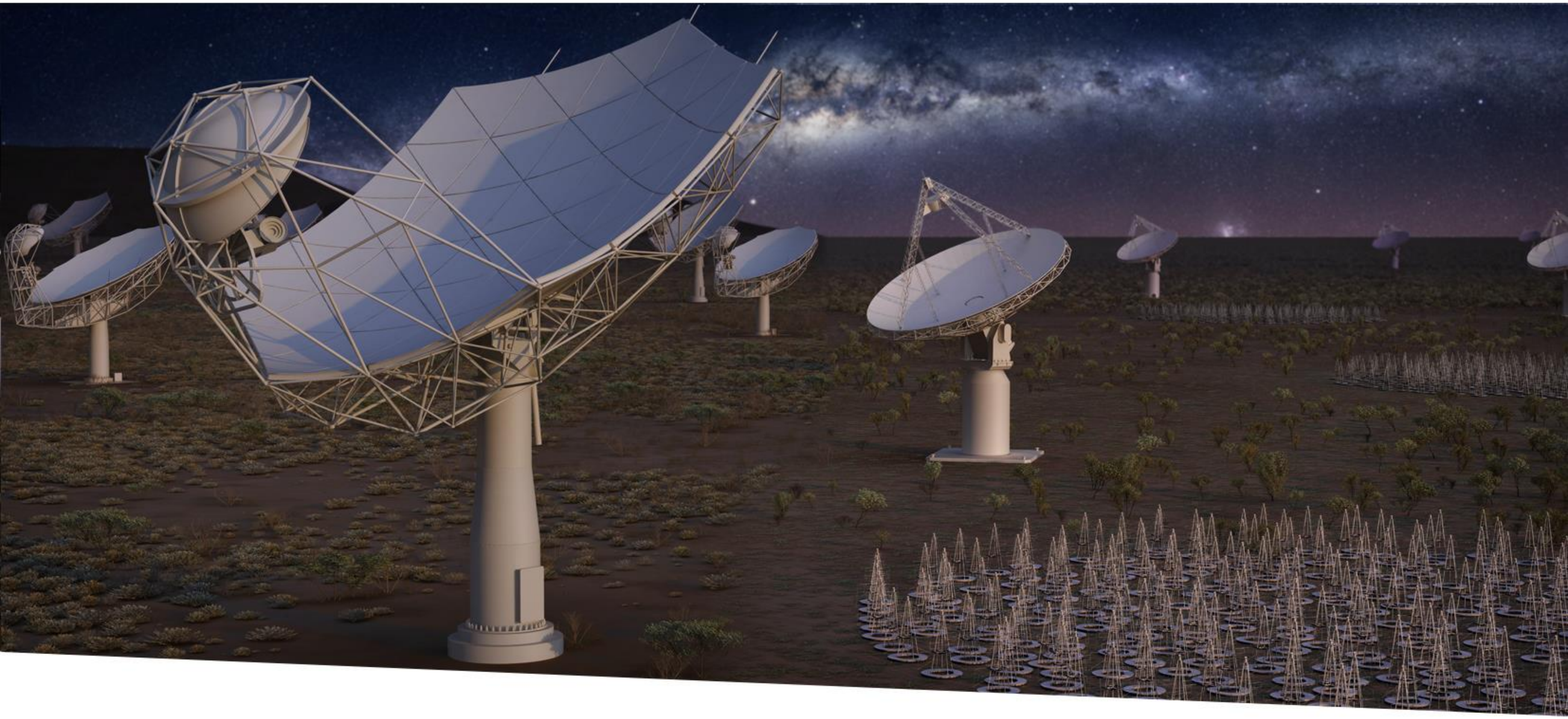
Systems Engineering at SKAO



SQUARE KILOMETRE ARRAY

Exploring the Universe with the world's largest radio telescope

Daniel Hayden

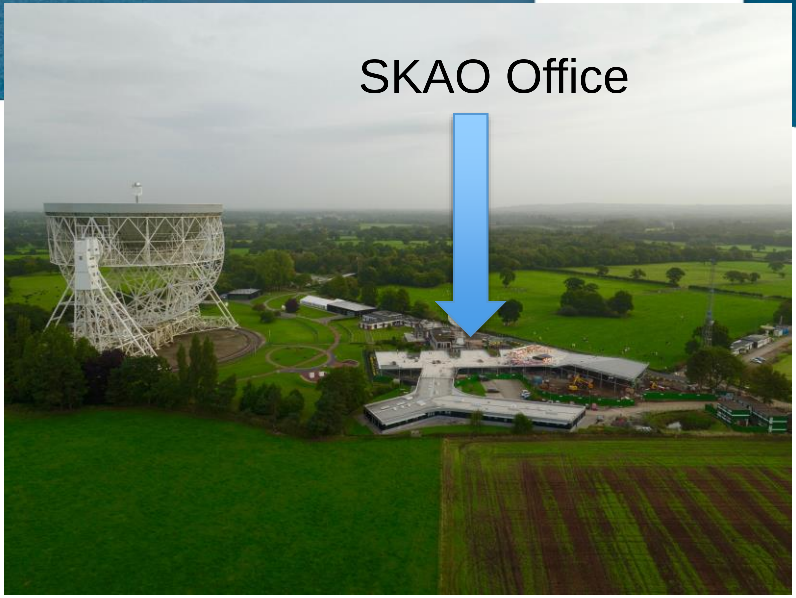
31 May 2019

About me

- I grew up in Johannesburg.
- For my undergrad I studied physics and astronomy at the University of Cape Town (UCT) and the University of South Africa (UNISA).
- I did a Masters in space science and technology in Europe, through a programme called Erasmus Mundus.
- For the last several years I've been working as a Systems Engineer for the SKA project near Manchester, England.

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 at Jodrell Bank near Manchester, England.

(skiddle.com)

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

Jodrell Bank during Bluedot festival

Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

Overview

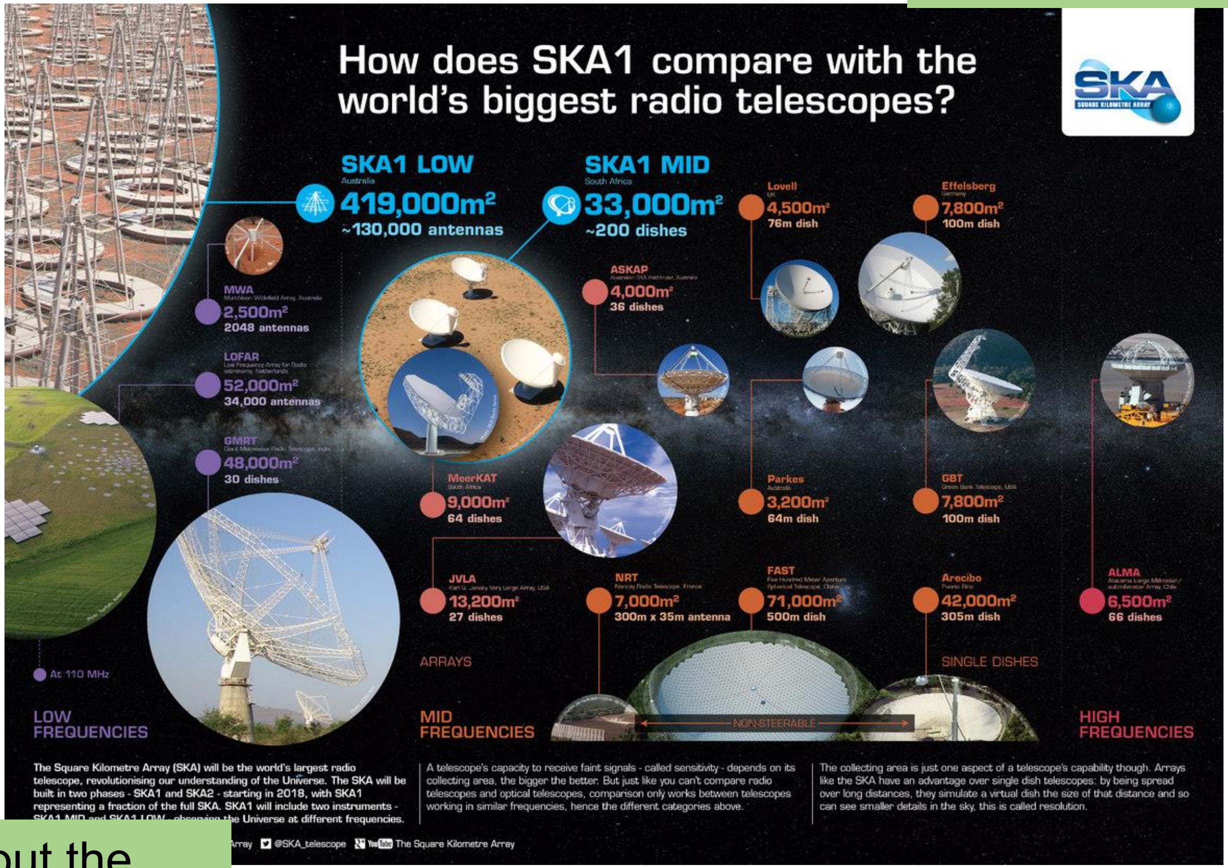
- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

SKA1 – a quick recap

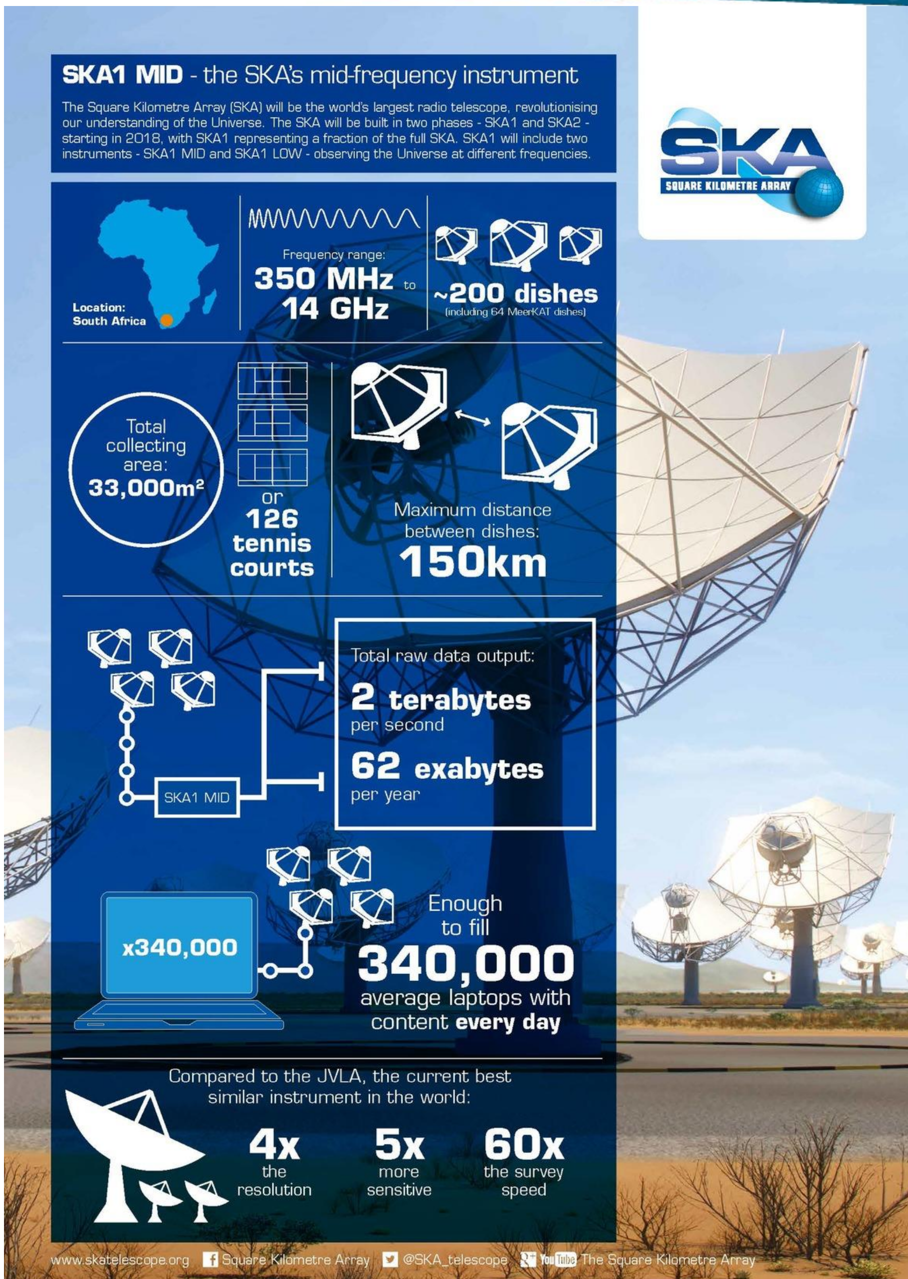
https://www.youtube.com/watch?v=Hog411ZSzEY

