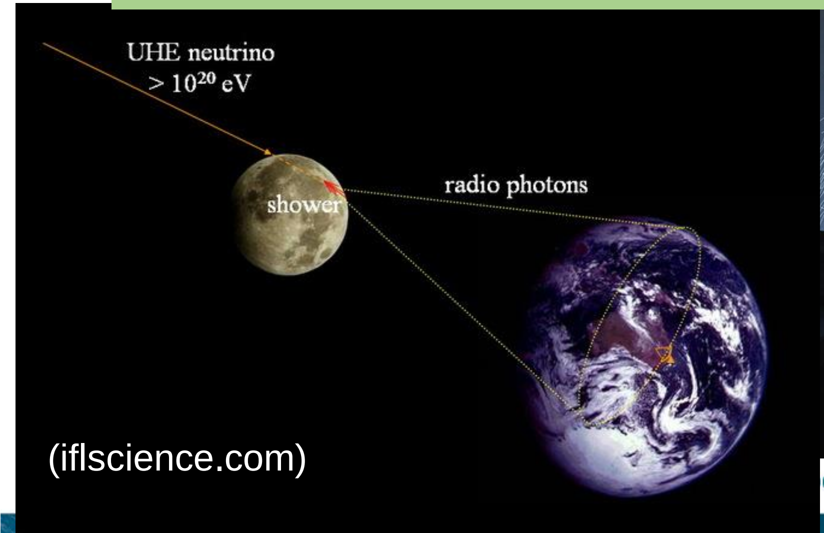
- An ECP was raised that SKA1-Low should be able to study fast transients by buffering several seconds worth of station beam data for all 512 stations that can be triggered when an event is detected.
- This was approved. The effects included:
  - Modifying one Element’s design to implement the buffer itself.
  - Creating a new logical interface to define the offloading of data from the buffer to the science processor.
  - Creating two new physical network interfaces at the point of offload and the point of ingest
  - Modifying the control interface to include the sending of trigger messages and associated data.
  - Defining how the transient buffer will be used in operations (different triggering scenarios, resource availability checking).

ECPs – example 2

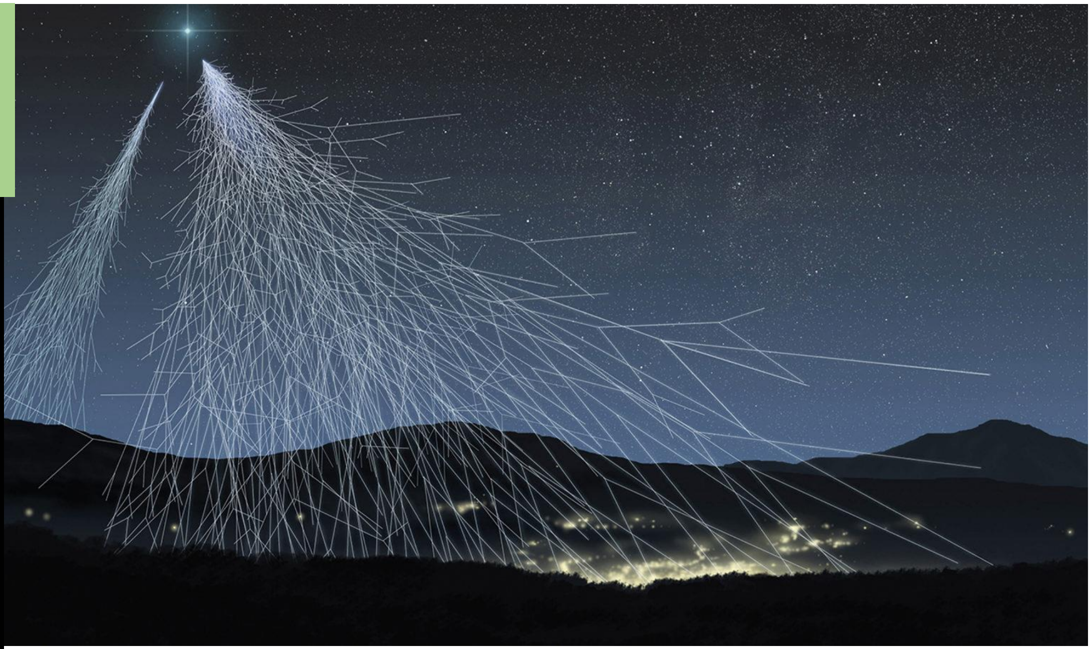
- Several ECPs were raised for different custom experiments to use the SKA for other science cases (i.e. cosmic ray showers, lunar detection of cosmic rays).

