ADELARD
part of nccgroup

CITY
UNIVERSITY OF LONDON
EST 1894

ASSURING TRANSFORMATIVE TECHNOLOGIES

Assurance 2.0 and Clarisssa tool chain

Prof Robin Bloomfield FREng
City, University of London and Adelard
May 2024

robin.bloomfield@nccgroup.com
r.e.bloomfield@city.ac.uk

Adelard, 24 Waterside, 44-48 Wharf Road, London N1 7UX
T +44 20 7832 5850 E office@adelard.com W www.adelard.com

CLARISSA

Honeywell | ADELARD part of nccgroup | SRI International | UTD THE UNIVERSITY OF TEXAS AT DALLAS

DARPA AUTOMATIC RAPID CERTIFICATION OF SOFTWARE (ARCOS)

Consistent Logical Automated Reasoning for Integrated System Software Assurance (CLARISSA)

ARCOS Technical Area 3

Srivatsan Varadarajan
John Rushby
Robin Bloomfield
Gopal Gupta

Srivatsan Varadarajan
Program Manager / PI
Honeywell International

- Anitha Murugesan
- Isaac Hong Wong
- David Counts

John Rushby
SRI

- Sam Owre

Robin Bloomfield
Adelard (NCC)

- Robert Stroud
- Kateryna Netkachova
- Elitza Karadotcheva
- Marco Passuello

Gopal Gupta
UT Dallas

- Elmer Salazar
- Sarat Varanasi
- Parth Padalkar

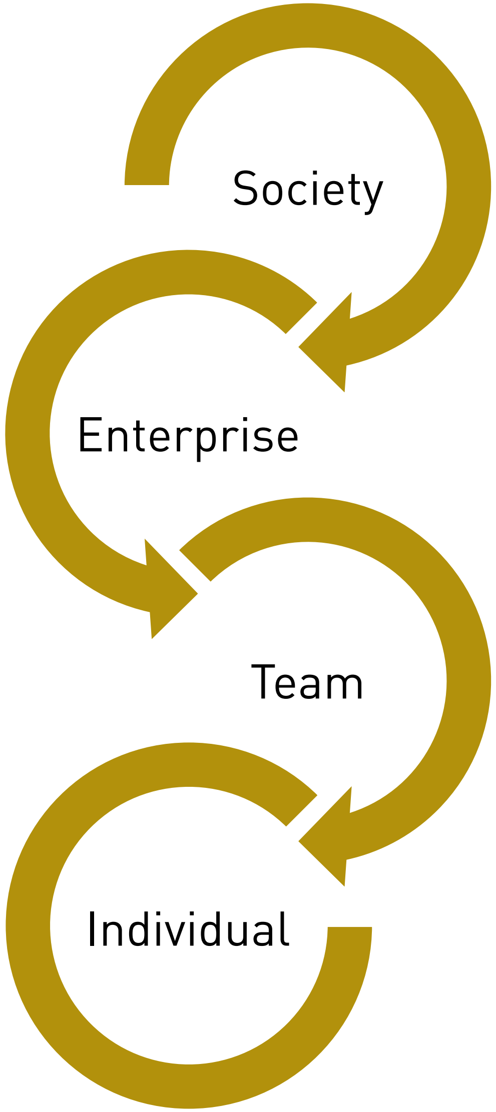
Disclaimers: This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

OUTLINE

- Acknowledgements
- Background
  - Why should you be interested in assurance cases?
  - The nature and challenge of Transformative Technologies
- Assurance 2.0 methodology and technology
  - Overview, key stuff
- Assurance Technology
  - Synthesis
- Correct by construction
  - Protection System
- Conclusion

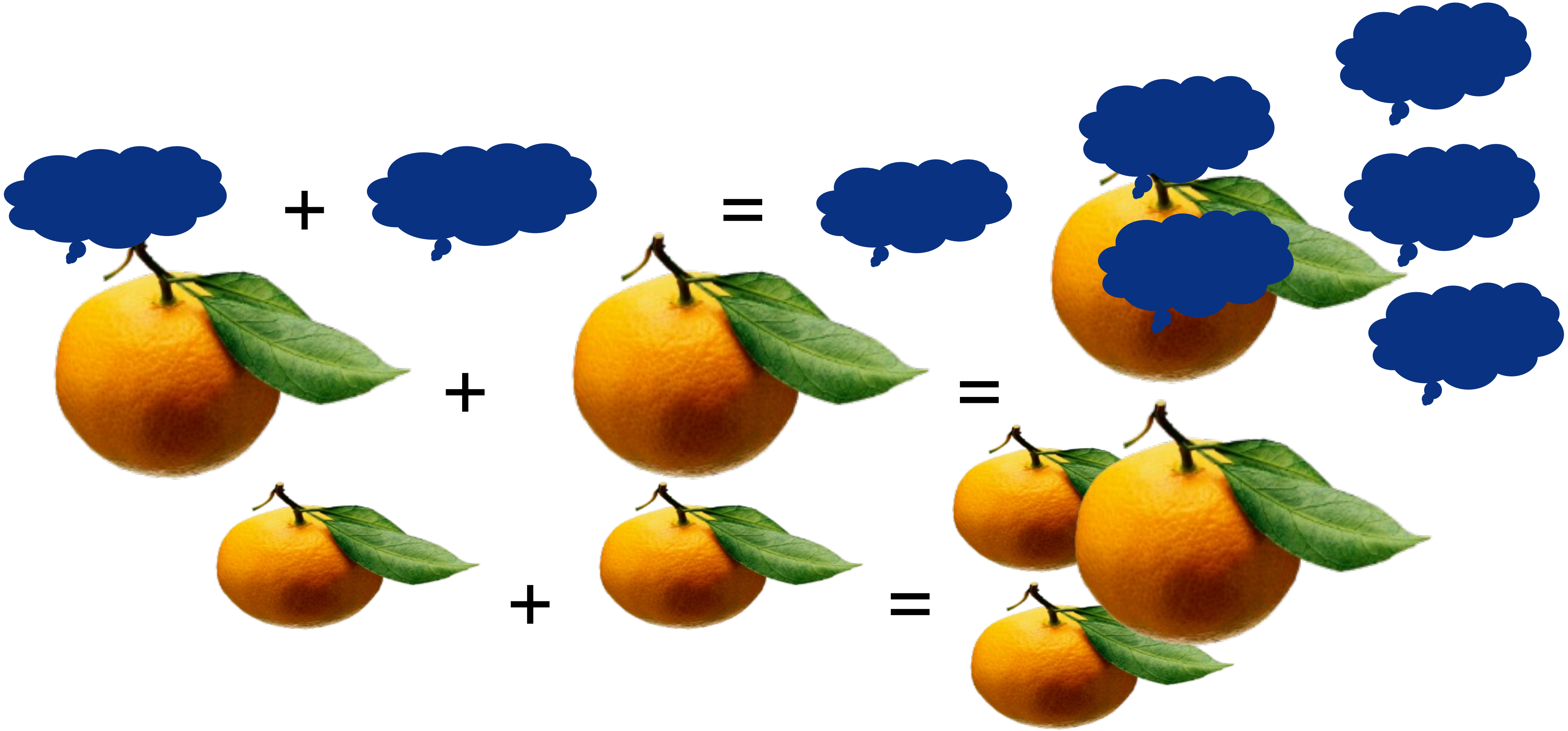
COMMUNICATION, UNDERSTANDING, REASONING

- An assurance case is
  - "a documented body of evidence that provides a convincing and valid argument that a system is adequately dependable (or not) for a given application in a given environment"
- An assurance case has two roles:
  - communication is essential, to build confidence and consensus
  - recording understanding and reasoning
    - both are required to have systems that are trusted and trustworthy



ADELARD part of nccgroup

A FORMAL METHODS PERSPECTIVE
1 +1 = 2?



ADELARD part of nccgroup

DEDUCTIVE AND INDUCTIVE ARGUMENTS –WHY SEPARATE OUT?

Science of security – importance of deductive/inductive split

“We now detail security research failures to adopt accepted lessons from the history and philosophy of science.

A. Failure to observe inductive-deductive split

Despite broad consensus in the scientific community, in Security there is repeated failure to respect the separation of inductive and deductive statements “

SoK: Science, Security, and the Elusive Goal of Security as a Scientific Pursuit

Cormac Herley
Microsoft Research, Redmond, WA, USA
cormac@microsoft.com

P.C. van Oorschot
Carleton University, Ottawa, ON, Canada
paulv@scs.carleton.ca

DOI: 10.1109/SP.2017.38
Conference: 2017 IEEE Symposium on Security and Privacy (SP)

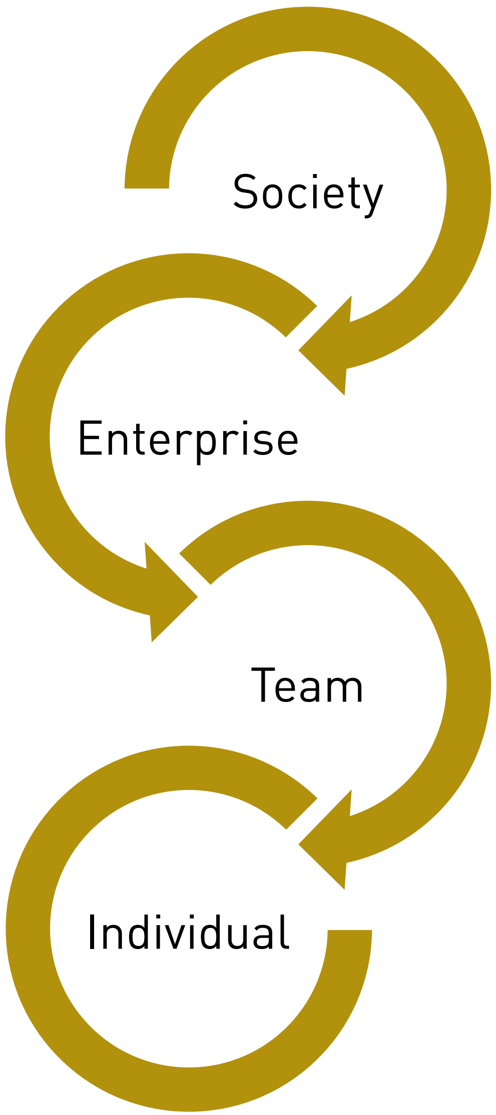
If it's Provably Secure, It Probably Isn't: Why Learning from Proof Failure is Hard