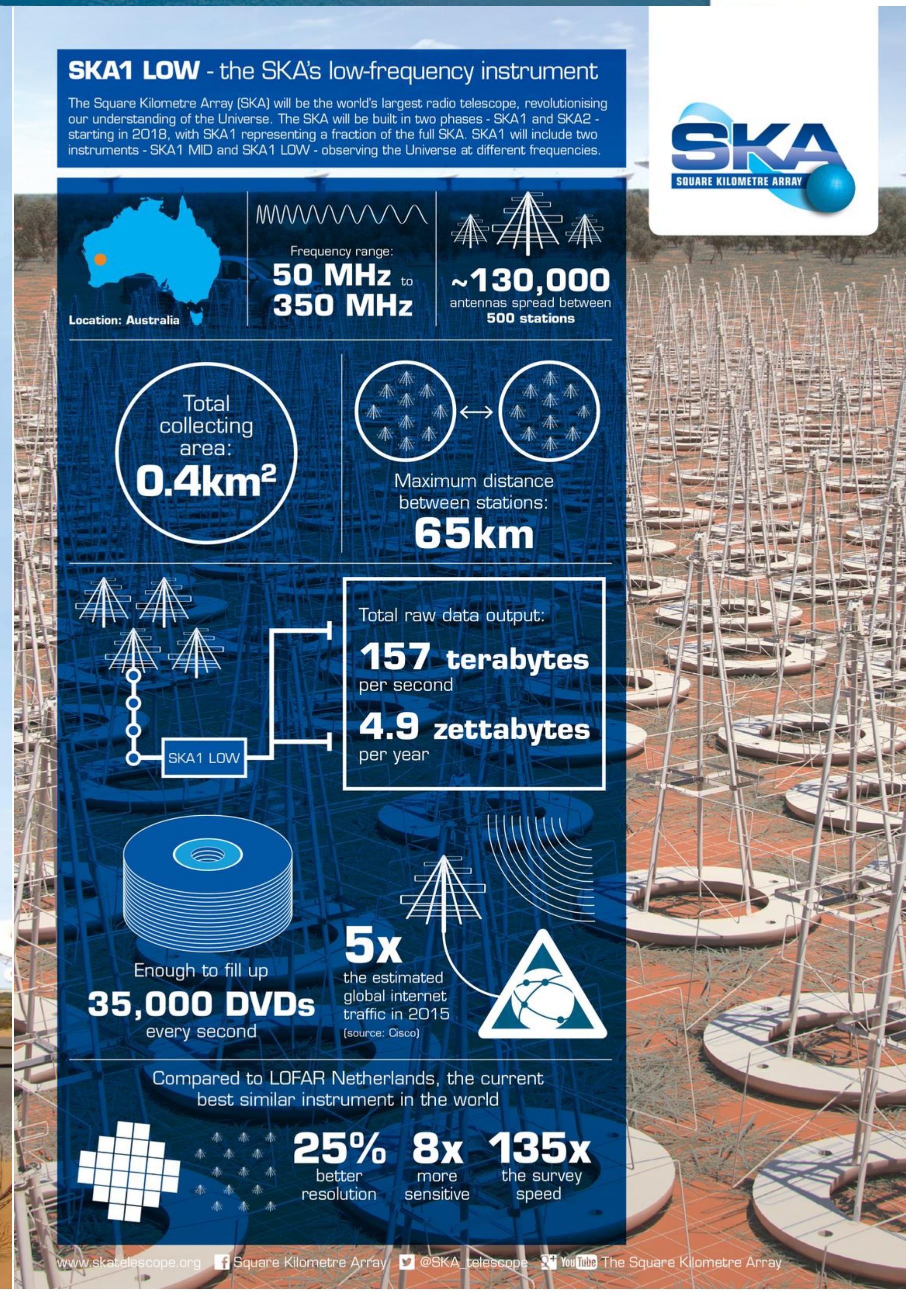
SKA1 - a quick recap

SKA1 MID – S.A
SKA1 LOW – Australia



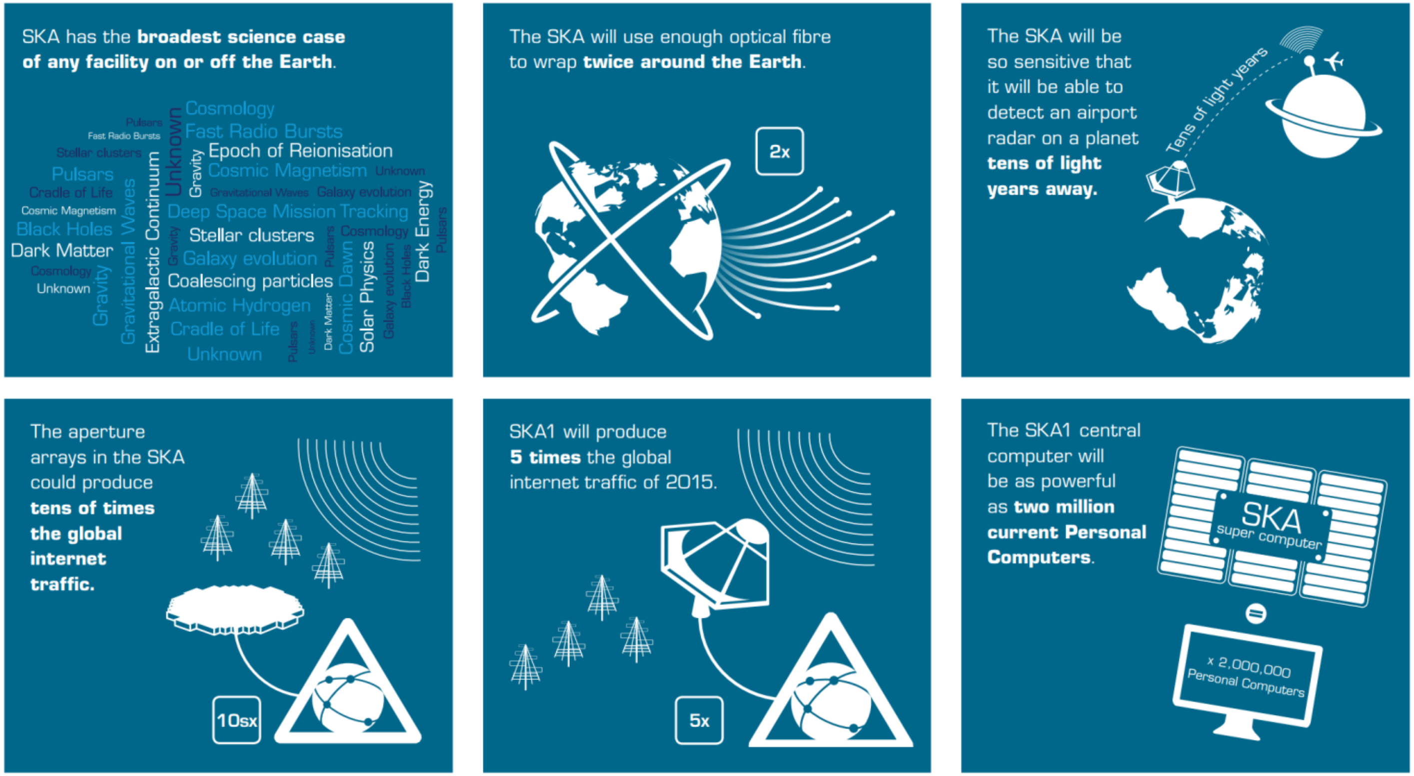
You've heard about the SKA1 in several talks at this training, so here's just a quick recap


SKA1 MID - the SKA's mid-frequency instrument
The Square Kilometre Array (SKA) will be the world's largest radio telescope, revolutionising our understanding of the Universe. The SKA will be built in two phases - SKA1 and SKA2 - starting in 2018, with SKA1 representing a fraction of the full SKA. SKA1 will include two instruments - SKA1 MID and SKA1 LOW - observing the Universe at different frequencies.
SKA
SQUARE KILOMETRE ARRAY
Location: South Africa
Frequency range:
350 MHz to 14 GHz
~200 dishes
(including 64 MeerKAT dishes)
Total collecting area: 33,000m²
or 126 tennis courts
Maximum distance between dishes:
150km
Total raw data output:
2 terabytes per second
62 exabytes per year
SKA1 MID
x340,000
Enough to fill
340,000
average laptops with content every day
Compared to the JVLA, the current best similar instrument in the world:
4x the resolution
5x more sensitive
60x the survey speed
www.skatelescope.org
Square Kilometre Array
@SKA_telescope
The Square Kilometre Array


SKA1 LOW - the SKA's low-frequency instrument
The Square Kilometre Array (SKA) will be the world's largest radio telescope, revolutionising our understanding of the Universe. The SKA will be built in two phases - SKA1 and SKA2 - starting in 2018, with SKA1 representing a fraction of the full SKA. SKA1 will include two instruments - SKA1 MID and SKA1 LOW - observing the Universe at different frequencies.
SKA
SQUARE KILOMETRE ARRAY
Location: Australia
Frequency range:
50 MHz to 350 MHz
~130,000
antennas spread between 500 stations
Total collecting area:
0.4km²
Maximum distance between stations:
65km
Total raw data output:
157 terabytes per second
4.9 zettabytes per year
SKA1 LOW
Enough to fill up
35,000 DVDs
every second
5x
the estimated global internet traffic in 2015
(source: Cisco)
Compared to LOFAR Netherlands, the current best similar instrument in the world
25% better resolution
8x more sensitive
135x the survey speed
www.skatelescope.org
Square Kilometre Array
@SKA_telescope
The Square Kilometre Array

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

SKA1-MID

~2 Pb/ s → LOW-FREQUENCY APERTURE ARRAY

7.2 Tb/ s

8.8 Tb/ s → SIGNAL AND DATA TRANSPORT

CSP — CENTRAL SIGNAL PROCESSOR — ~50 PFlops

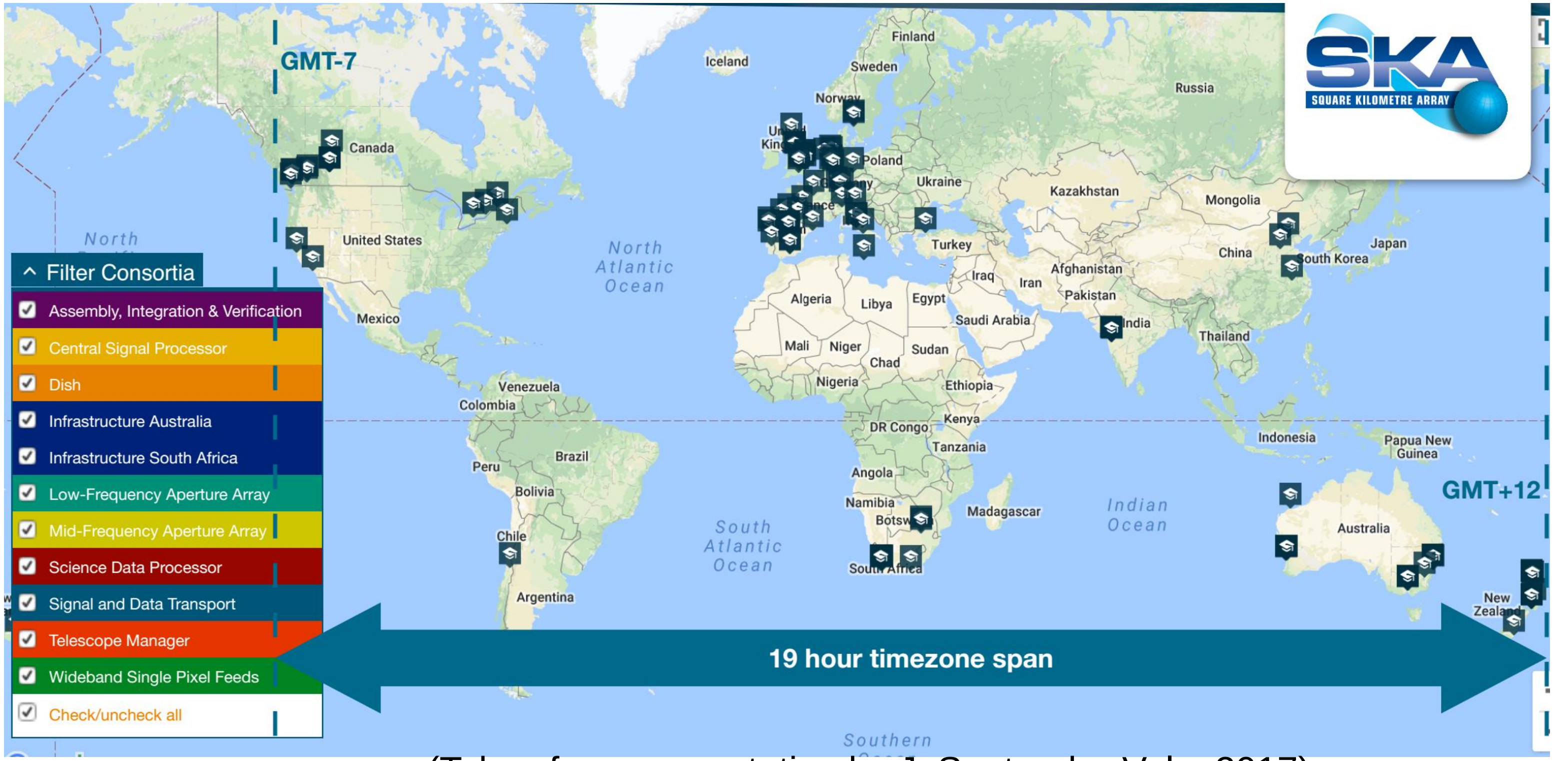
~5 Tb/ s

SDP — SCIENCE DATA PROCESSOR — ~250 PFlops

~300 PB/ yr

SKA1 – 2 types of challenge

- **The organisational challenge** – The SKA is an international project, currently with **13** member 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)

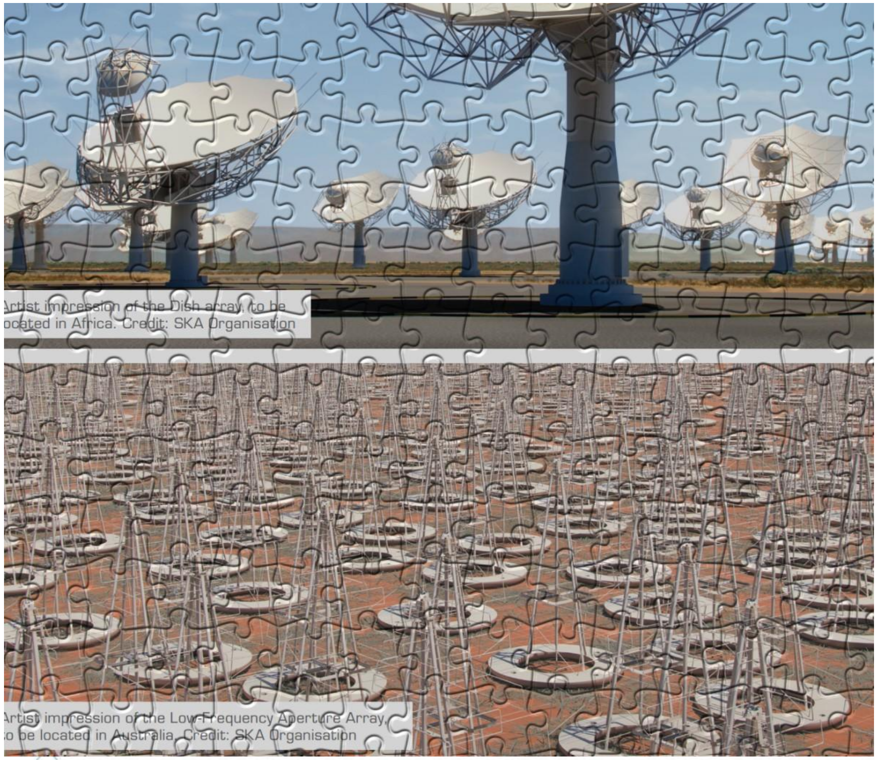
Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

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!