- Rather than accommodate each individual custom experiment, it was decided to identify specific changes to the SKA design that would accommodate a broad range of possible custom experiments in the future.

These changes were consolidated in a new set of ECPs.



(iflscience.com)

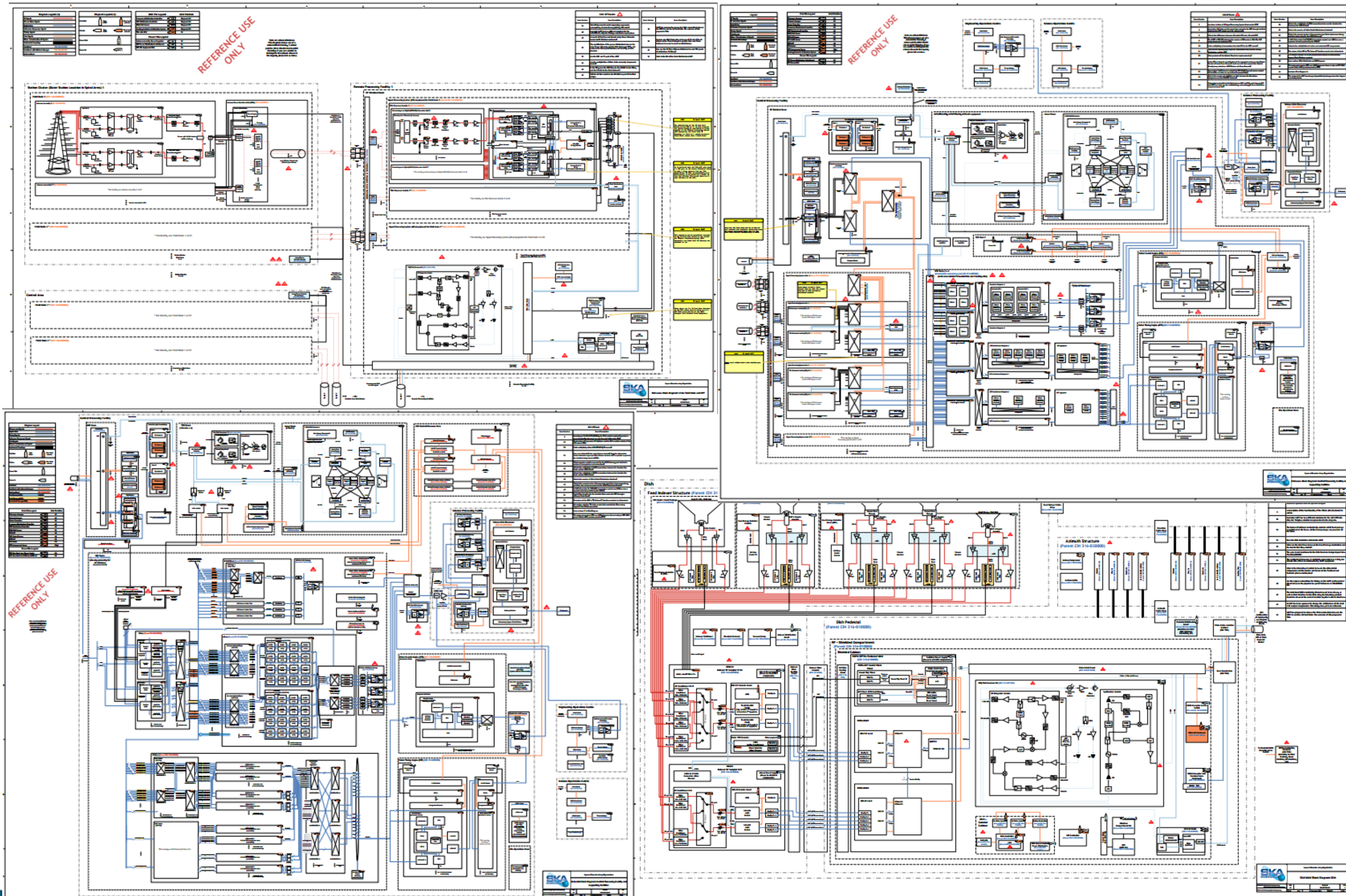


(symmetrymagazine.org)

Integrated design

- The SKAO reviews the individual Element designs, especially at milestones such as Element Preliminary Design Review (PDR), Critical Design Review (CDR).
- But how is the ‘System Design’ reviewed – particularly at System PDR, CDR?
- Activities are performed which try to integrate the designs across all Elements, for a particular view (e.g. hardware, networks, monitoring and control).
- In this way, issues to do with the system design can be identified.