Ross Anderson[1], Nicholas Boucher[2]

[1] Universities of Cambridge and Edinburgh
[2] University of Cambridge

ADELARD part of nccgroup

Reasoning and communication

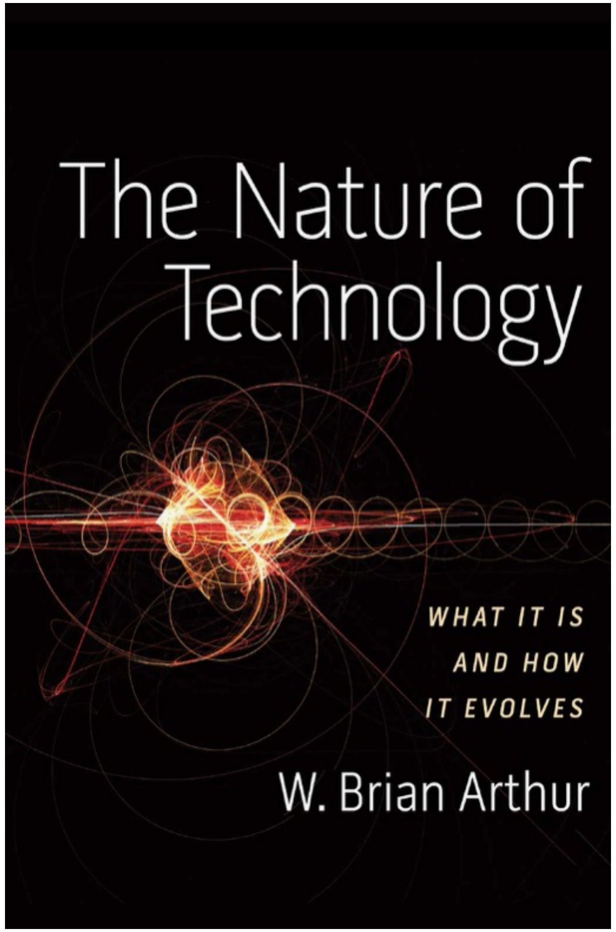


ASSURANCE 2.0

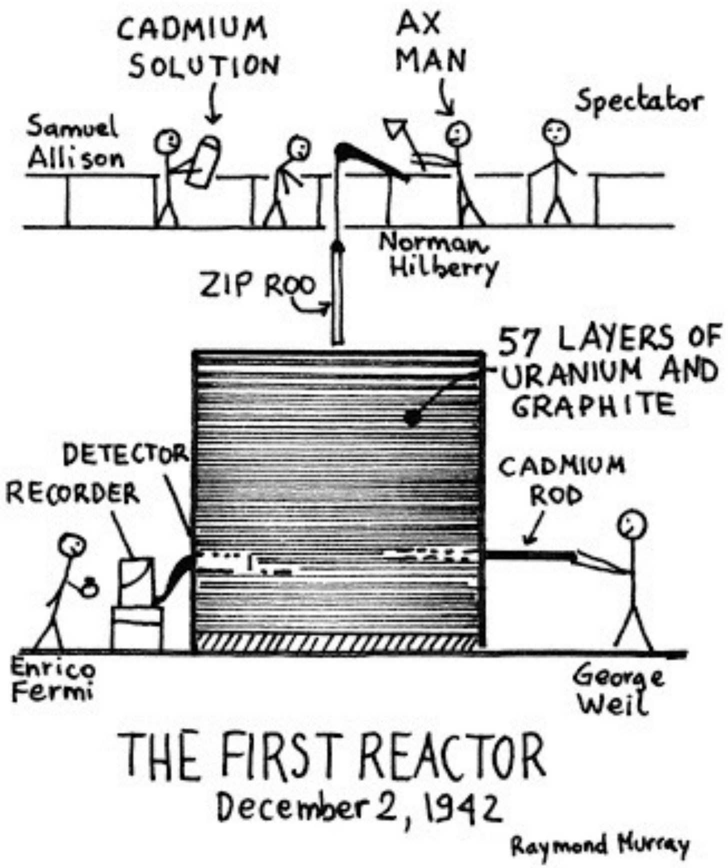
R Bloomfield and J Rushby, Assurance 2.0 Manifesto https://arxiv.org/abs/2004.10474

TECHNOLOGY EVOLUTION

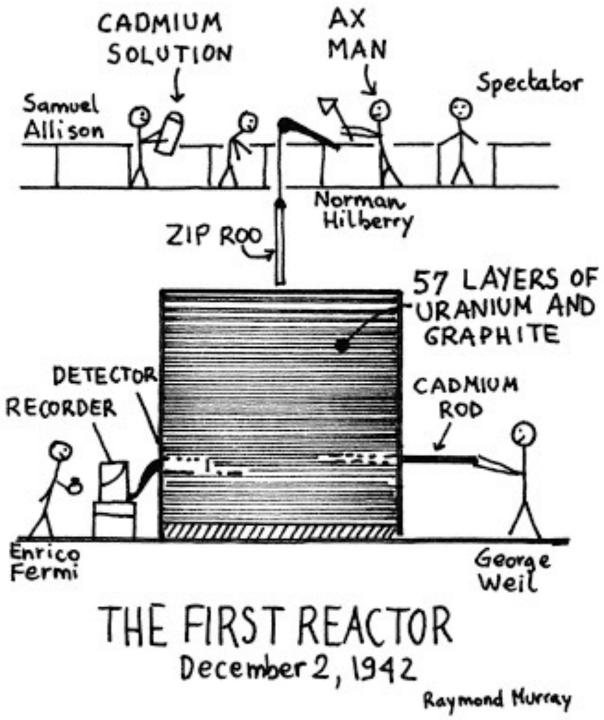
- Structural deepening - adaptations to remove obstacles, improve performance but
  - “Over time it becomes encrusted with systems and subassemblies hung onto it to make it work properly, handle exceptions, extend its range of application, and provide redundancy"
- Adaptive stretch – for new applications or requirements
- Structural deepening, lock-in, and adaptive stretch—have a natural cycle
- Eventually old principle is strained beyond limits and gives way to a new one



REACTOR PROTECTION SYSTEMS



REACTOR PROTECTION AND CONTROL SYSTEMS



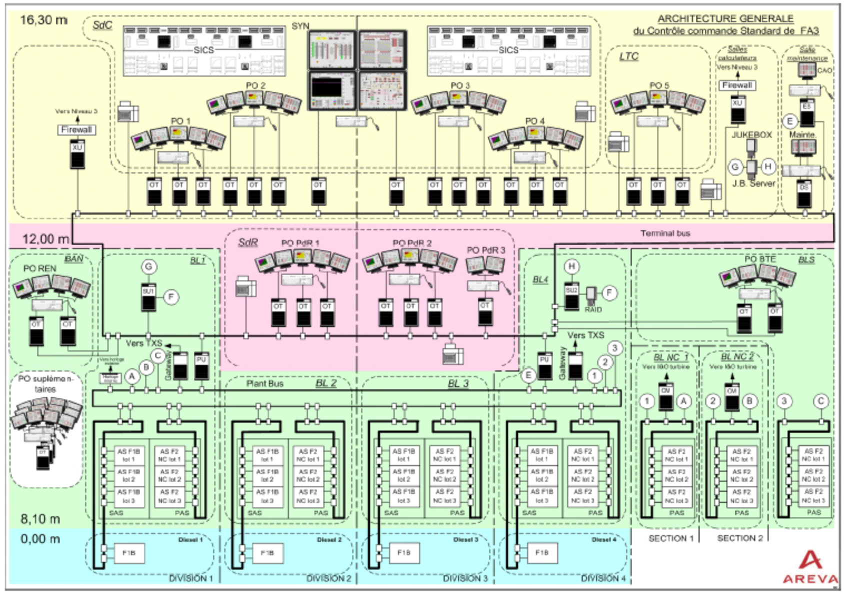


FIGURE 1: SCHEMA D'ARCHITECTURE DU CONTROLE COMMANDE STANDARD DE FA3

TRANSFORMATIVE TECHNOLOGIES

- Are *new* and *themselves changing*
  - Evidence base, fluid
- *Change the world*
  - Are performative
  - Change the system they are part of e.g. user adaptation
  - Change the wider system e.g. risk preference, adversary behaviour, markets
- Integrate *many existing technologies*
  - Build on existing systems and software
    - E.g quantum, LLM, formal method
- *Challenge status quo*

TRANSFORMATIVE TECHNOLOGIES

- Are *new and themselves changing*
  - Evidence base, fluid
- *Change the world*
  - Are performative
  - Change the system they are part of e.g. user adaptation
  - Change the wider system e.g. risk preference, adversary behaviour, markets
- Integrate *many existing technologies*
  - Build on existing systems and software
    – E.g quantum, LLM, formal method
- *Challenge status quo*