QUESTION TIME:

Discuss for 2 min:

What presents a bigger challenge to a large design project – technical complexity or organisational complexity?

The need for systems engineering

- In addition to designing the detailed parts of the telescope, you also need a system view 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**?

- Rule of thumb: if one person can't hold a view of the whole system in their mind, you need Systems Engineering! **Corollary:** No one person has the whole view of the system in their mind. So where does this view exist?

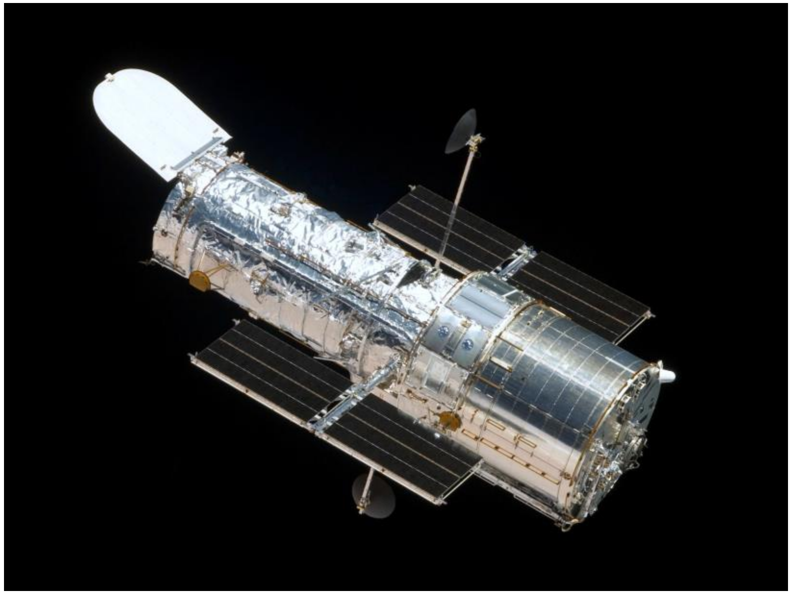
- The system view needs to be created and managed using formal techniques. This is what systems engineering does.

(https://blog.prototypr.io/)

The need for systems engineering

What if you don't do systems engineering?

- **A lesson from Hubble**. A review found that "People working on the design of the solar arrays were not coordinating with people working on 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."

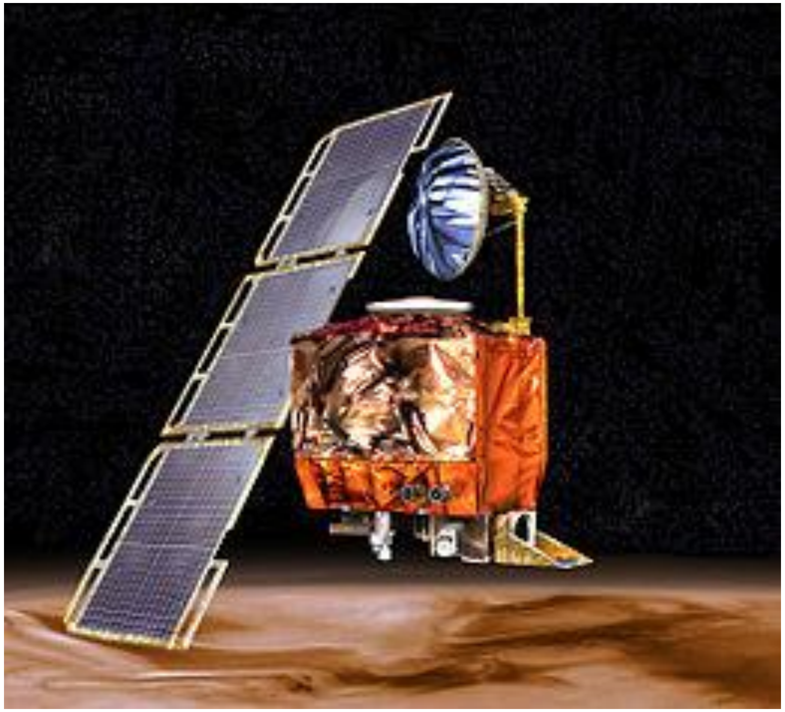


(NJ Slegers, 2012 – inspired by a presentation by Simon Wright)

The need for systems engineering

What if you don't do systems engineering?

- The Mars Climate Orbiter was a robotic space probe launched by NASA on December 11, 1998.
- Mission was to study the Martian climate, Martian atmosphere, and surface changes.
- However, on September 23, 1999, communication with the spacecraft was lost as it passed too close to the planet.
- Why? The ground-based computer software produced output in non-SI units instead of the SI units specified in the contract between NASA and Lockheed. This resulted in a miscalculation by the trajectory calculation software.



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)

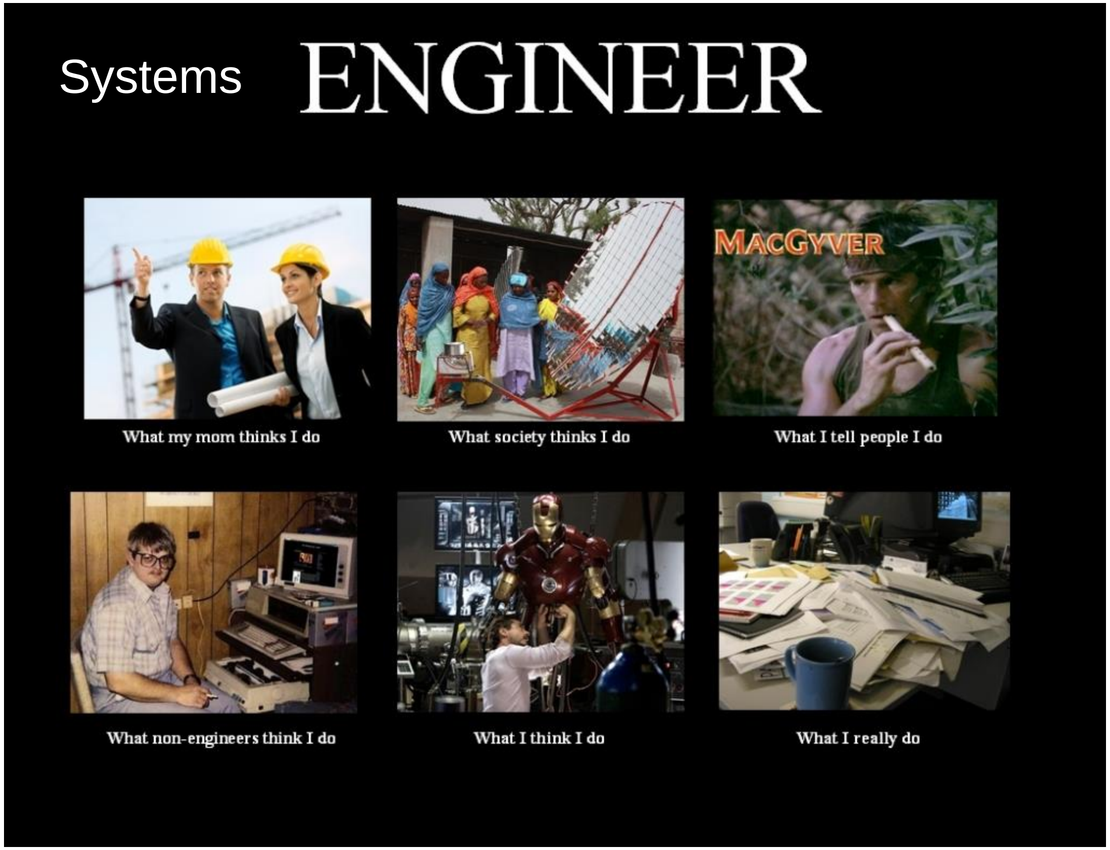
This is one of several possible definitions. But one that I like.

QUESTION TIME:

Discuss for 2 min:

When building large systems, what are two reasons that systems engineering is more necessary today than it was in the past?

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



(whatmyfriendsthinkido.net)

Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

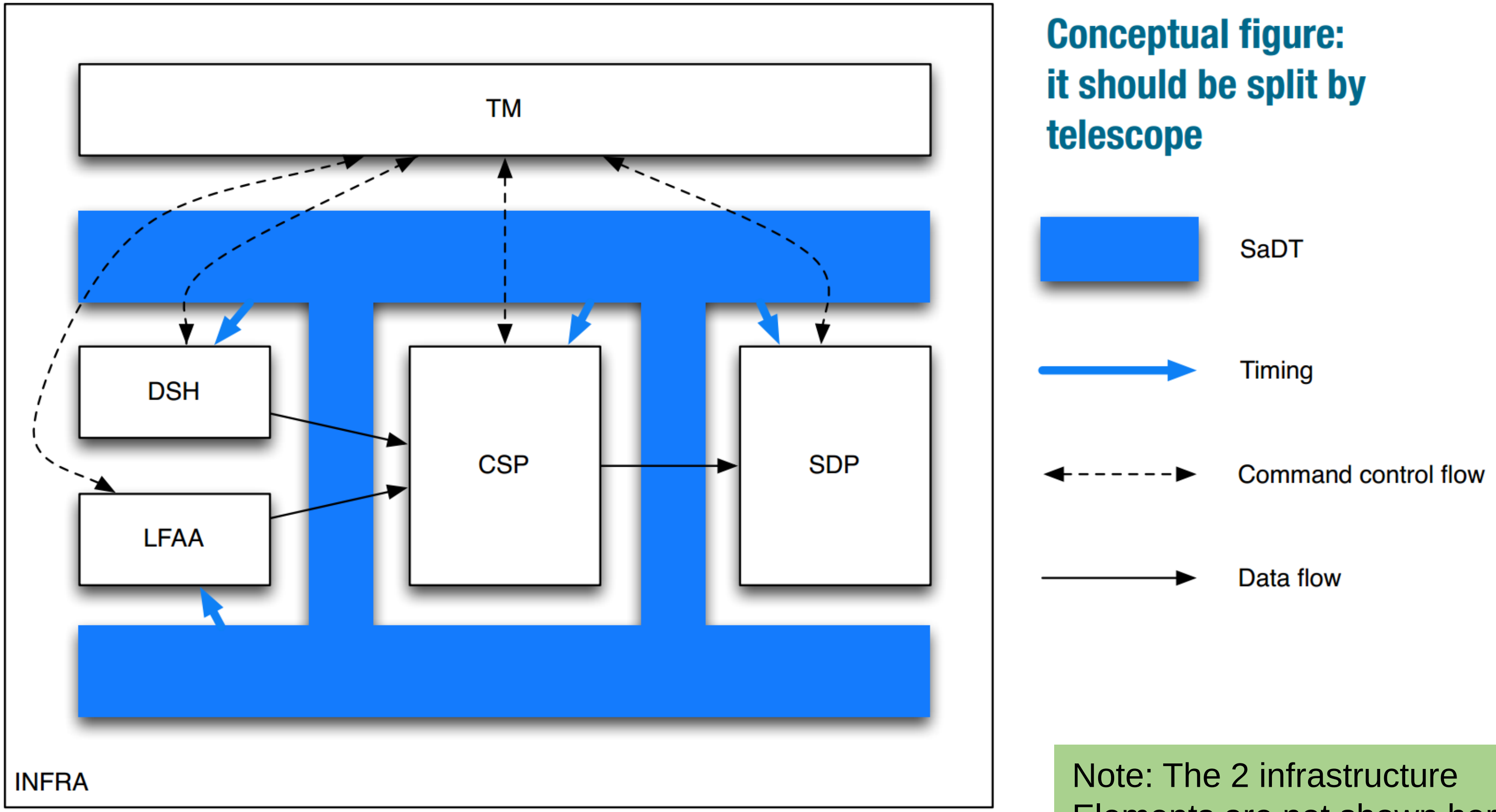
Interfaces

- During the design phase of the SKA, 8 design consortia were each responsible for delivering the design of one sub-system or 'Element'.
- Therefore the system was broken up into these different Elements.
- Between these Elements, interfaces exist that need careful definition and management.

SKA Elements

- DSH – Dish
- LFAA – Low Frequency Aperture Array
- SADT – Signal and Data Transport
- TM – Telescope Manager
- CSP – Central Signal Processor
- SDP – Since Data Processor
- INAU – Infrastructure Australia
- INSA – Infrastructure South Africa

SKA Elements



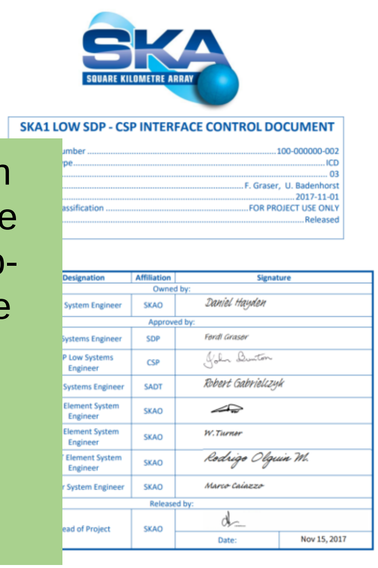
(Taken from presentation by J. Santander-Vela, 2014)

Conceptual figure: it should be split by telescope

Note: The 2 infrastructure Elements are not shown here

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

...umber	100-000000-002
...pe	ICD
	03
	F. Graser, U. Badenhorst
	2017-11-01
...assification	FOR PROJECT USE ONLY
	Released

Designation	Affiliation	Signature
Owned by:		
System Engineer	SKAO	Daniel Hayden
Approved by:		
...ystems Engineer	SDP	Ferdl Graser
...P Low Systems Engineer	CSP	
...Systems Engineer	SADT	Robert Gabrielczyk
Element System Engineer	SKAO	
Element System Engineer	SKAO	W. Turner
Element System Engineer	SKAO	Rodrigo Olguin M.
...System Engineer	SKAO	Marco Caiazzo
Released by:		
...ead of Project	SKAO	Date: Nov 15, 2017

To define an interface in an ICD, you need to define the characteristics of each sub-system at the interface, the media involved in the interaction, and the characteristics of the thing crossing the interface
(Wheatcraft, 2010)

- Because an interface is a boundary between two sub-systems, its definition depends on how your system is divided.
- This N Squared diagram shows all the External interfaces between the Elements. The SKAO was responsible for these.
- Each of these interfaces is defined and detailed in an Interface Control Document (ICD).
- Since each Element consists of many sub-systems, there are also multiple Internal interfaces for each Element. The consortia were responsible for these.