Integrated design – an example

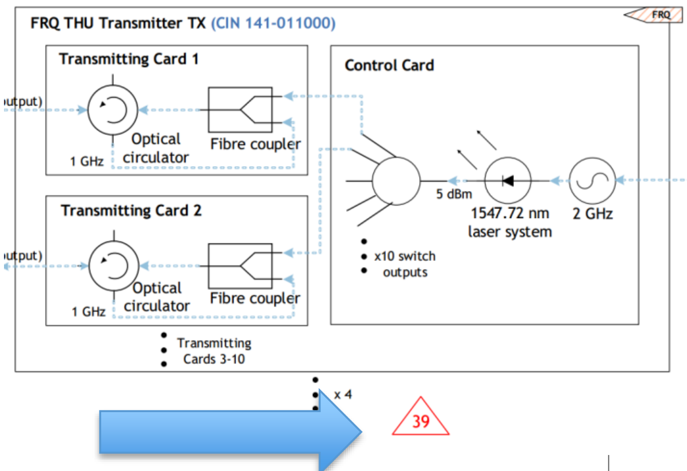


Integrated design – an example

- As the diagrams are created by drawing from multiple design documents, issues or uncertainties are noticed.
- These are identified and recorded for resolution.

Legend	
RF Signals	
RF over fibre Signals	
Digital Data	
Reference Frequency Signals	
Timing Signals	
C&M Signals	
Other / Combination of Signals	
Mechanical Couplings	
C&M Out	Over NSDN / Over local network
C&M In	Over NSDN / Over local network
Power Out	
Power In	

C&M Pin Legend		Grid Position
Telescope Manager	TM	B1
Network Manager	NM	B2
Clocks Controller	CLK	B3
Frequency Distribution Controller	FRQ	B2
UTC Distribution Controller	UTC	B2
DDBH EMS Server	DDBH	C1
CSP LMC Server	CSP	C4
CBF MACE Server	CBF	C2
SDP LMC	SDP	A4
PSS Master	PSS	C4
PST Management Server	PST	D3,4
Monitor, Control & Calibration System	MCCS	C1
Power Pin Legend		
CPF Power Distribution Switchboard	CPF	B2
CBF Power LRU	CBF	C2