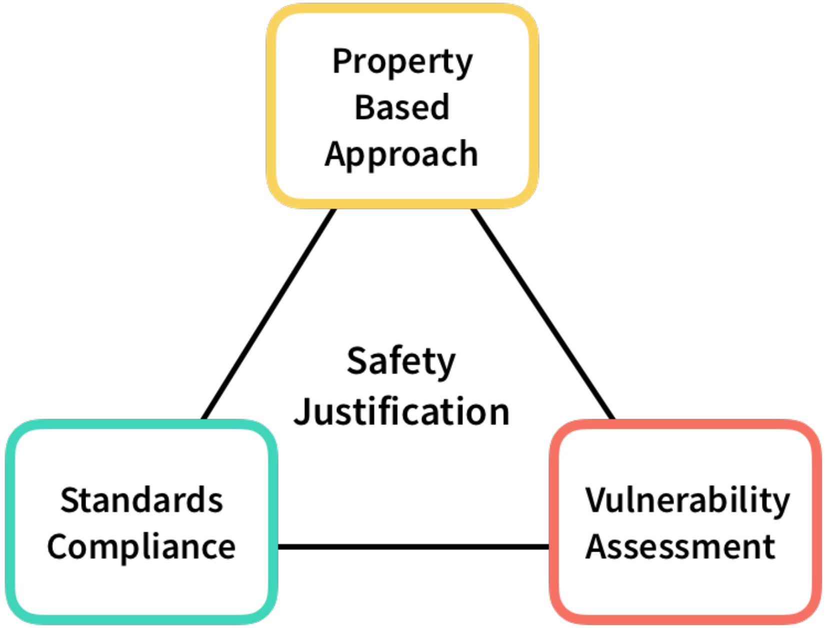
Define an Assurance horizon
Up to which we can assure, can we detect when we get past it
Define broader socio-technical system scope
Open Systems Dependability Perspective

IEC 62853

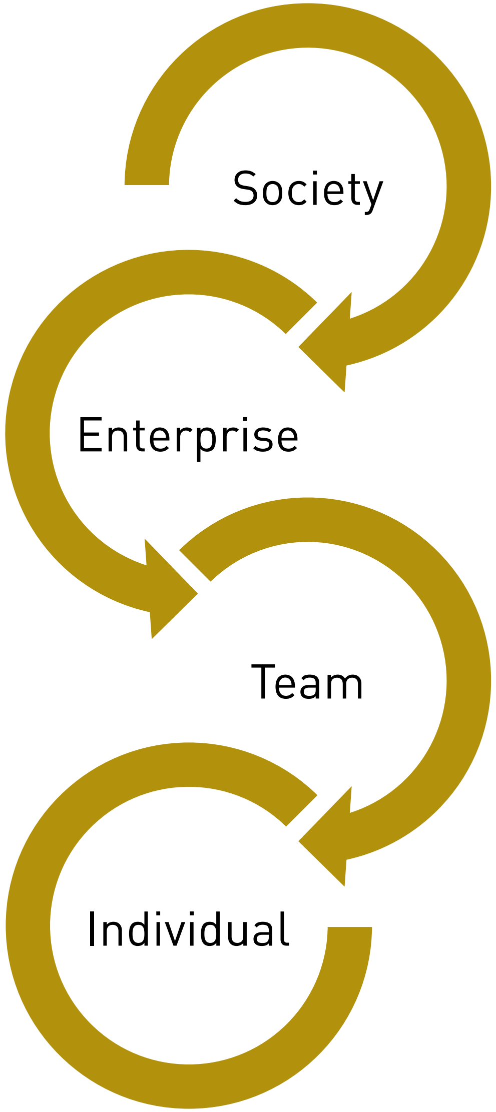
"PERFORMATIVE MODELS" – CHANGE THE WORLD

TRANSFORMATIVE TECHNOLOGIES

- Are *new* and *themselves changing*
  - Evidence base, fluid
- *Change the world*
  - Are performative
  - Change the system they are part of e.g. user adaptation
  - Change the wider system e.g. risk preference, adversary behaviour, markets
- *Integrate many existing technologies*
  - Build on existing systems and software
    - E.g quantum, LLM, formal method
- *Challenge status quo*



Reasoning and communication



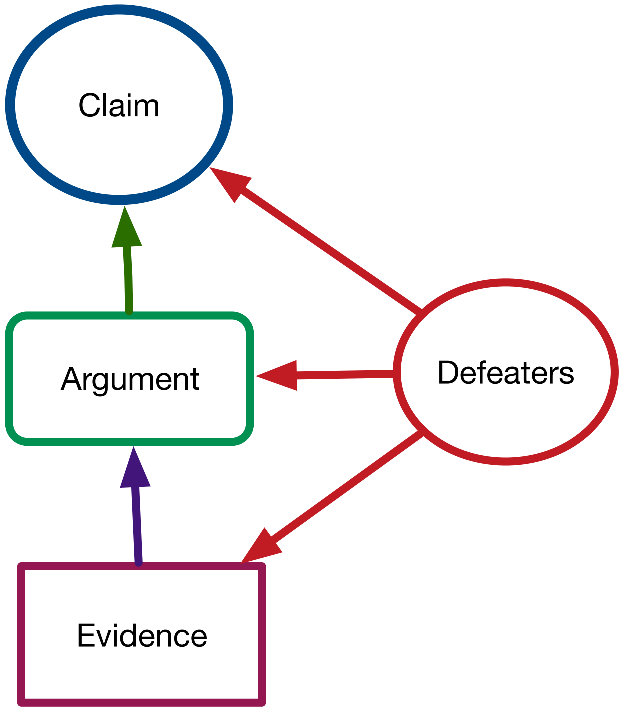
ASSURANCE 2.0

R Bloomfield and J Rushby, Assurance 2.0 Manifesto
https://arxiv.org/abs/2004.10474

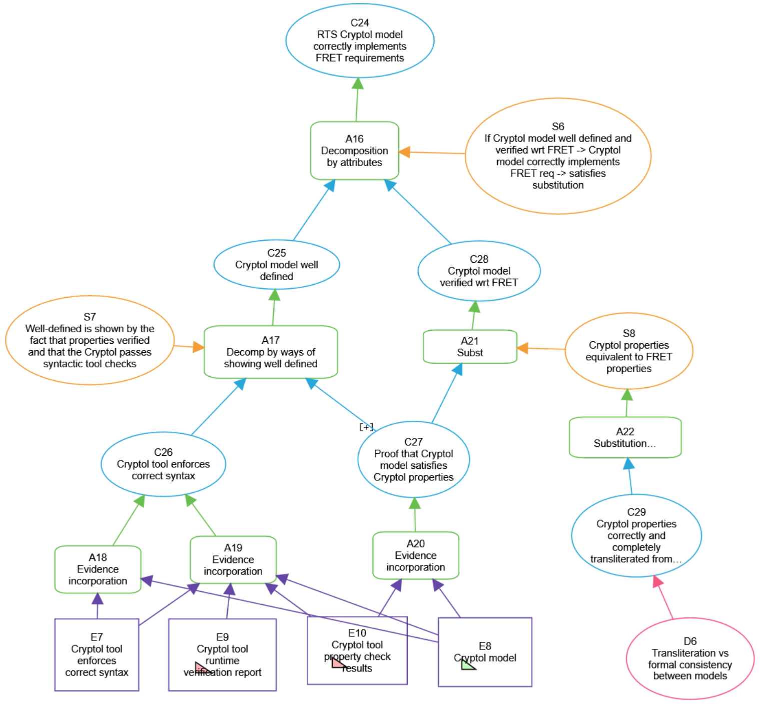
ASSURANCE 2.0 KEY POINTS

- Assurance 2.0: "Simplicity Through Rigor“
- Key topics
  - Claims, Argument, Evidence (CAE) and Defeaters
  - CAE Blocks
  - Positive, negative and residual doubt perspective
  - Evidence and confirmation theory
  - Summary report
- Explicit attention to bias – confirmation theory, defeaters, counter cases
- A completed assurance case is an engineered artifact
  - Stopping rule of review, challenge and no unresolved doubts, “indefeasibility”
- Clarissa/ASCE provides tooling for the argument, links to native tools of the other elements

CLAIMS, ARGUMENTS, EVIDENCE, DEFEATERS



- ***Claims*** **-** assertions put forward for general acceptance
- ***Arguments*** **-** link the evidence to the claim
- ***Evidence*** **-** the basis of the justification of the claim
- ***Defeater*** **–** reasons for doubting



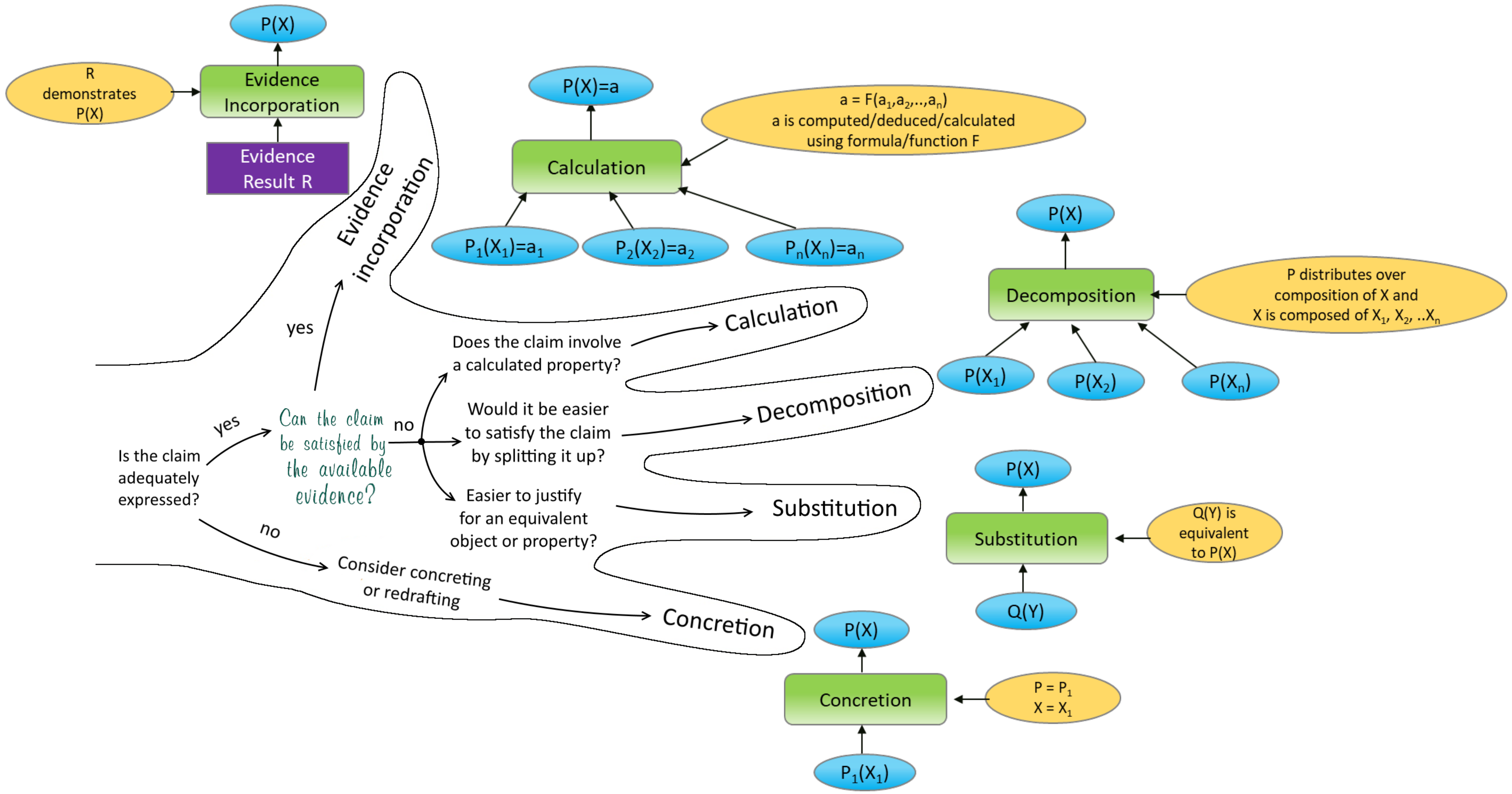
BACKGROUND TO BLOCKS – EMPIRICALLY BASED