Interfaces – external and internal

- The distinction between External and Internal interfaces is based on organisational boundaries during the design phase.

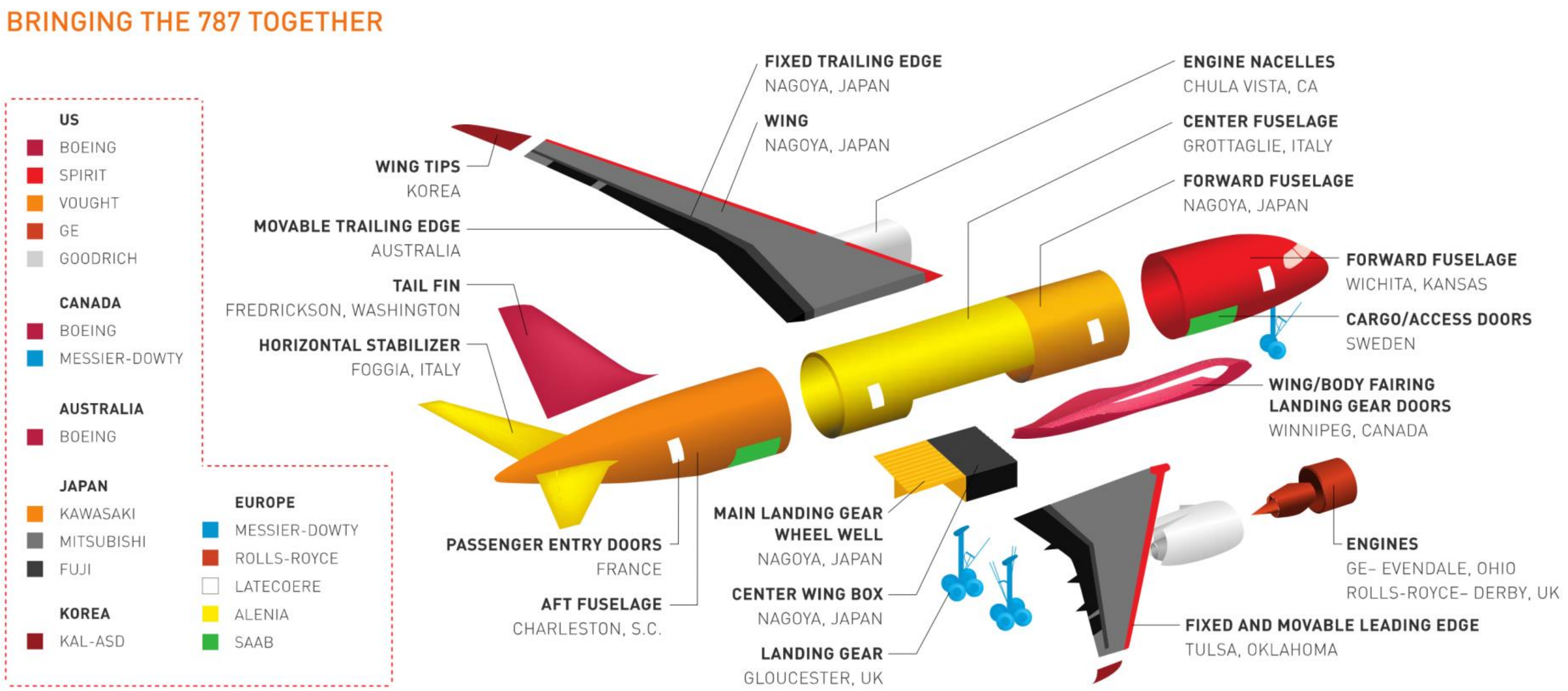
- For the construction phase, the boundaries might be different. In this case, the content of the external and internal ICDs will need to be re-packaged along these new boundaries.

- The distinction between External and Internal interfaces will fall away during construction.

(fbaforward.com)

The moral: always remember 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.

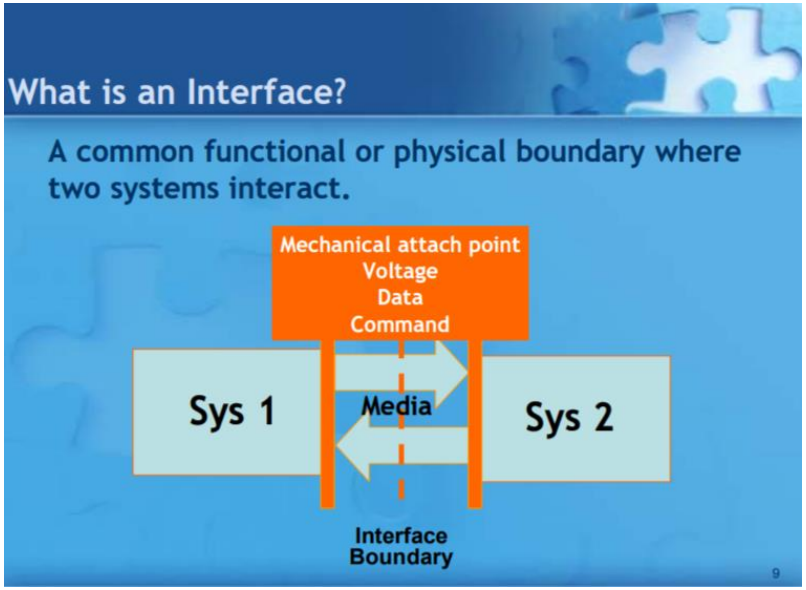
Interfaces – another example



(Fundamentals of Systems Engineering – MIT Open Courseware)

Interfaces – so what are they?

- System 1 and system 2 might be self-consistent, but they must also 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).

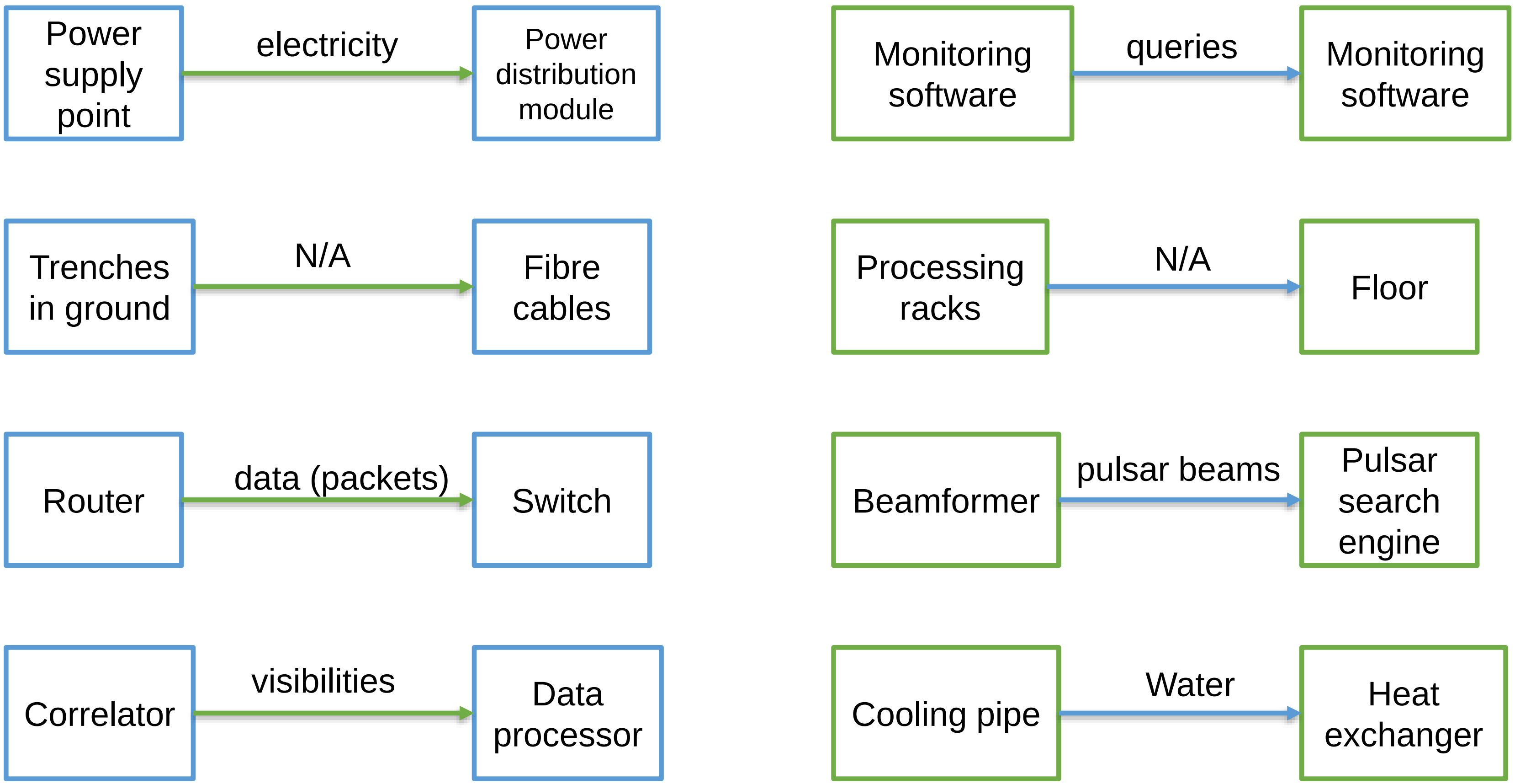


(Wheatcraft, 2010)

- An interface is not a thing in itself. It is a “boundary where, or across which, two or more parts interact.” (Wheatcraft, 2010).

Interfaces – so what are they?

The SKA has many kinds of interfaces. Some examples are:



QUESTION TIME:

Turn to your partner and discuss for 2 min:

Think of an example from your life when you needed to define and manage an interface. Think out of the box!

QUESTION TIME:

Turn to your partner and discuss for 2 min:

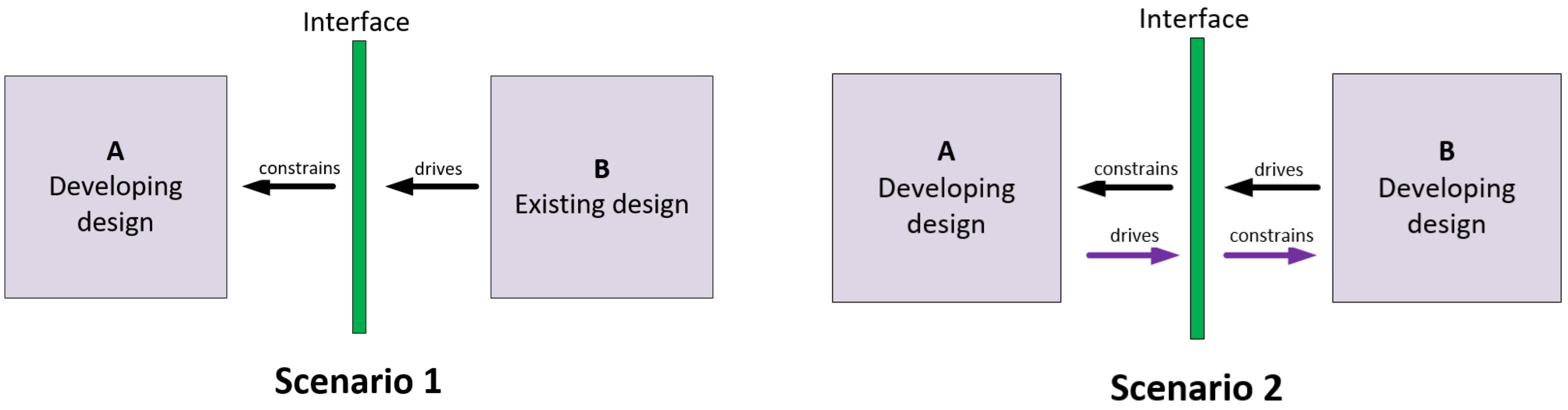
If an Interface Control Document (ICD) defines the interface between two sub-systems, can you think of 2 occasions when this document might be used in the life-cycle of a project?

Interfaces – process challenges

There are several ‘process’ challenges to do with developing ICDs. Such as…

(ewocnj.org)

CHALLENGE 1



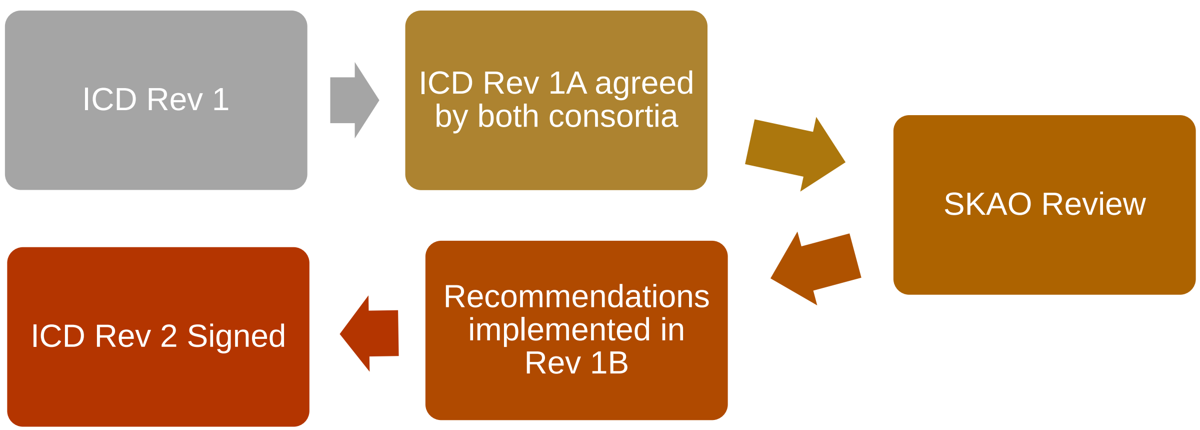
Scenario 1: System A (in development) has an interface with system B (already exists). ‘B’ drives the definition of the interface which constrains the design of ‘A’.