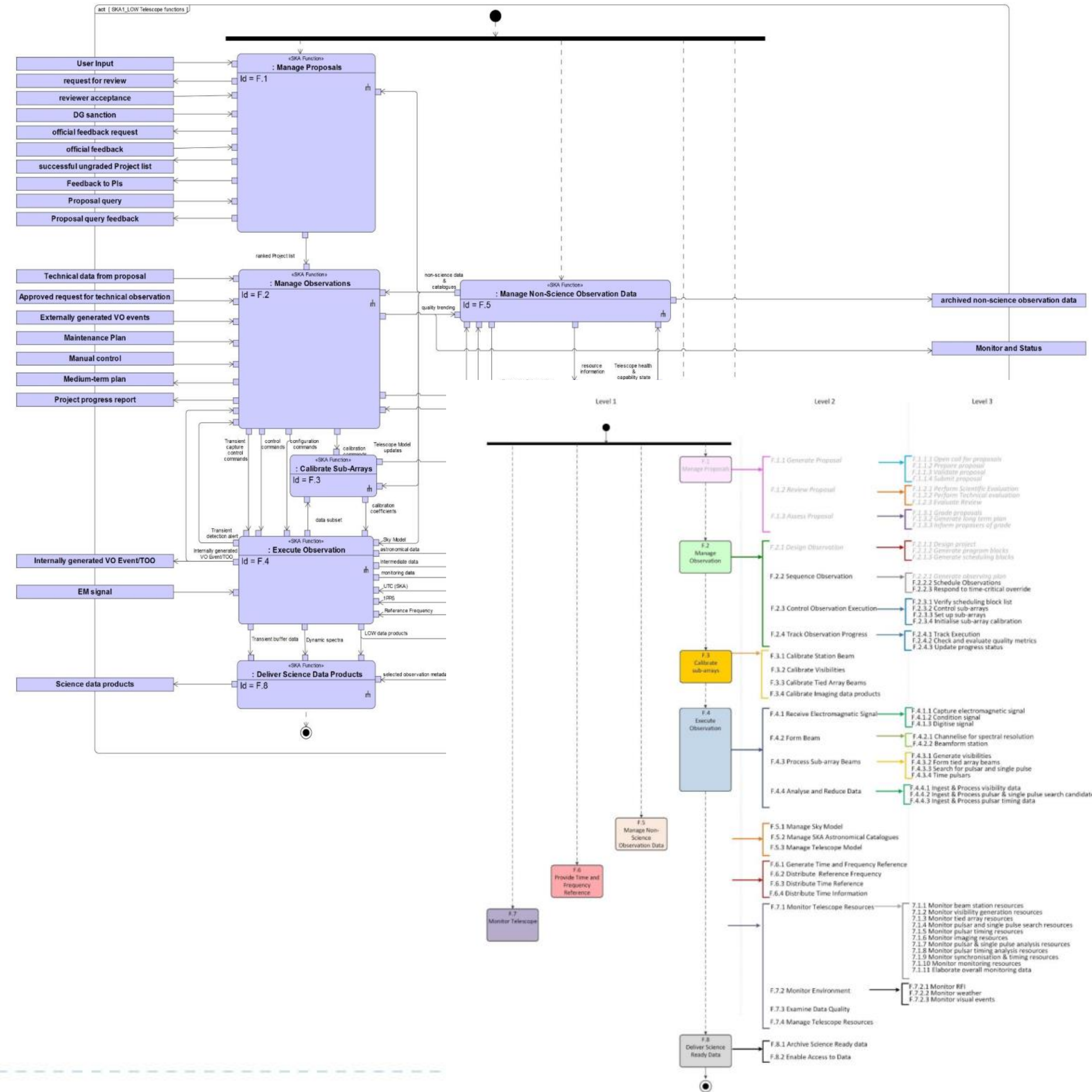
List of Issues

Issue Number	Issue Description
2	Are there 4 Aisles of 19 Signal Processing System Racks in the CPF?
	How is the M&C data and the signal data routed to the RPF through the Data Switch in the CPF (which ports for which data?)
	What is the difference between a Perentie LRU and a Gemini LRU?
	The SADT to CSP ICD indicates the number of PSS nodes is 350. The CBF design document indicates 250.
	Is the multiplicity of connections from the CBF to the PST correct?
	What are the specifications of power distribution between the Primary Distribution and the CPF?
	What product will the Optical Circuit be implemented by?
	In the CBF, what is the specification of the connection between the Optical Circuit and each Cross Connect? And how do 36 fibres (out of the Optical Circuit) carry data from 48 FPGAs to each Cross Connect?
	How do the 2 MACE servers control the CBF LRUs? Can each server control all the LRUs or is there a separation of responsibility?
	What is the number of visibilities per link between the Correlator Subsystems and the CSP-SDP Network?
	What is the bandwidth of the link between CBF and PSS, and between CBF and PST? And what are the multiplicities, and routings, of these links? How many beams per link?

Issue Number	Issue Description
29	What is the multiplicity of NSDN connections between the Junction Box Panel and the NSDN Racks?
31	What is the number of White Rabbit Distribution Switches?
32	What is the correct number of transponders (and their input connections, and their data rates) for the NSDN Gateway?
33	Is M&C information for CSP-SDP transmission carried over NSDN or over a separate network provided by the NRENs?
35	Indicate the multiplicities of active and redundant CBF components.
37	The nature of the SDP to 'To External' Interface needs to be elaborated.
38	It is unclear whether TM is the product that controls the Observatory Support Tools Platform product.
39	Same as Issue 27 in Field Node and RPF Diagram.
43	TM equipment receives PTP over NSDN, however there is only an NTP NSDN server. Which NSDN product provides PTP?
44	See Issue 53 in Diagram 1.
45	The design of the CBF has changed (unpublished) changes since the diagram was last updated.

Integrated design – another example

- A functional architecture was created in a top-down fashion.
- This consists of several layers of de-composed functions, with each layer showing how functions are related through their inputs and outputs).
- It is planned to repeat this exercise in a bottom-up fashion, using the consortia CDR designs as inputs.