- Smart sensor safety case for the nuclear industry
- CCF case from previous research results
- The safety of a computer based medical device
- Generic medical device safety case
- The dependability of an electronic funds transfer system
- Changes to a payments system
- A defense training system
- Safety of changes to a command and control system
- An approach to assessing safety of ordnance
- A weapons safety case
- A case supporting vulnerability testing of an eVoting machine

Language initially unconstrained CAE and GSN

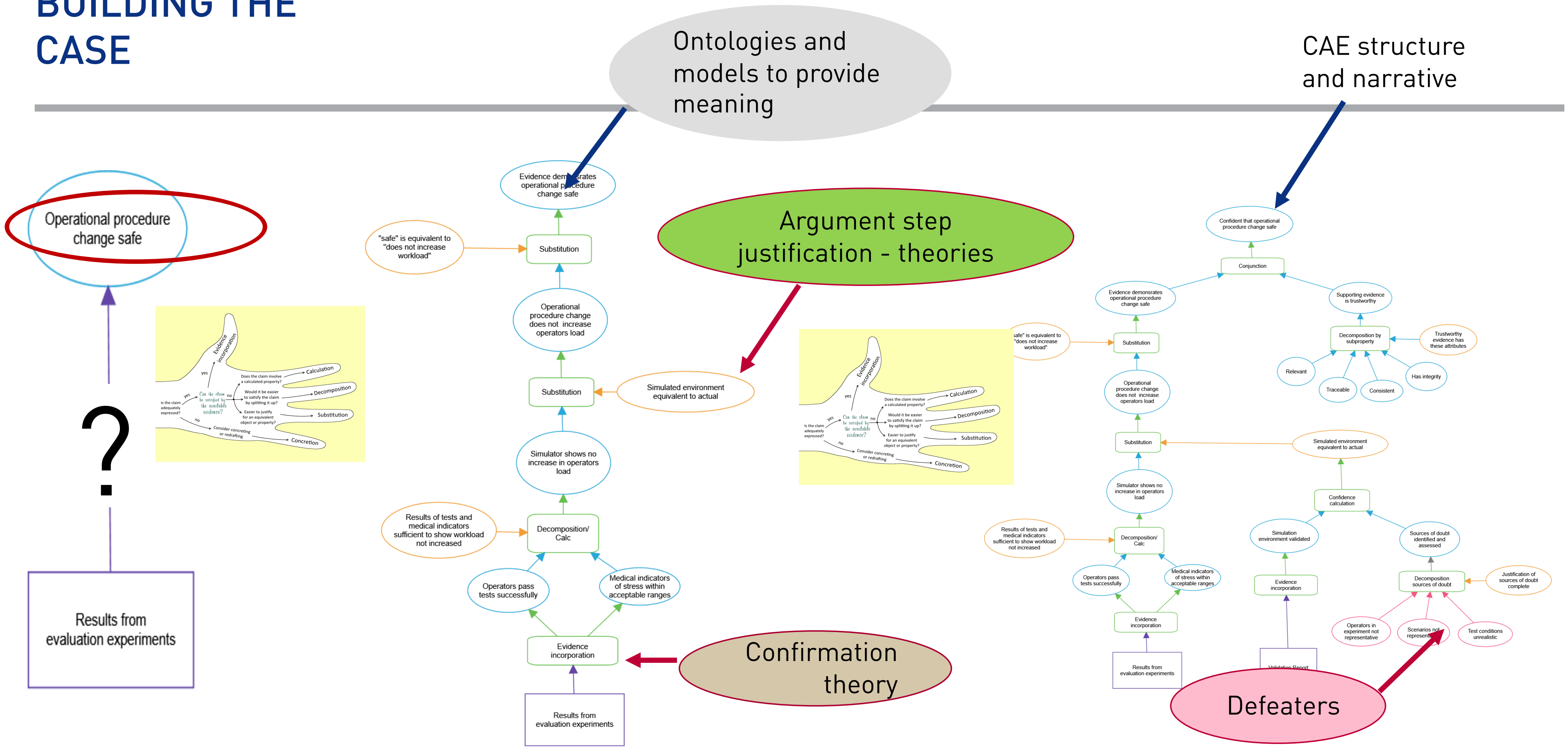
Empirically found a small set of constructs expressive enough - CAE “Blocks”

ADELARD part of nccgroup



P(X)
R demonstrates P(X)
Evidence Incorporation
Evidence Result R
Evidence incorporation
P(X)=a
Calculation
a = F(a1,a2,..,an)
a is computed/deduced/calculated
using formula/function F
P1(X1)=a1
P2(X2)=a2
Pn(Xn)=an
P(X)
Decomposition
P distributes over composition of X and X is composed of X1, X2, ..Xn
P(X1)
P(X2)
P(Xn)
yes
Does the claim involve a calculated property?
Calculation
Would it be easier to satisfy the claim by splitting it up?
Decomposition
Is the claim adequately expressed?
yes
Can the claim be satisfied by the available evidence?
no
Easier to justify for an equivalent object or property?
Substitution
no
Consider concreting or redrafting
Concretion
P(X)
Substitution
Q(Y) is equivalent to P(X)
Q(Y)
P(X)
Concretion
P = P1
X = X1
P1(X1)

BUILDING THE CASE

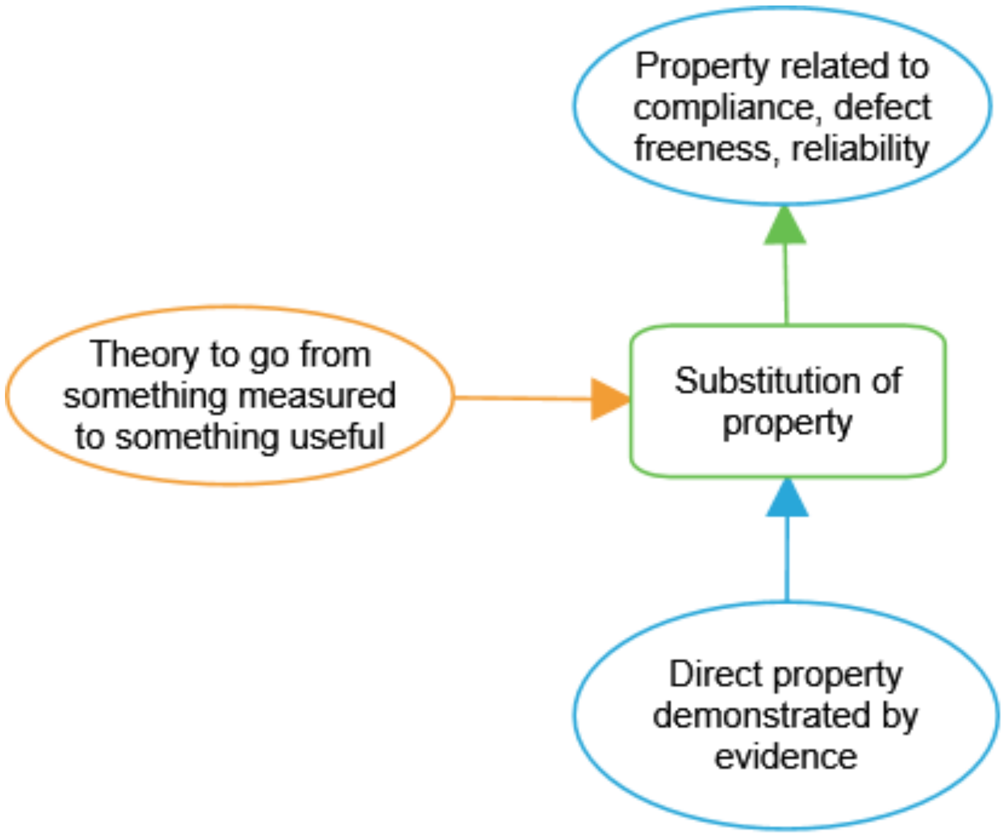


Synthesis tools

Assurance Case report

THEORIES: FROM SOMETHING MEASURED TO SOMETHING USEFUL

- Analysis
  - Verification analysis -> correct
  - Abstract interpretation -> absence of RTE
  - Prototype -> Production
- Ontology
  - Types
  - System
- Rewrite rules
  - Grammar



DEVELOPMENT AND ASSESSMENT OF ASSURANCE CASES
Positive, negative, residual doubts