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 has to evolve iteratively.

CHALLENGE 2

- The definition of an interface may have changed a while before this change is formally captured in an ICD.
- Although the design of an interface is always changing, it has to be frozen and reviewed at various times to provide a stable baseline that can be referenced elsewhere in the design. The baselining process can lag behind the actual state of the design.

Process description

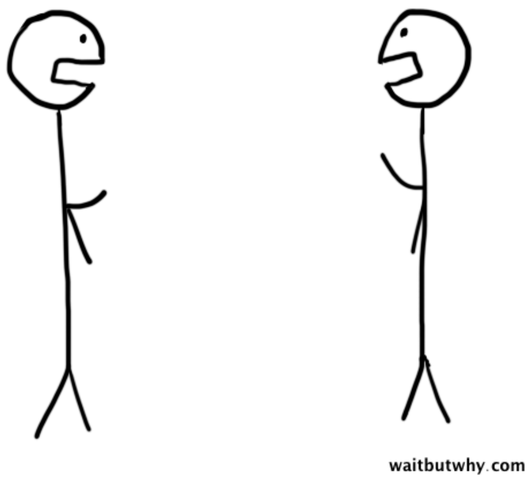


An aside - baselines and the meaning of a ‘signature’

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

Yes, but at least it’s baselined!

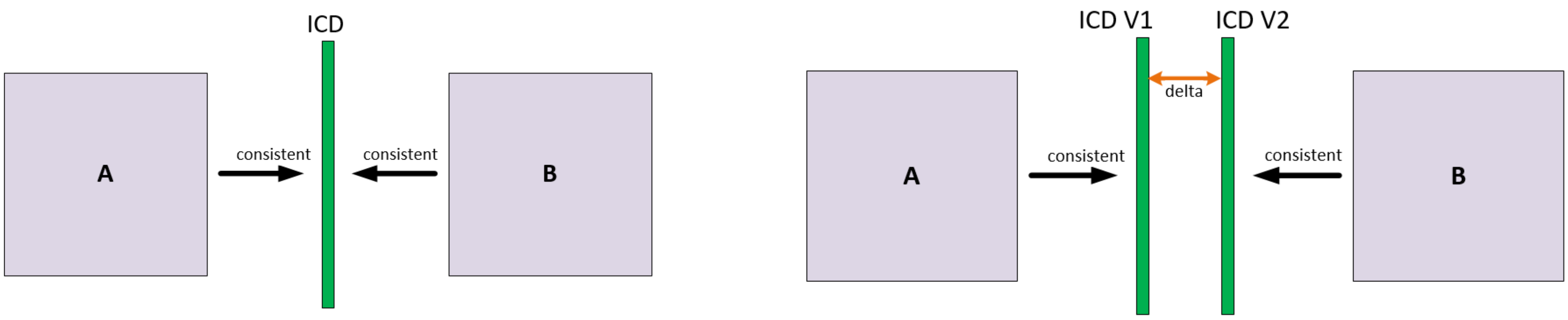
- Imagine design **A** refers to design **B** which refers to design **C**.
- But these designs are always changing, so which version of ‘A’ and ‘B’ and ‘C’?
- Answer: ones that have been agreed and frozen, even if they are not the most correct and current.



Often this is the meaning of a document signature. It doesn’t mean the document is 100% correct, but rather that it has been agreed and baselined.

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 consistent with each other.



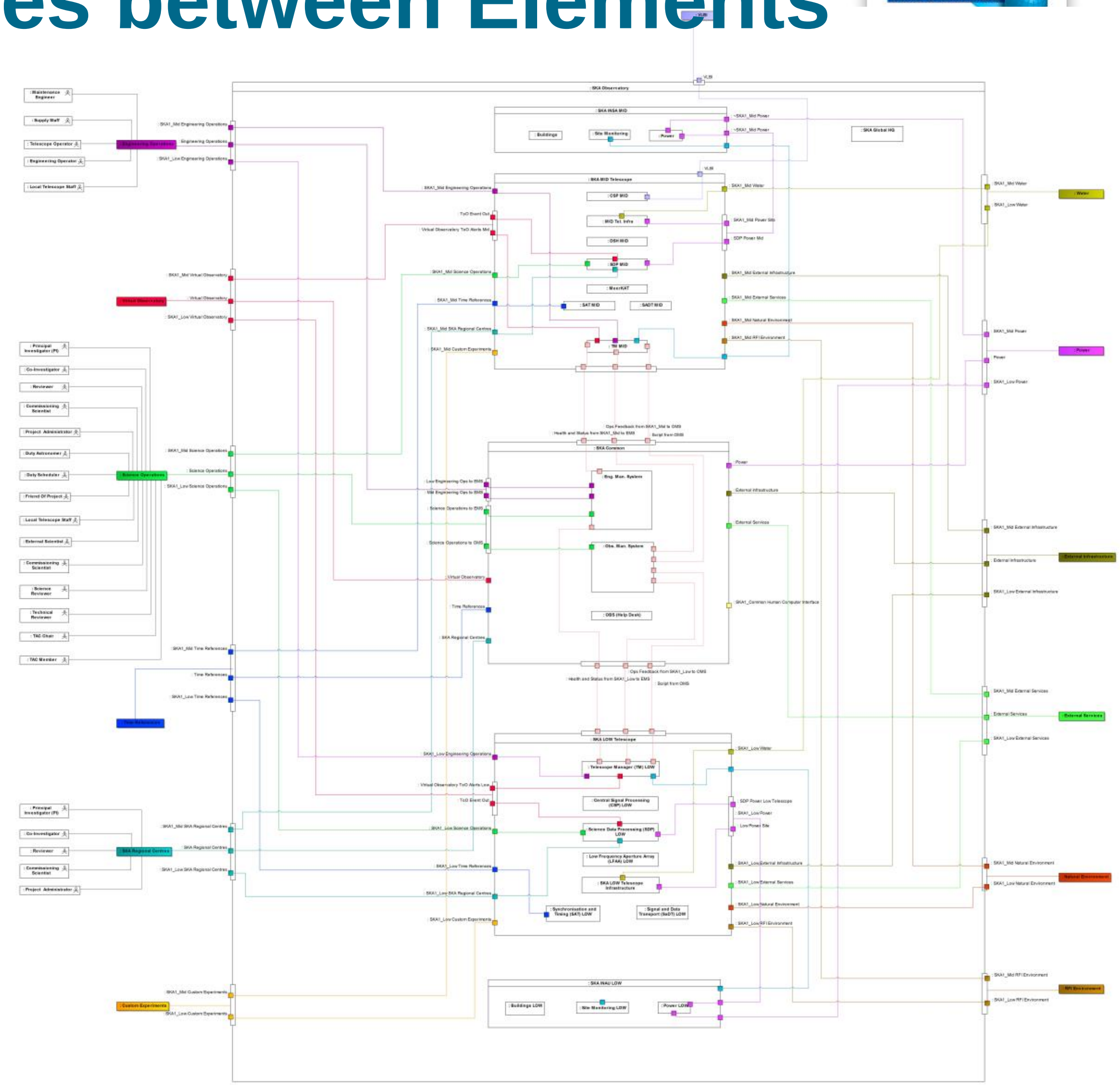
Scenario 1   Scenario 2

Some 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 of data 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 the telescopes and systems that are common to both Telescopes (e.g. Engineering and Observation Management Systems).
  - ➢ Between the telescopes and systems external to SKA.

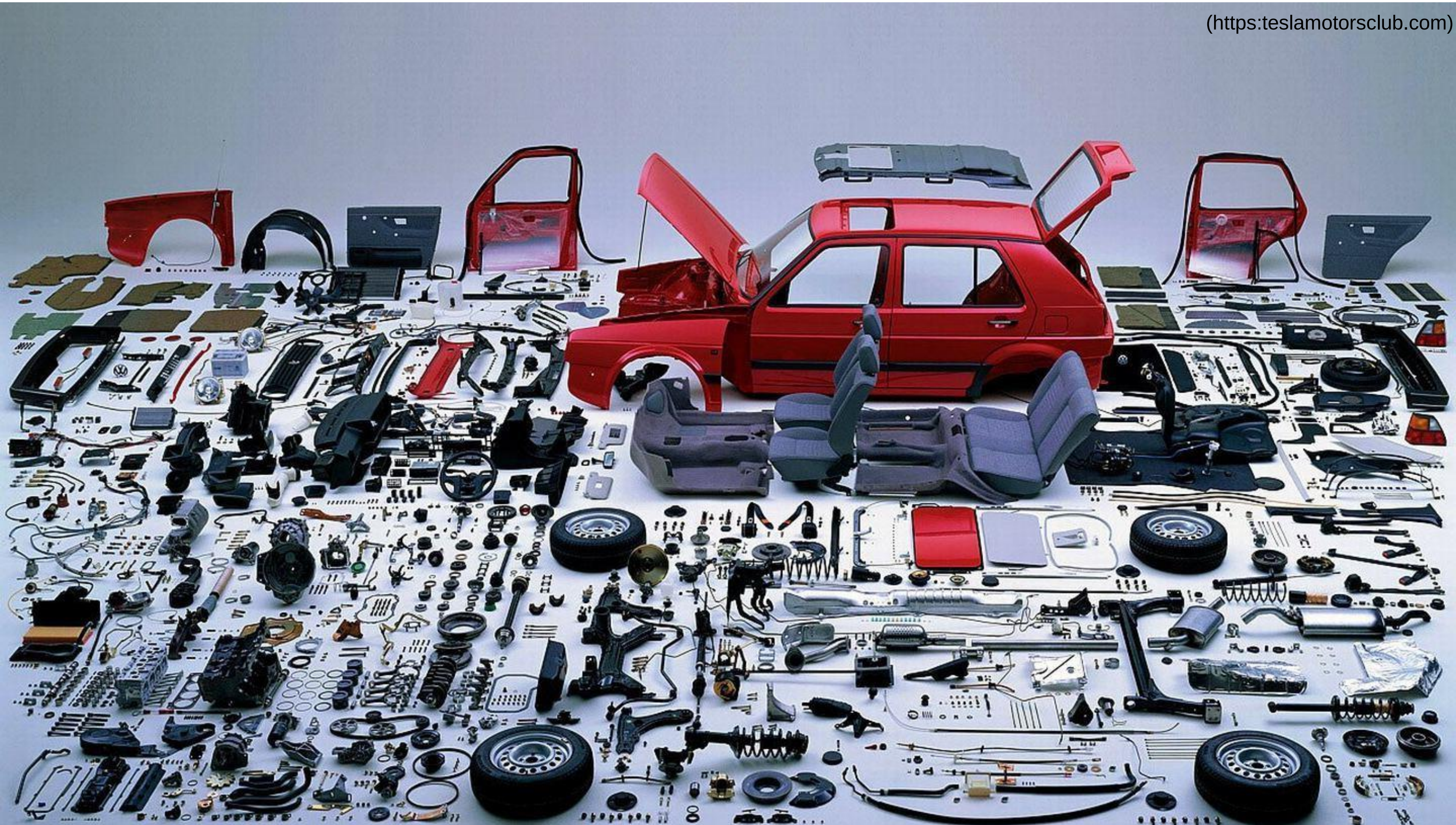


All these interfaces need to be identified and managed!

Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

Product breakdown structures – how to split up a system



(https:teslamotorsclub.com)

Product breakdown structures

(springbok-puzzles.com)

- You may have wondered by now – what determines how the SKA is divided into its 8 Elements?
- There are many ways to split up a system.
- A system can be decomposed along:
  - Functional boundaries
  - Physical boundaries (LRUs)
  - Organisational boundaries
  - Contractual boundaries
- Sometimes it is necessary to use different 'product breakdown structures (PBS) ' and be able to translate from one to the other.

Product breakdown structures

Something to be careful of...

As I said, a system is sometimes decomposed along organisational boundaries. But if you're not careful, this can lead to…

Conway's law!

> organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.
> — *M. Conway*[2]

QUESTION TIME:

Turn to your partner and discuss for 2 min:

Can you think of an example of where Conway’s law could come into effect?

Product breakdown structures

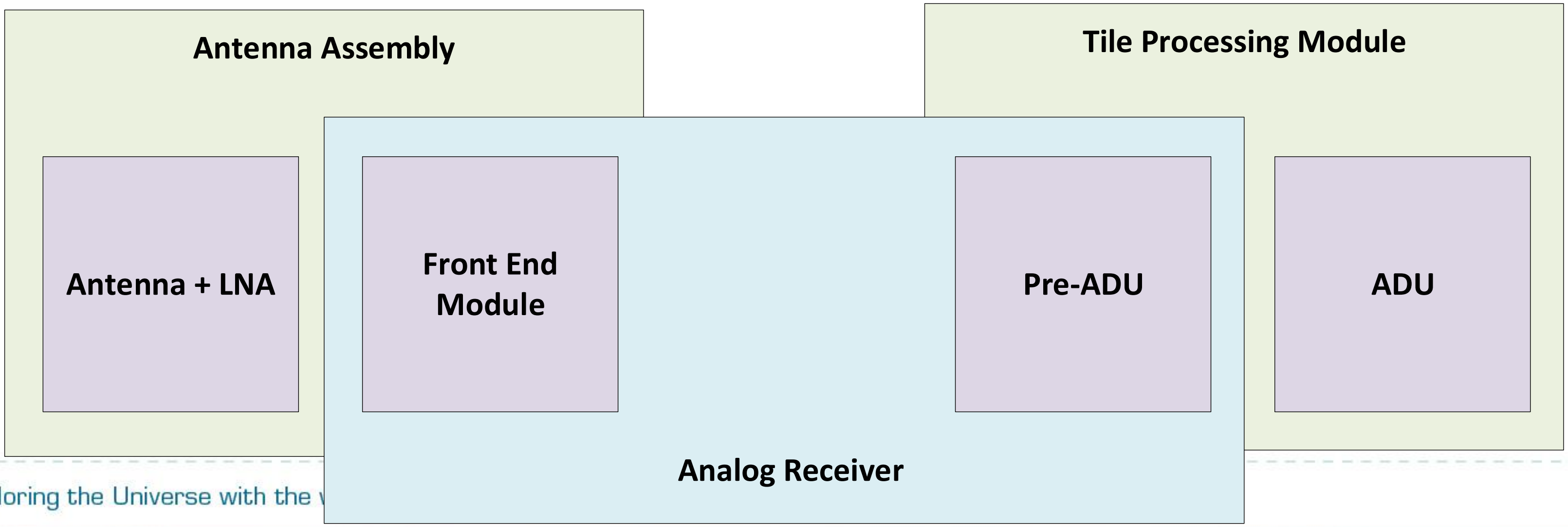
- The SKA has a single Product Breakdown Structure (PBS) which is along physical boundaries. These products will be grouped into ‘work packages’ which will be contracted out for construction. Therefore work packages can’t cut across product divisions.