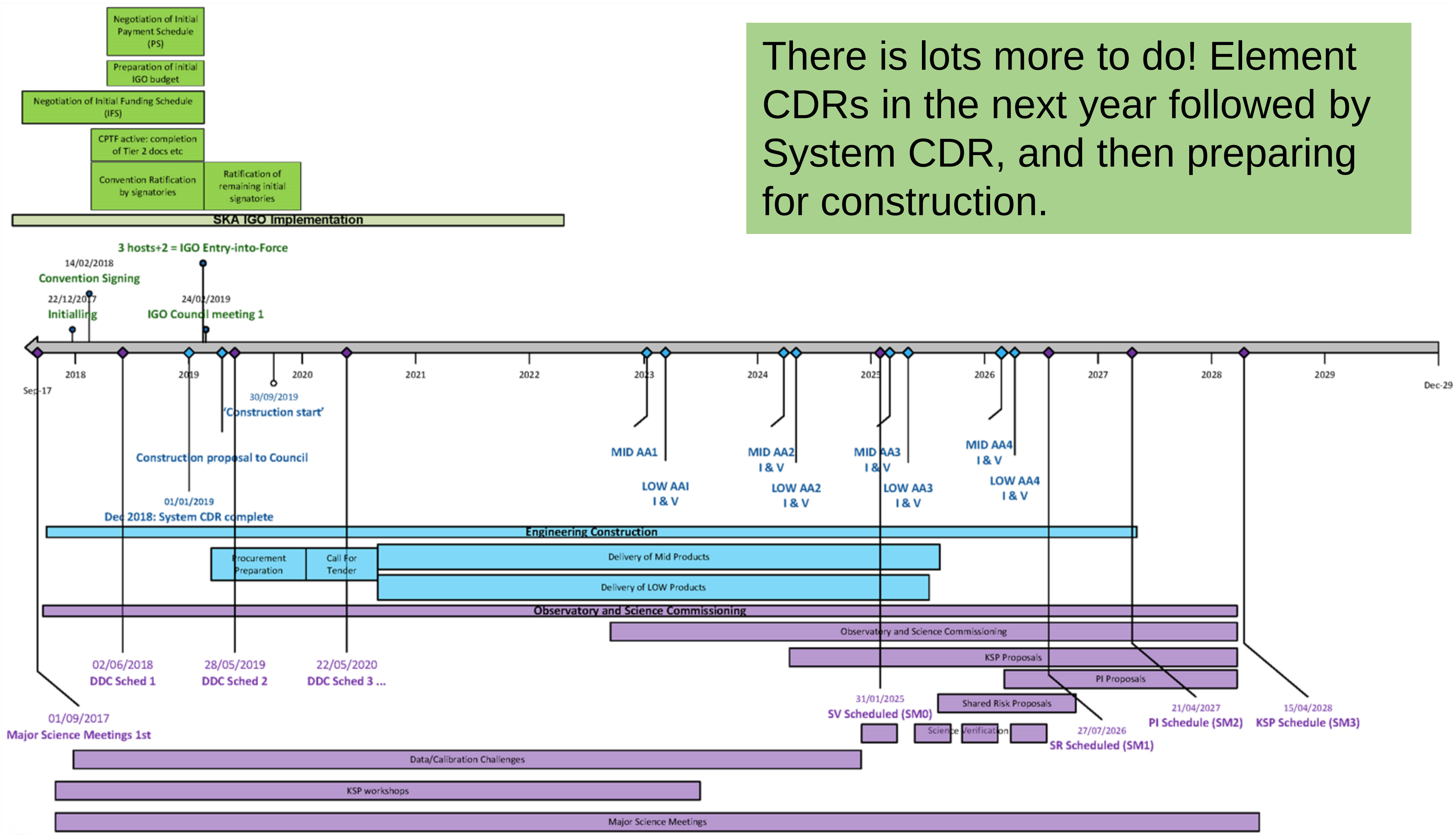
Tools

- Several tools are used for SE related activities:
  - Jama Contour (requirements management)
  - eB (configuration management and PBS management)
  - Confluence (collaboration)
  - Jira (ticket creation and resolution)
  - Cameo Systems Modeller (systems modelling)
  - Visio (diagramming)

(images subject to copyright)

SKA1 Integrated Schedule

There is lots more to do! Element CDRs in the next year followed by System CDR, and then preparing for construction.



SQUARE KILOMETRE ARRAY

Exploring the Universe with the world's largest radio telescope

Thank you!

Questions?