- **Positive:** logical soundness of argument plus scientific assessment of theories
  - Soundness is logical validity (checkable) plus credibility of evidence and reasoning
  - Credibility of evidence is "weighed" by confirmation measures
    - *Forces contemplation of defeaters at evidence level*
  - And ensured for reasoning steps by (checkable) side-conditions (for deductiveness)
- **Negative:** active search for and resolution of defeaters
  - Defeaters are retained to assist evaluators
  - Value their coverage, significance, and diversity more than quantity
- **Residual Doubts:** what about the gaps?
  - Localized for analysis as potentially valid defeaters, inductive steps
  - Need to assess risk: consequences and likelihood
  - We propagate probabilistic belief in several ways to assist different stakeholders
    - Internalized explicitly within claims and associated models/theories
    - Conservative sum of doubts
  - Purpose is to explore assessments and tradeoffs, not deliver verdict
- Overall evaluation yields degree of belief in top claim
  - Sentencing statement or Assurance Case report supports overall verdict

ADELARD part of nccgroup

CONFIRMATION MEASURES

https://tahb.shinyapps.io/confirmation_theory/

- Type 1 - this measure looks at the impact of the evidence on our belief in the claim.
  - P(C) is our confidence in the claim, given no other information. We want to assess the value of additional facts contributed by the evidence and then assign a value to P(C|E) The measure we use is the Keynes one:

$$\text{Keynes}(C, E) = \log \frac{P(C|E)}{P(C)} \equiv \log \frac{P(E|C)}{P(E)}$$

- Type 2 – this measure asks us to compare our belief in the likelihood of the evidence, given the claim is true, vs. if it is false (i.e., P(E |C) vs. P(E |¬C)). We use the Kemeny-Oppenheim (KO) or the Good measure:

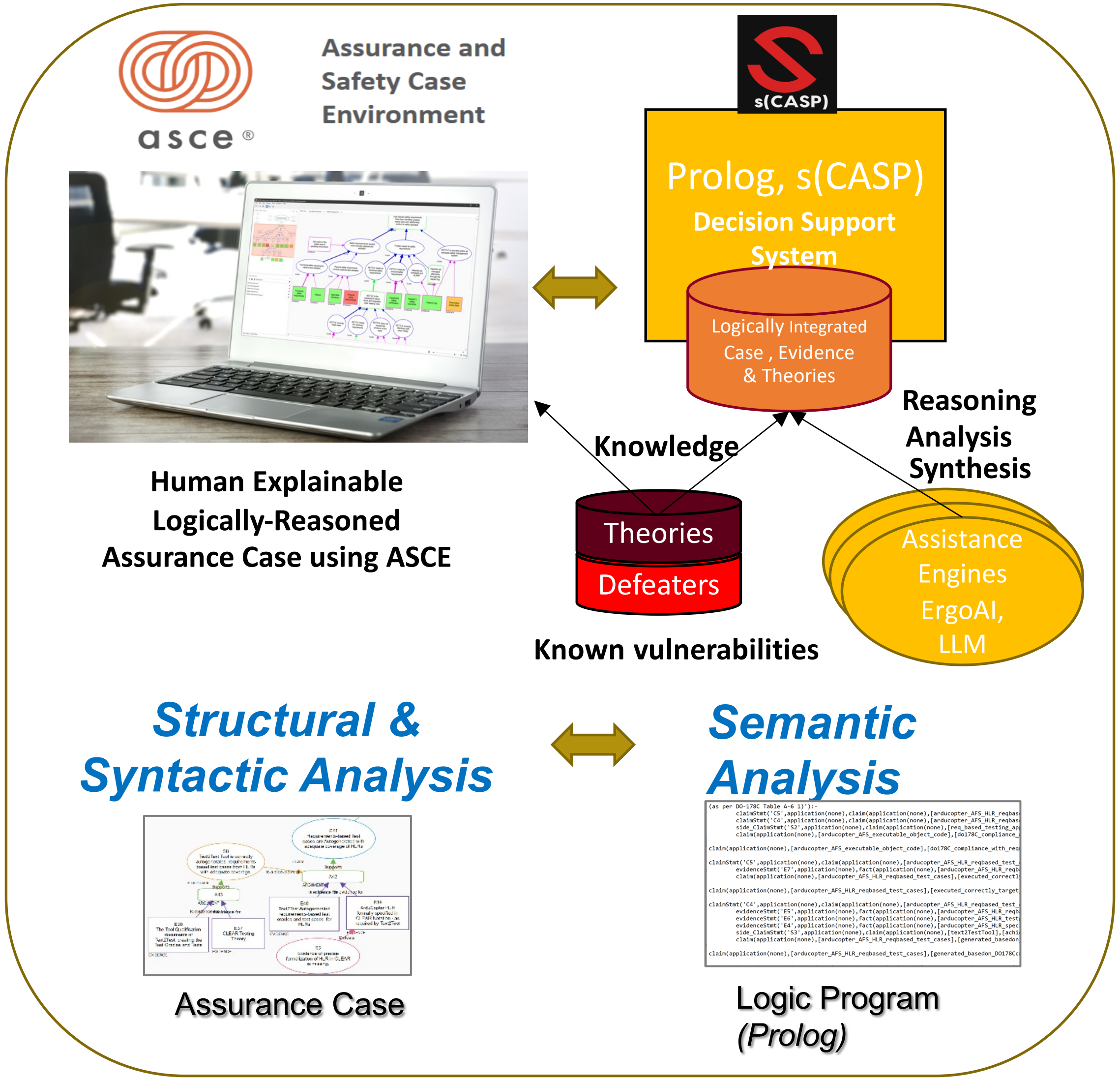
$$\text{KO}(C, E) = \frac{P(E|C) - P(E|\neg C)}{P(E|C) + P(E|\neg C)}$$

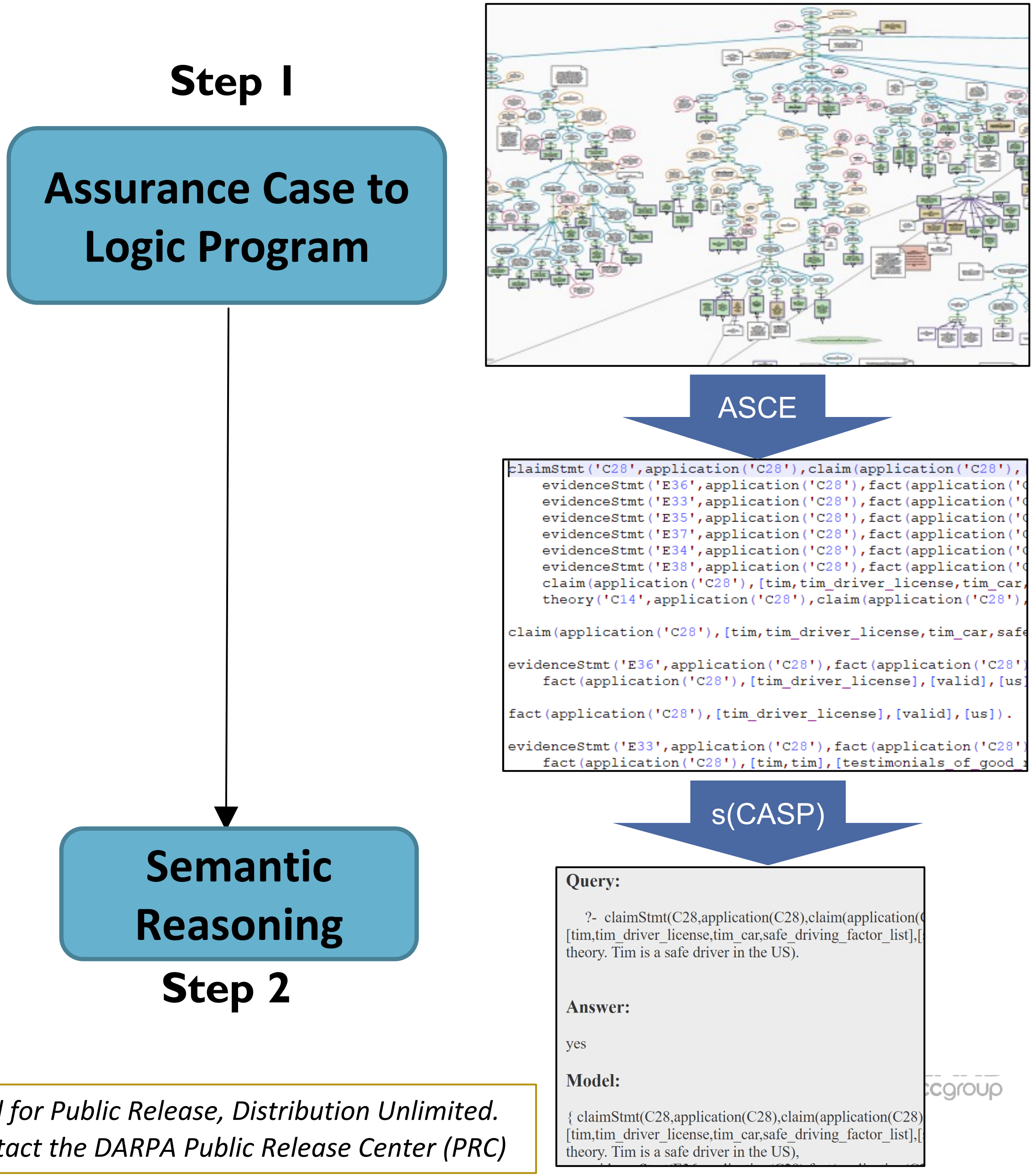
$$\text{Good}(C, E) = \log \frac{P(E|C)}{P(E|\neg C)}$$

  - In considering the negative claims additional assumptions or defeaters might be discovered