- Some considerations that determine how to divide up your products:
  – What products make the most sense to design as separate units?
  – What products do you want to test as separate units against separate sets of requirements?
  – What division of products minimises the complexity of interfaces?

- But deciding on a single, optimal physical PBS is not easy. Here’s an example… see next page.

Product breakdown structures

- For SKA1-LOW, the design of the Tile 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.
- But the design of the Pre-ADU is closely 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?



Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

Requirements

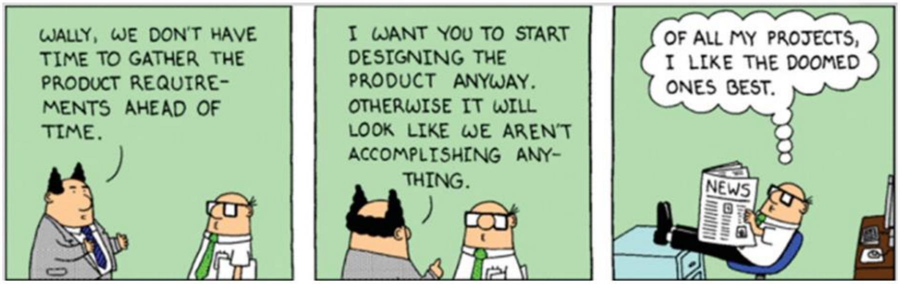
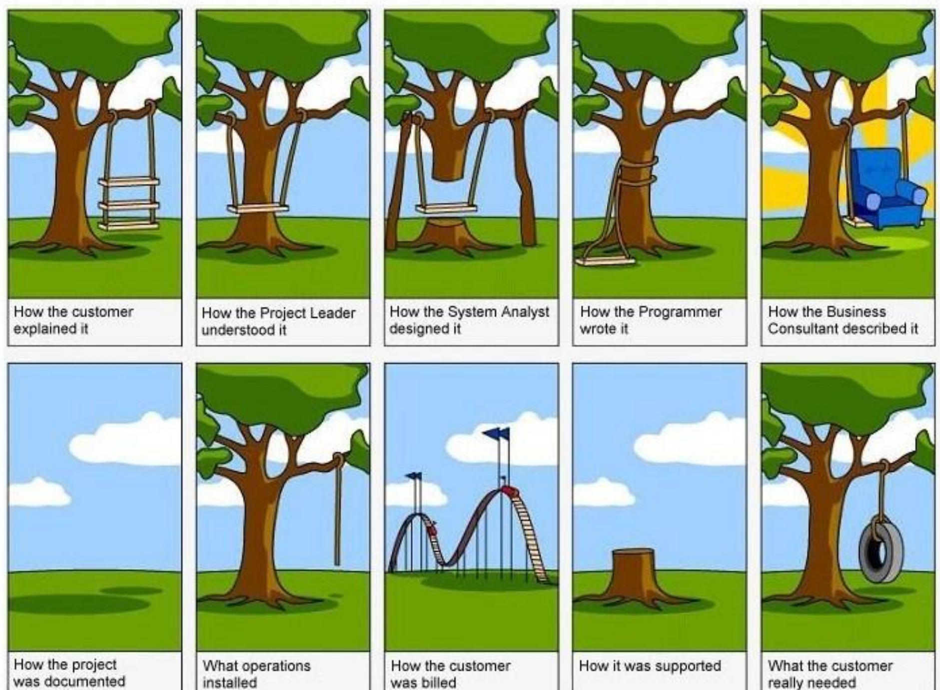


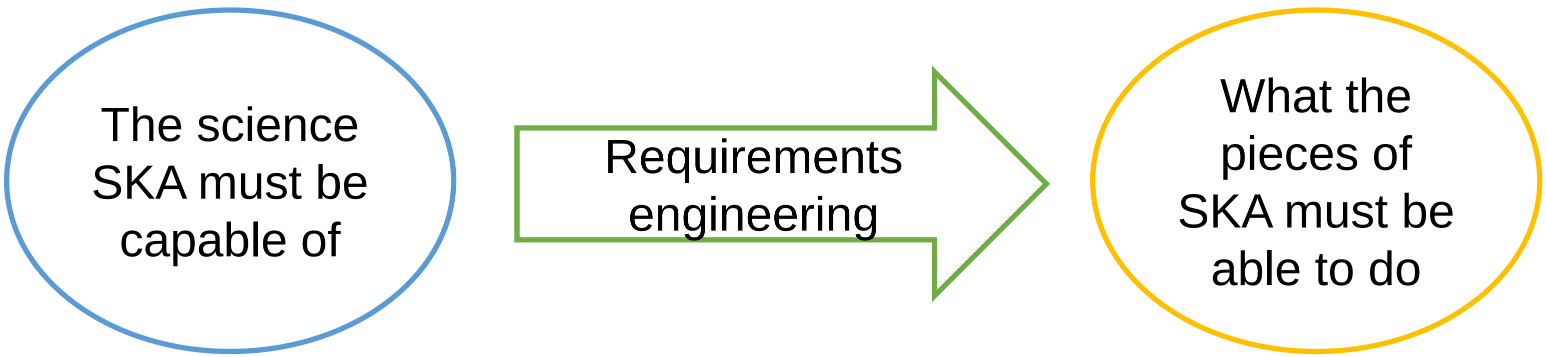
FIGURE 3.4 Importance of the concept stage. DILBERT © 1997 Scott Adams. Used with permission from UNIVERSAL UCLICK. All rights reserved.



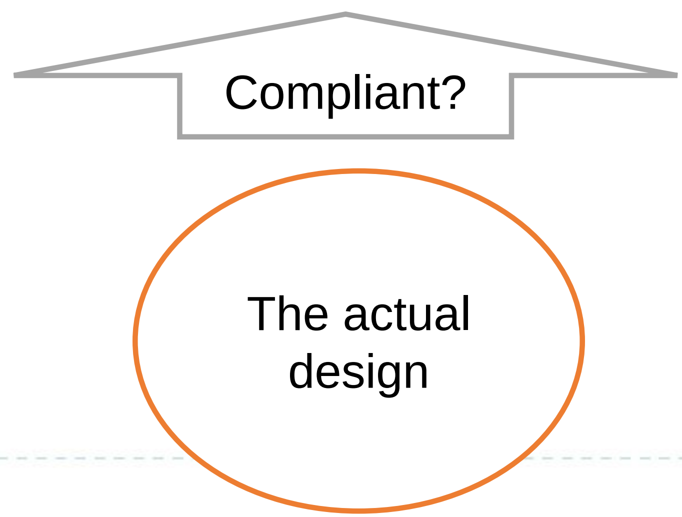
(medium.com)

Requirements

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



- Requirements engineering translates between: what science the SKA must 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.

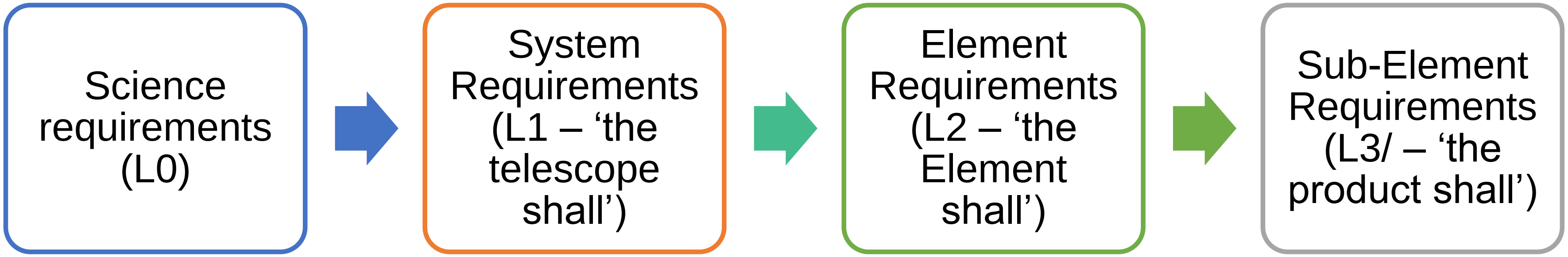


Requirements

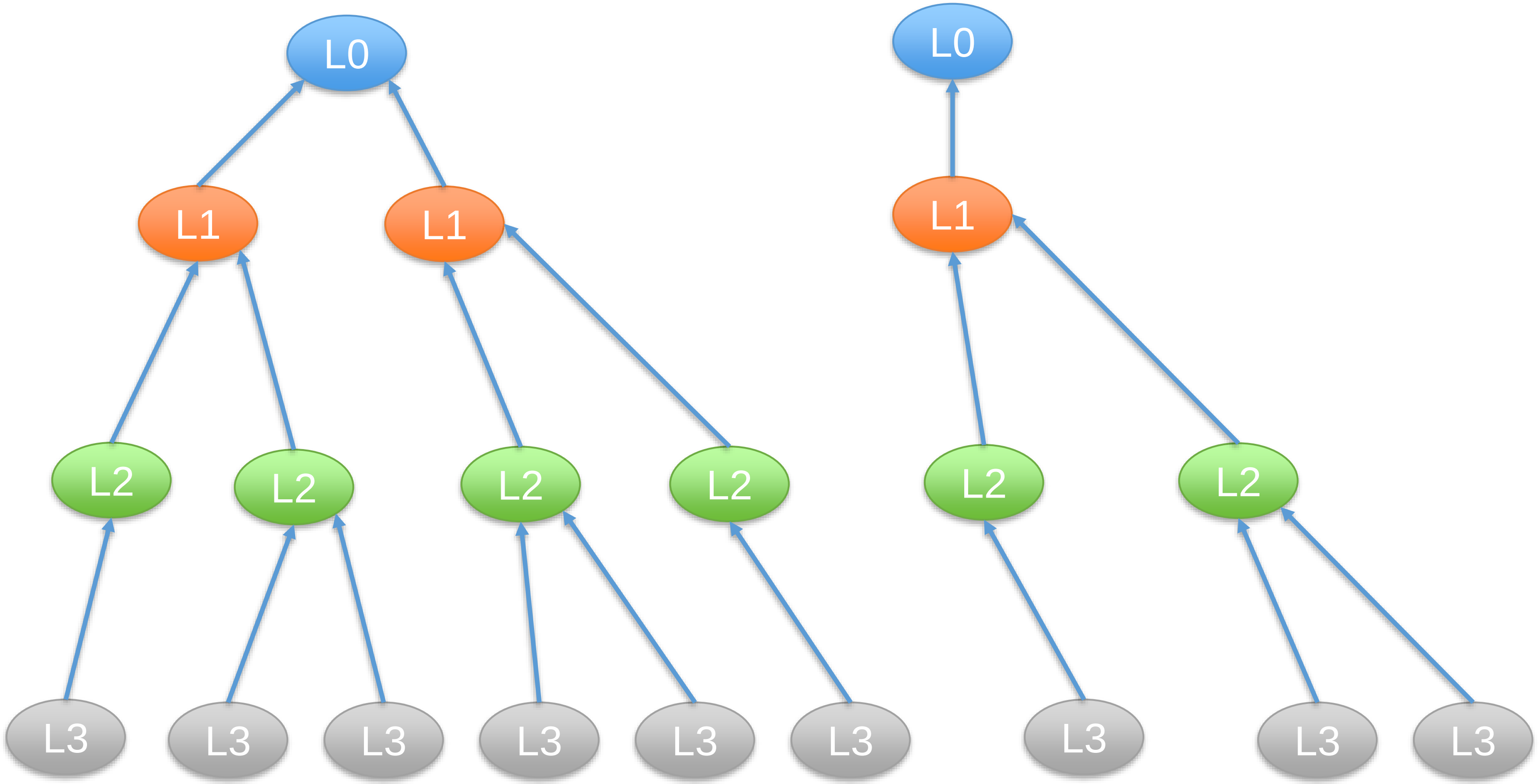
- There are about ~ **600** SKA system L1 requirements. The SKAO allocates each of these to one or more Elements.

- The Element's consortium then analyses this L1 requirement and determines what condition their Element needs to fulfil so that this L1 requirement is achieved. This leads to L2 requirements **(~2500).**

- The same allocation and derivation process is applied one level deeper in the product hierarchy to obtain L3 requirements.



Requirements



QUESTION TIME:

Requirements must be well-written, and there are many ways to write bad requirements. See if you can identify what's wrong with the following requirements:

- The car shall be able to accelerate from 0 to 100km/h in 2 sec using nitro-boosters.
- The range shall be as long as possible.
- Upon contact with air, the glue shall dry within 10 sec and it shall be safe to touch.
- The user shall quickly see her balance on the screen.