TECHNOLOGY AND TOOLS

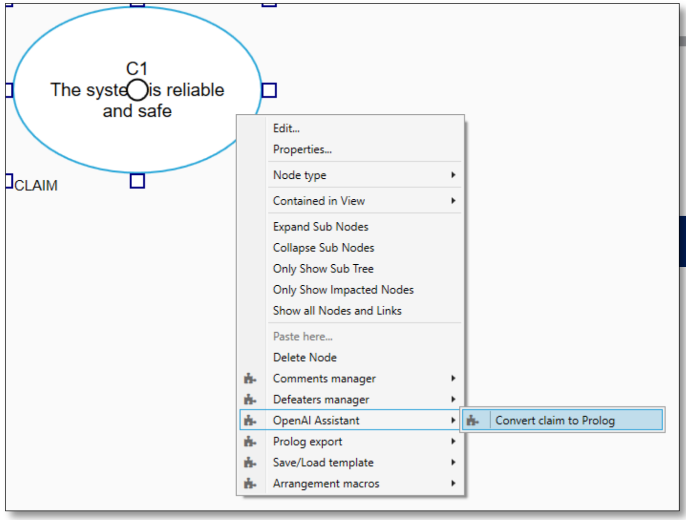
CLARISSA TOOLS

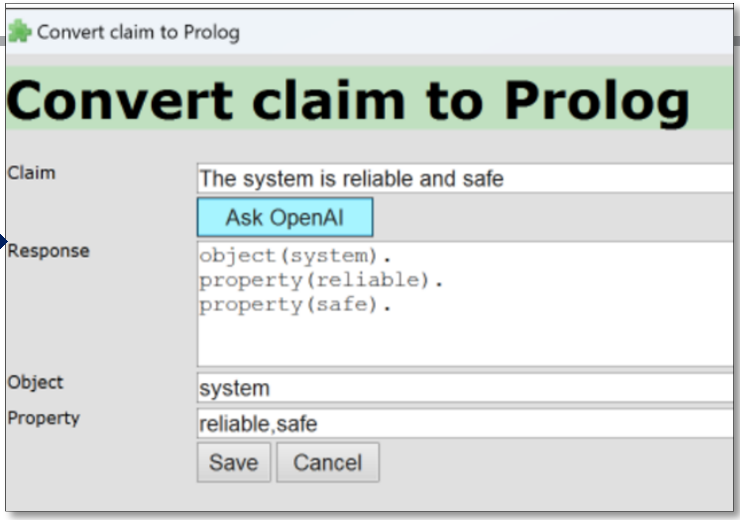




Distribution Statement "A": Approved for Public Release, Distribution Unlimited.
If you have any questions, please contact the DARPA Public Release Center (PRC)

LLM SUPPORT FOR FORMALIZATION





- Assessing accuracy of translation and back translation
  - Using corpus of anonymized claims, based on actual cases
  - Accuracy of NL -> formalized claims, currently ~96%
- A failure mode is likely to occur at ~4% e.g. *Claim is too generic where more context is needed, elaboration of existing claims or unreliable external sources*

Extracted claims

Formalised claims

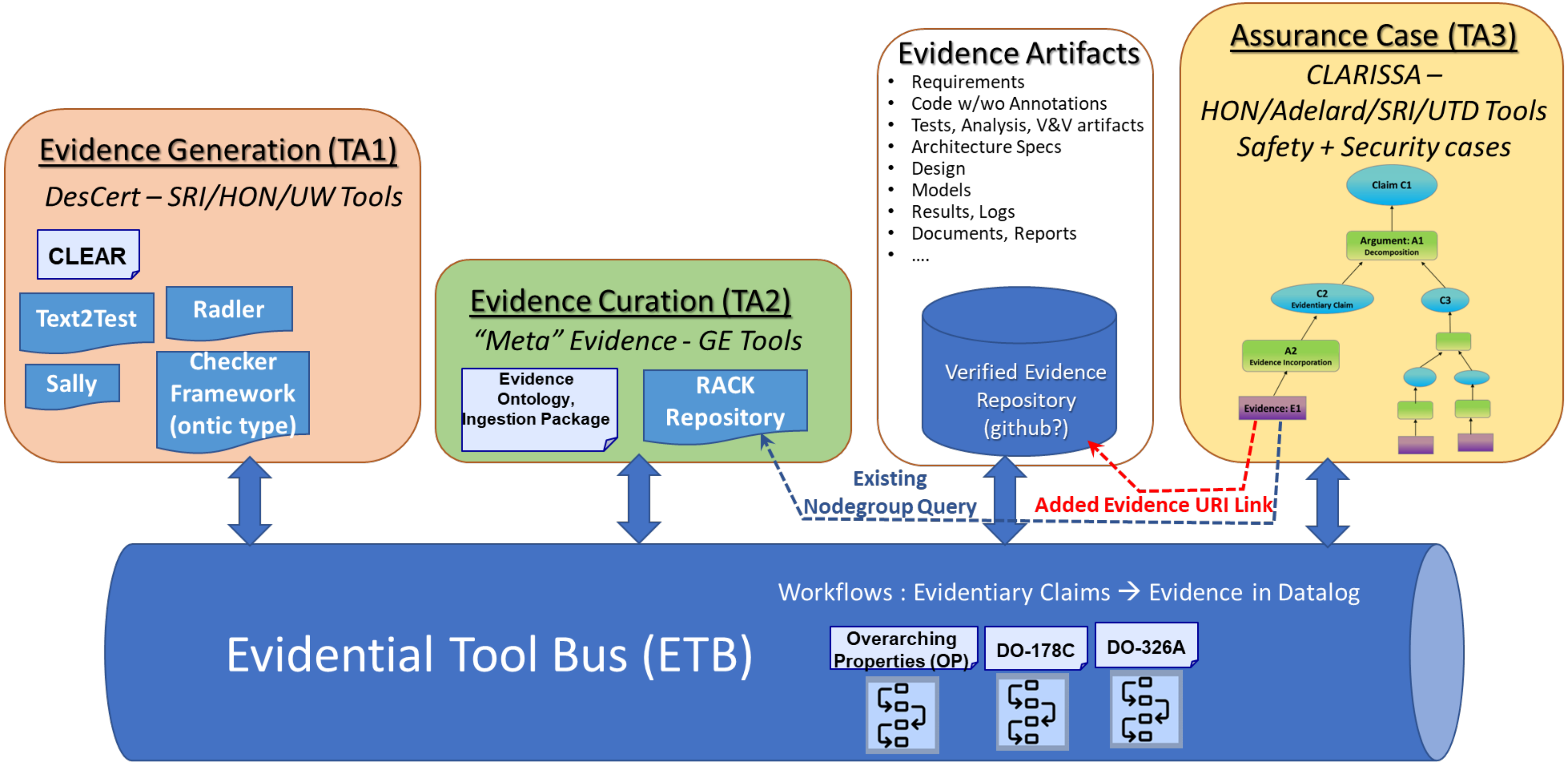
5. claim5: FRET's realizability checker is used to verify the correctness of requirements, ensuring the existence of output values that satisfy the requirements.
6. claim6: The RTS system is specified using Cryptol, creating an executable model that refines the architecture and requirements.
7. claim7: The correctness of the Cryptol model is ensured by writing theorems (properties) in Cryptol language for each FRET requirement, proving the model satisfies the requirement.
8. claim8: The correctness of the RTS implementation is demonstrated through three strategies: correct-by-construction generation of components, formal static verification of C source code, and runtime verification (testing) with

```
claim5: object(FRET),
property(realizability_checker_used_to_verify(correctness_of(requirements))),
property(ensuring_existence_of(output_values_that_satisfy(requirements))).
claim6: object(RTS_system), property(specified_using(Cryptol)),
property(creating(executable_model_that_refines(architecture_and_requiremen
ts))).
claim7: object(Cryptol_model),
property(correctness_ensured_by(writing_theorems(properties)_in(Cryptol_lan
guage_for(each_FRET_requirement)))),
property(proving(model_satisfies(requirement))).
claim8: object(RTS_implementation),
```


Distribution Statement "A": Approved for Public Release, Distribution Unlimited.
If you have any questions, please contact the DARPA Public Release Center (PRC)

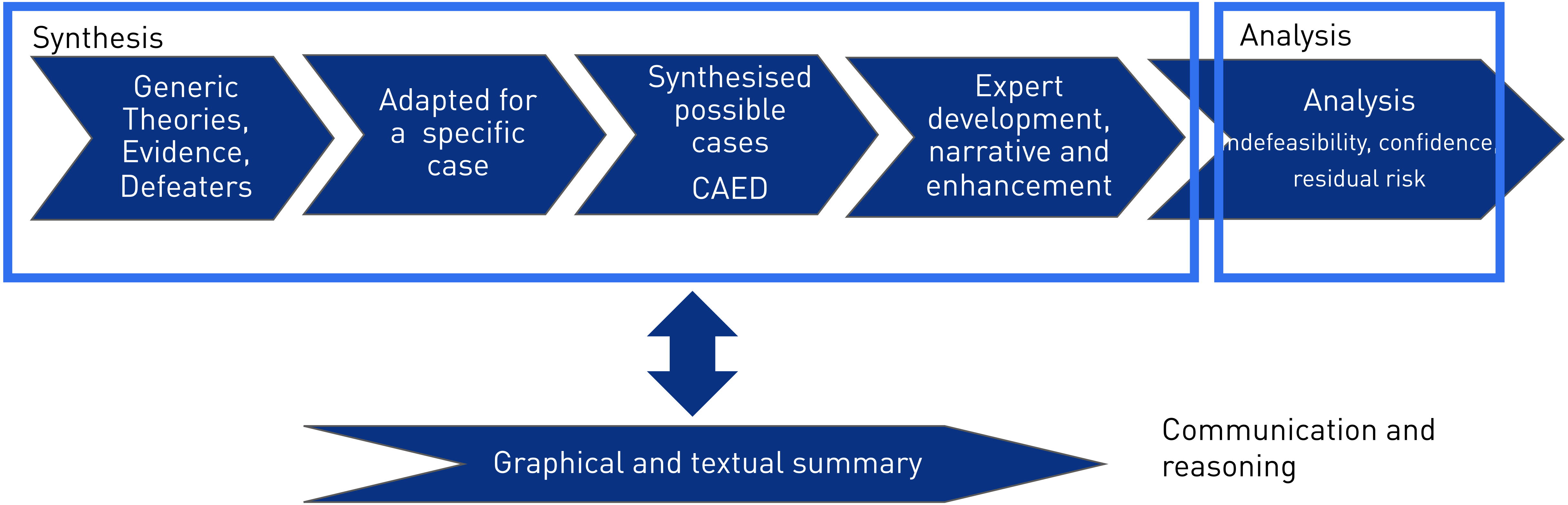
ADELARD part of nccgroup

CONTINUOUS ASSURANCE INTEGRATION



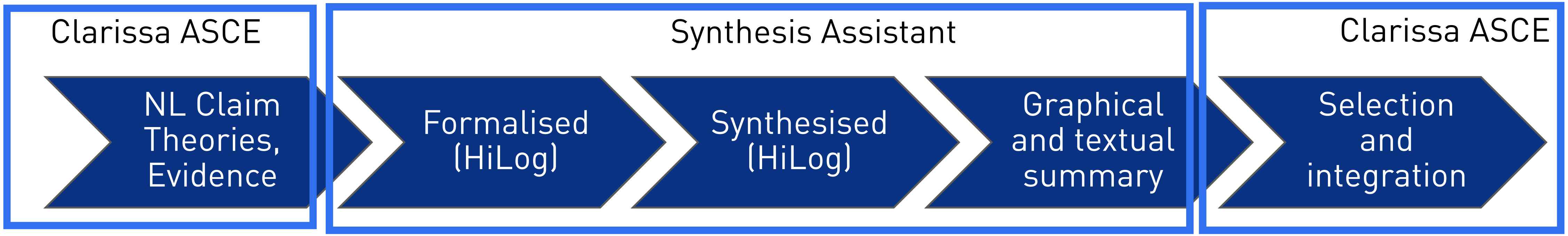
AUTOMATION AND RIGOR IN ASSURANCE CASES

Idealized workflow from generic theories to final case and judgement

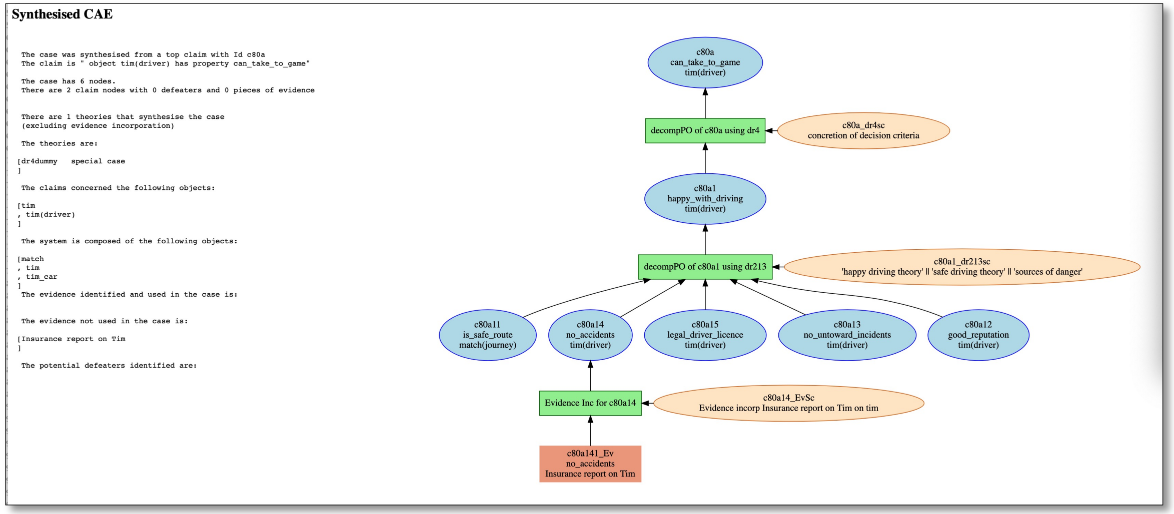


ASSURANCE CASE SYNTHESIS

Synthesis Assistant is a research tool designed to synthesize claims, arguments and evidence structures from a root or top-level claim.



- Given:
  - Top-level claim (defined in ErgoAI or node imported from an ASCE file)
  - Definition of the system structure
  - Possible defeaters
  - Theories used to develop the case
  - Evidences for the case
  - LLM support



SUPPORTING EVALUATION AND COMMUNICATION

- Shift review effort to
  - Understanding theories
  - Assess their relevance and validity
  - Trust in tools
- Complexity reduction
  - Benefits increase with size of case
  - (Experimentation)
- Generate all cases wrt a constraint
  - Select on cost or some psychological complexity metric
- Checks for
  - Unused evidence, components

Summary of case

```
 The case was synthesised from a top claim with Id c2000
 The claim is " object sys2(sys) has property is_dependable"

 The case has 31 nodes.

 The theories used to synthesise the case are:

tr1   Split of time into epochs of now and future
tr2000   Split of time into epochs of now and future
tr2001   Double decomp Platform Application
tr2001a   Split on hw/sw
tr2004   Model relating confidence in zero defects to reliability
tr2005   Model relating OpEX and defects to reliability
tr2006   Conservative substitution - WCT
tr5   Concretion of timeliness to time response

 The evidence identified and used in the case is:

opex_report
wct_analysis_report


 The potential defeaters identified are:

Known vulnerabilities in this platform
These models have onerous assumptions
wct not feasible for codebase size
```

THE HARDENS SAFETY CASE STUDY

THE RTS CASE STUDY

HARDENS

- **HARDENS** (*High Assurance Rigorous Digital Engineering for Nuclear Safety*) is a R&D project run by Galois for the *Nuclear Regulatory Commission* (NRC)
- the purpose of HARDENS is to demonstrate and educate about cutting-edge, high-assurance model-driven engineering
  - our focus is on nationally critical infrastructure, and thus safety-critical embedded systems
- within HARDENS, Galois has designed and built a demonstration Reactor Trip System (RTS) that is representative of a Digital Instrumentation & Control (DI&C) system for a Nuclear Power Plant (NPP)
  - the RTS is fault-tolerant and high-assurance
  - the RTS has a physical manifestation (an FPGA board plus sensors/actuators) and a set of digital twins

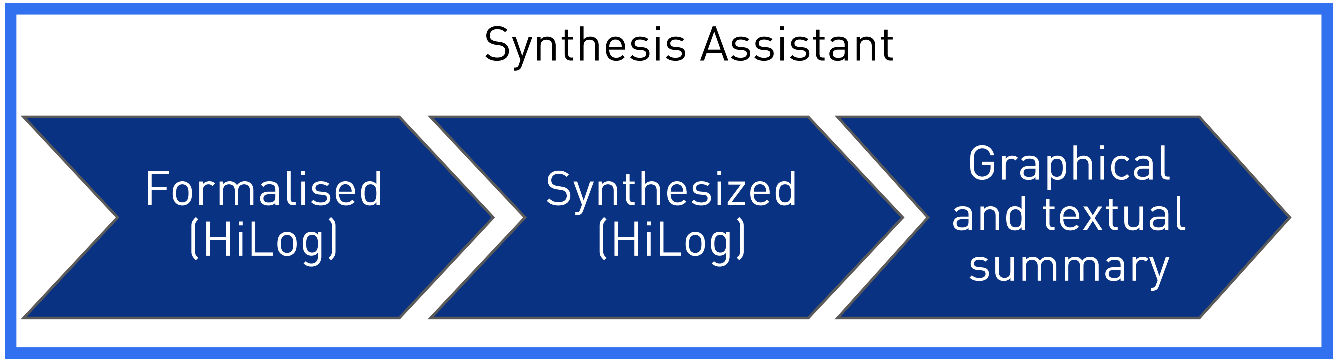
CASE STUDY

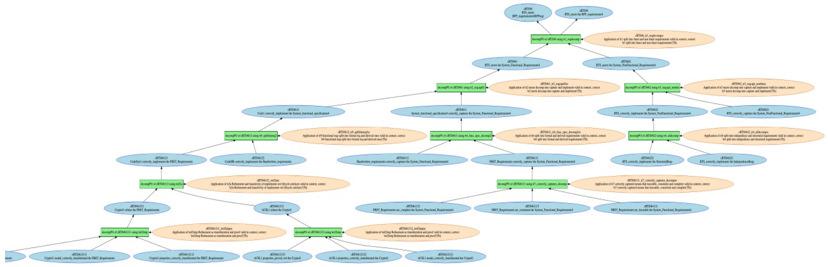
- Hardens demonstrates wide range of options
  - Impressive demo of capability
  - Options for deployment, proposed deployment
  - Not all complete
- Evidence
  - Artifacts in traditional sense not present, but instructions on how to generate them
- Attempted rationale reconstruction
  - Always hard
    - even for well thought through and extensively documented project

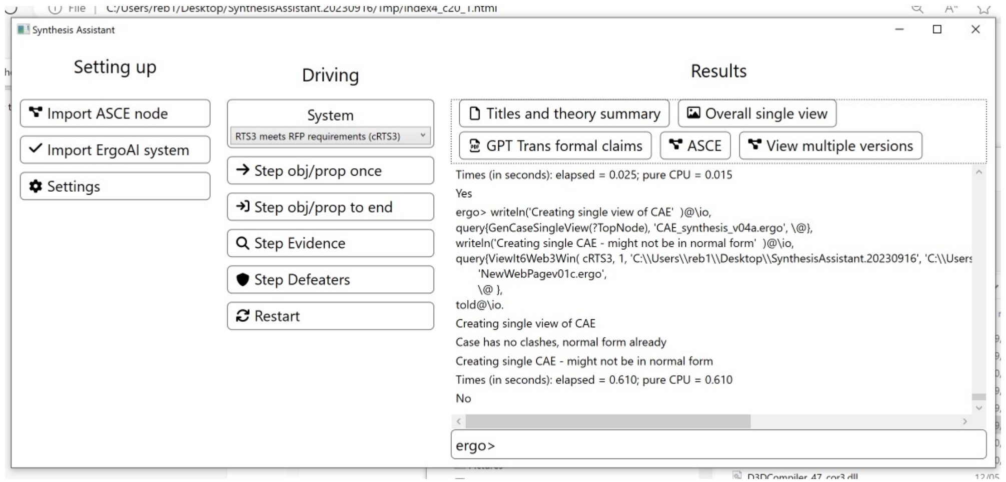
HARDENS

- **HARDENS** (*High Assurance Rigorous Digital Engineering for Nuclear Safety*) is a R&D project run by Galois for the *Nuclear Regulatory Commission* (NRC)
- the purpose of HARDENS is to demonstrate and educate about cutting-edge, high-assurance model-driven engineering
  - our focus is on nationally critical infrastructure, and thus safety-critical embedded systems
- within HARDENS, Galois has designed and built a demonstration Reactor Trip System (RTS) that is representative of a Digital Instrumentation & Control (DI&C) system for a Nuclear Power Plant (NPP)
  - the RTS is fault-tolerant and high-assurance
  - the RTS has a physical manifestation (an FPGA board plus sensors/actuators) and a set of digital twins