Requirements

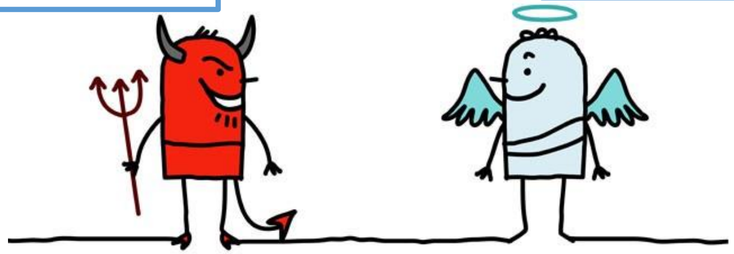
From these examples of “bad” requirements, we’ve learnt that:

- Requirements must be verifiable! Because once the system is built, it will need to be verified against these requirements. When writing a requirement, one should always have in mind how this requirement will be verified.

- Requirements should be solution neutral i.e. they should give the ‘what’ not the ‘how’. However, this becomes more difficult as you decompose requirements to lower levels.

Bad requirements

Good requirements



(medium.com)

QUESTION TIME:

Spend 2 minutes writing a good requirement that might have been used in the development of one of the following:

- A – Mr. Sticky tape for trapping flies
- B – New BMW i3 electric car
- C – EPFL Rolex Center

A

B

C

('Fundamentals of Systems Engineering' course – MIT Open courseware)

QUESTION TIME:

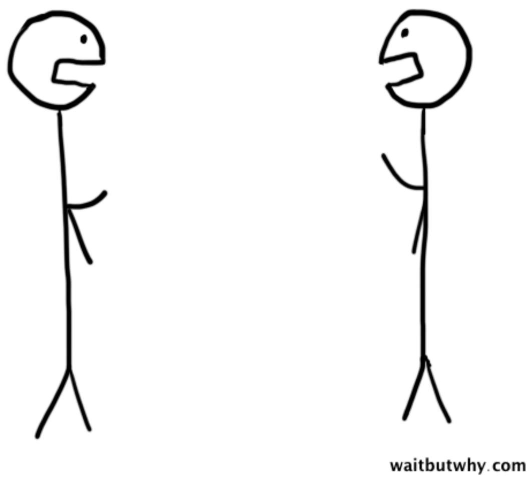
Turn to your partner and discuss for 2 min:

Verification means checking that the system was built right.
Validation means checking that the right system was built.

What is the difference between the two?

Requirements compliance

Jerry, why do we build a L1 compliance matrix?



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

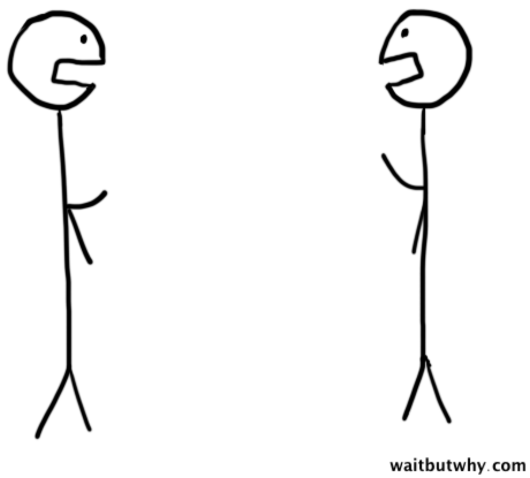
So how do we build such a matrix?

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

Requirements 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?

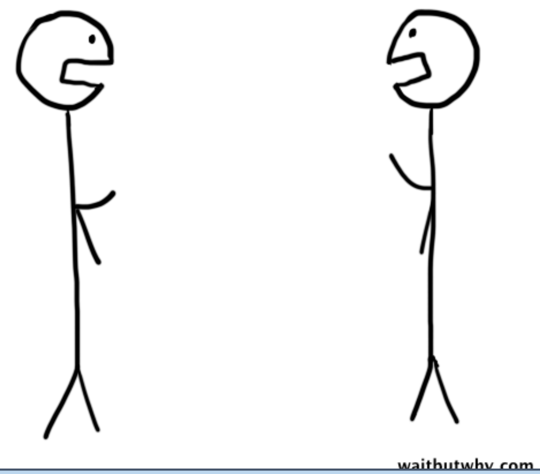
Sufficiency condition:

Achievement of all children requirements → Achievement of parent requirement.

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

Requirements 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?

QUESTION TIME:

We won't answer all these questions now, but turn to your partner and discuss for 2 min the following:

- What is the meaning of the word 'Compliant?'
- If something is not 'Compliant', what else could it be? Is there only one alternative?
- Can something be compliant only after it is built?

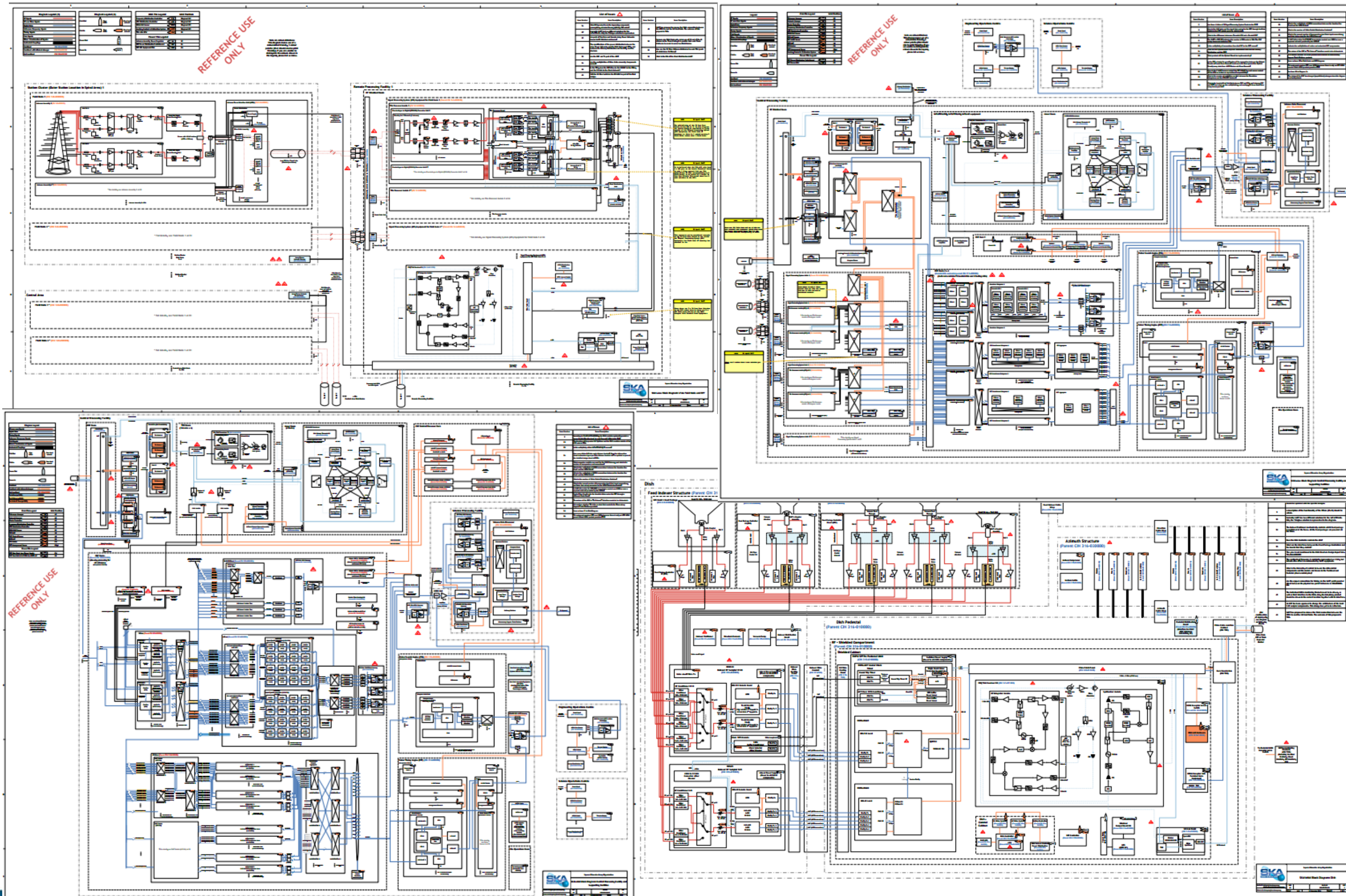
Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- Some SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Integrated design
- Looking forward

Integrated design

- Another important way to check compliance and consistency of the overall design, is to do something we call ‘integrated design’
- This means bringing together the designs of different parts of the telescope and ‘integrating’ them into a single view.
- There are many different kinds of views one can create e.g. hardware views, functional views, monitor and control views, interface views, etc.
- Different diagramming and modelling tools are used to create these views.
- In this way, gaps and inconsistencies can be identified and resolved.

Integrated design – with diagram tools



Integrated design – with diagram tools

- As the diagrams are created using multiple design documents, issues or gaps are noticed.
- These are identified and recorded for resolution.

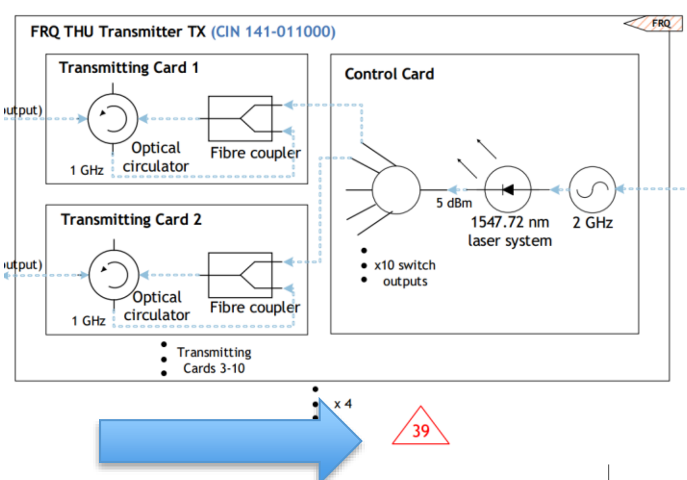
Legend	
RF Signals	
RF over fibre Signals	
Digital Data	
Reference Frequency Signals	
Timing Signals	
C&M Signals	
Other / Combination of Signals	
Mechanical Couplings	
C&M Out	Over NSDN / Over local network
C&M In	Over NSDN / Over local network
Power Out	
Power In	

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

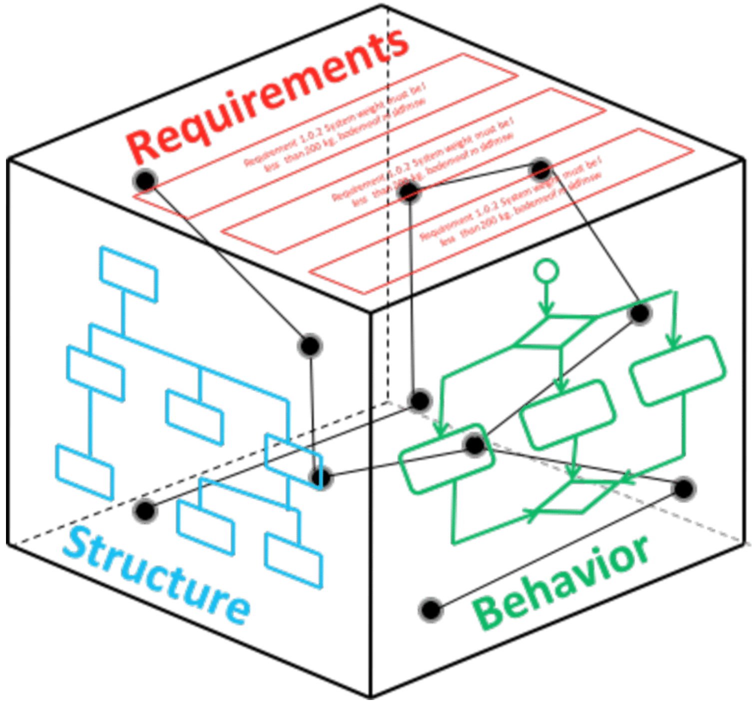
Issue Number	Issue Description
2	Are there 4 Aisles of 19 Signal Processing System Racks in the CPF?
	How is the M&C data and the signal data routed to the RPF through the Data Switch in the CPF (which ports for which data?)
	What is the difference between a Perentie LRU and a Gemini LRU?
	The SADT to 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 – with MBSE tools

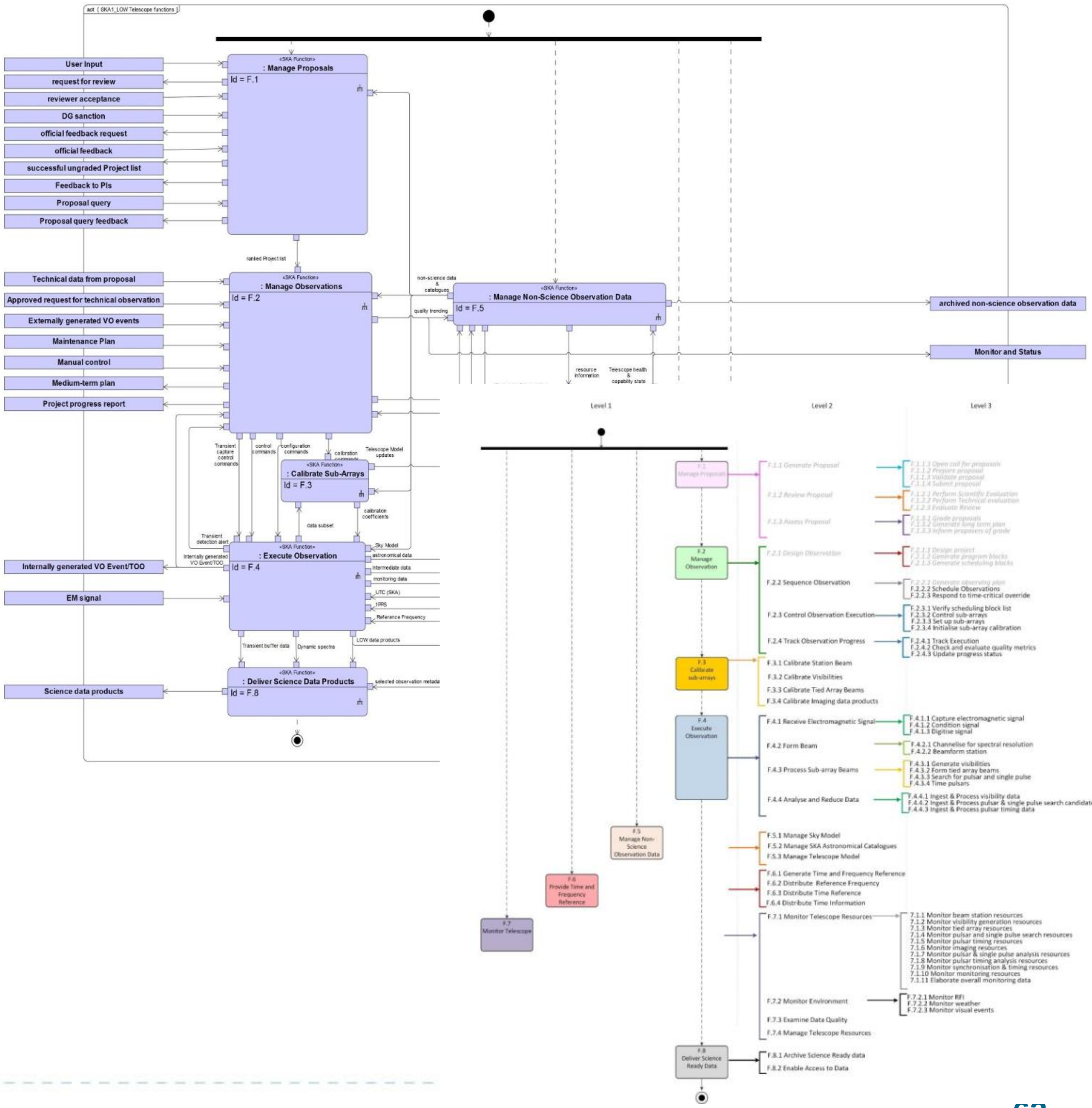
- You can use a diagramming tool to create multiple diagrams that describe a system.
- But these diagrams are not linked in any way. Hence there is no guarantee that they are all consistent.
- An alternative approach is called ‘Model Based Systems Engineering’ (MBSE).
- This approach uses modelling tools to create a single system model. The diagrams are just different views of the model. This guarantees that they are consistent with each other.



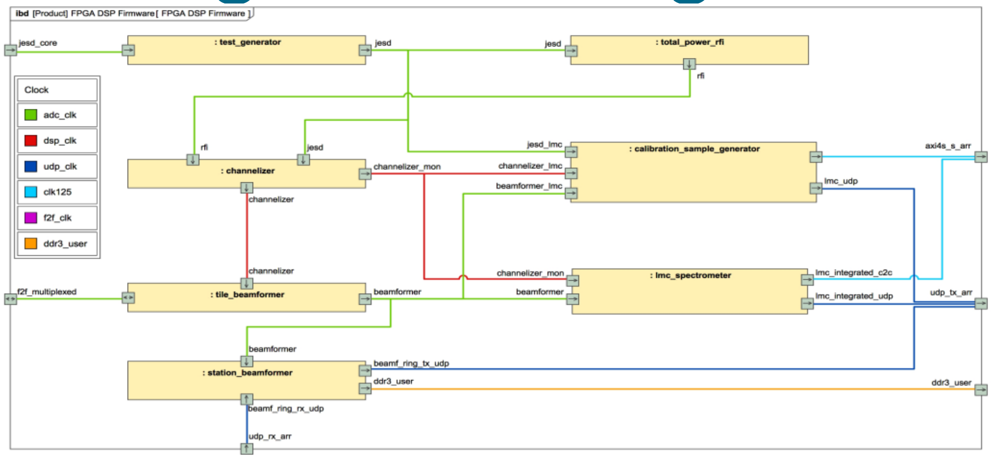
(Paper ‘Architecture to Geometry – Integrating System Models with Mechanical Design, M. Bajaj et al)

Integrated design – with MBSE tools

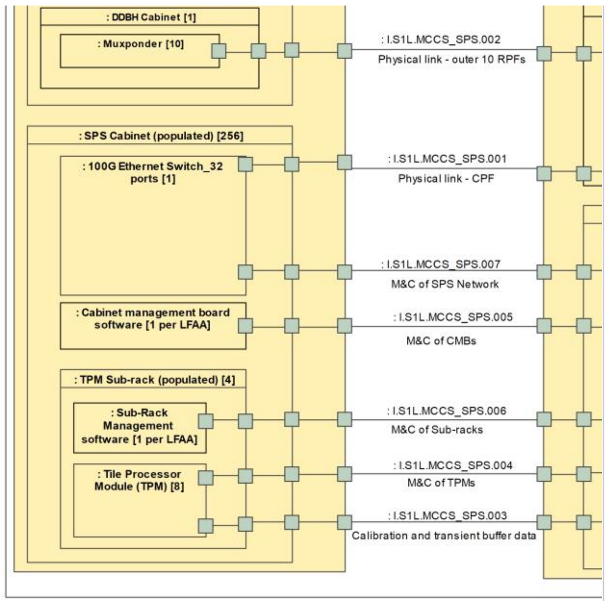
- A functional architecture was created in a top-down fashion.
- This consists of several layers of de-composed functions, with each layer showing how functions are related through their inputs and outputs.
- We are now doing a similar exercise in a bottom-up fashion, using the Element CDR designs as inputs.

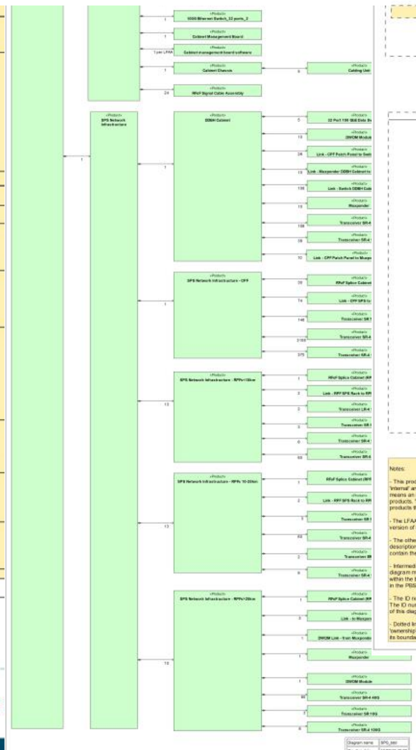


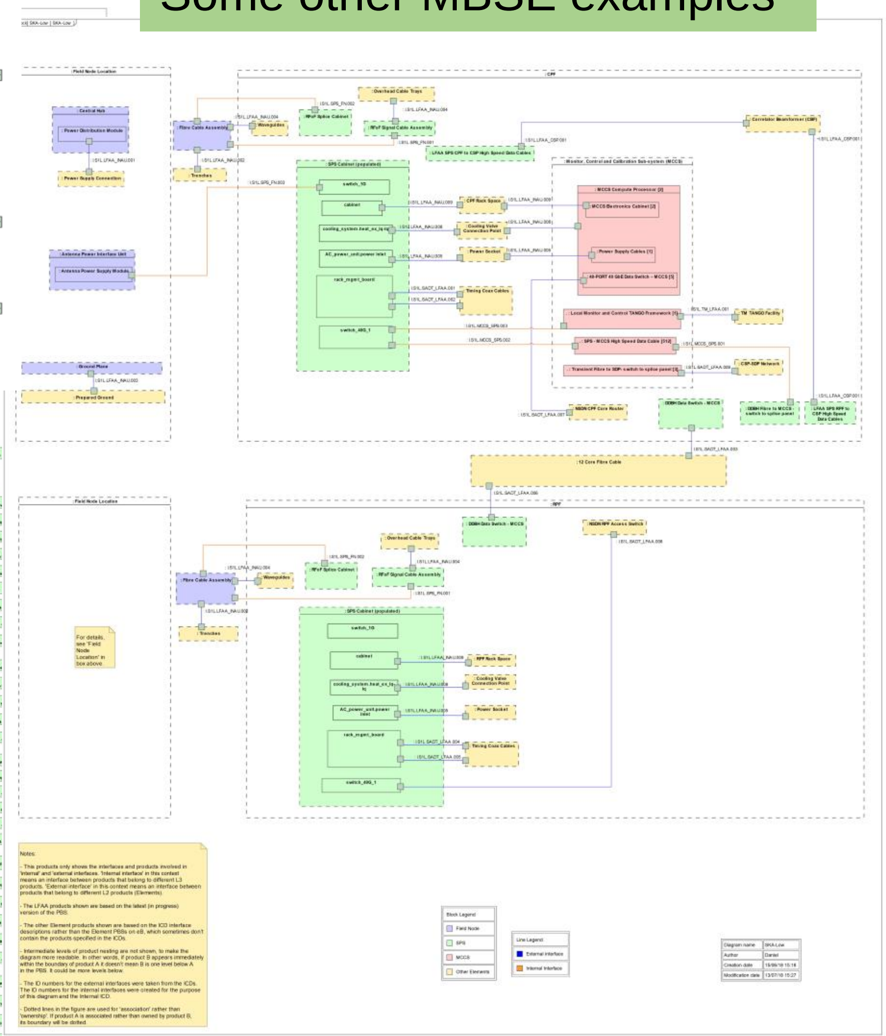
Integrated design – with MBSE tools



Some other MBSE examples

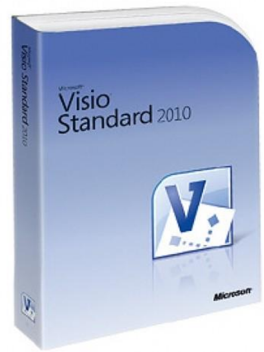






More on tools

- 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)

- These tools together make up an integrated tool environment
- Managing and selecting these tools needs to be carefully done
- Why? To ensure exchange of information between people and tools, and between the tools themselves, is efficient.

Overview

- A recap of the SKA
- The need for Systems Engineering (SE)
- SE activities at the SKAO. Working with:
  - Interfaces
  - Product breakdowns
  - Requirements
  - Change management
  - Integrated design & models
- Looking forward

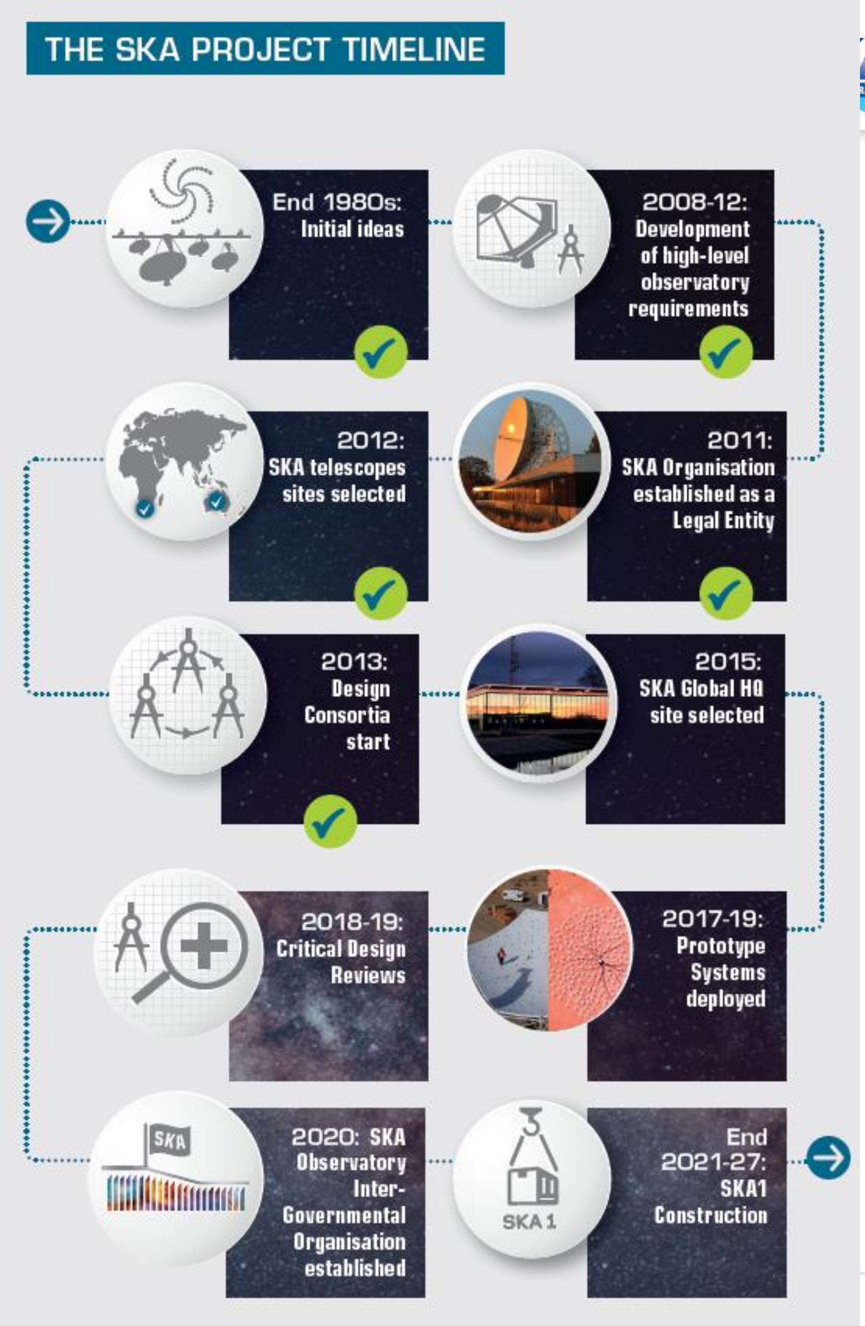
Looking forward

SKA Observatory will be established as an Intergovernmental Organisation in 2020, taking over from the SKA Organisation. It will undertake the construction and operation of the telescope.

As of March 2019, confirmed SKA Observatory members are

Looking forward

- Prototype systems have already begun to be deployed.
- The System Critical Design Review (CDR) will be end 2019.
- The SKA Observatory Inter-Governmental Organisation will be established in 2020
- Construction will run from 2021 till 2027



Thank you!

Any questions?