SYNTHESIS








```
The theories used are:

'tr1_reqdecomp   tr1 split into funct and non-funct requirements'
'tr2_reqcaptf2   tr2 meets decomp into capture and implement'
'tr3_reqcapt_nonfun   tr3 meets decomp into capture and implement'
'tr4_nfdecomp   tr4 split into indepedence and structural requirem
'tr6_func_spec_decomp2   tr6 split into formal and derived require
'tr7_correctly_captures_decomp   tr7 correctly captured means that
'tr9_splitfunreq2   tr9 functional reqs split into formal req and
'tref2a   tr2a Refinement and transitivity of implements wrt lifey
'tref2imp   tref2imp Refinement as transliteration and proof'

The claims concerned the following objects:

ACSL1(design)
Code1(code)
CodeHR(code)
CodeSyn1(code)
Cryptoll(spec)
FRET_Requirements(formal_fun_req)
Handwritten_requirements(derived_req)
IndependenceReqs(indepedence_req)
RFP_requirements4(RFPreq)
StructuralReqs(structural_req)
System_Functional_Requirements4(fun_req)
System_NonFunctional_Requirements4(nonfun_req)
System_functional_specification4(fun_spec)
```

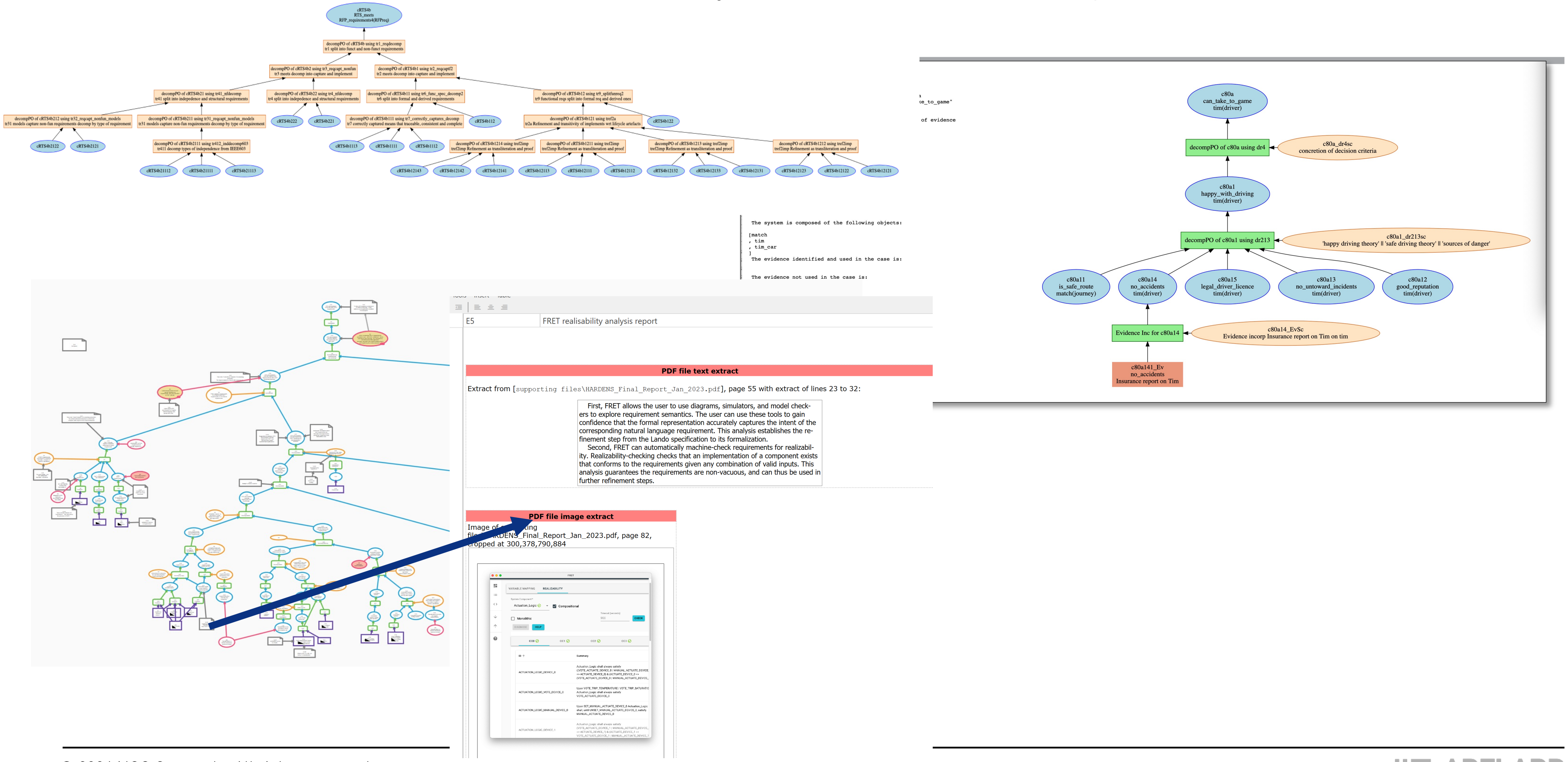

ADELARD part of nccgroup

SUMMARY - THEORY VIEW

- Presented case in terms of theories and CAE Blocks used
  - Support understanding of rationale
  - Importance of abstraction
- Core of case can be explained with 10 generic theories
- Some specific additions
  - Synthesis/handwritten
  - Implementation software/hardware

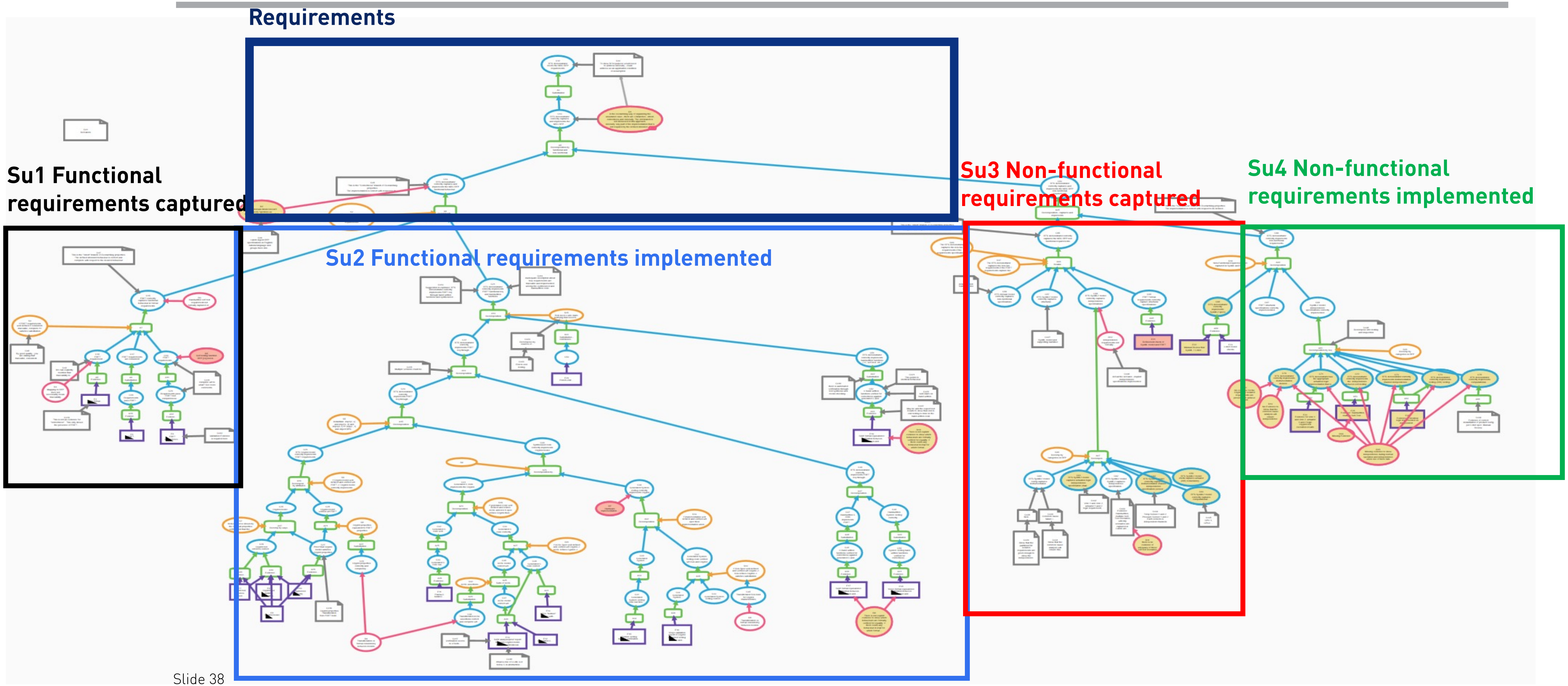
tr1 split into funct and non-funct requirements
tr2 meets decomp into capture and implement
tr31 models capture non-fun requirements decomp by type of requirement'
tr4 split into independence and structural requirements'
tr411 decomp types of independence from IEEE603'
tr6 split into formal and derived requirements'
tr7 correctly captured means that traceable, consistent and complete'
tr9 functional reqs split into formal req and derived ones'
tr2a Refinement and transitivity of implements wrt lifeycle artifacts'
tref2imp Refinement as transliteration and proof'

Different types of summary views and narrative to provide the overall story and nuances

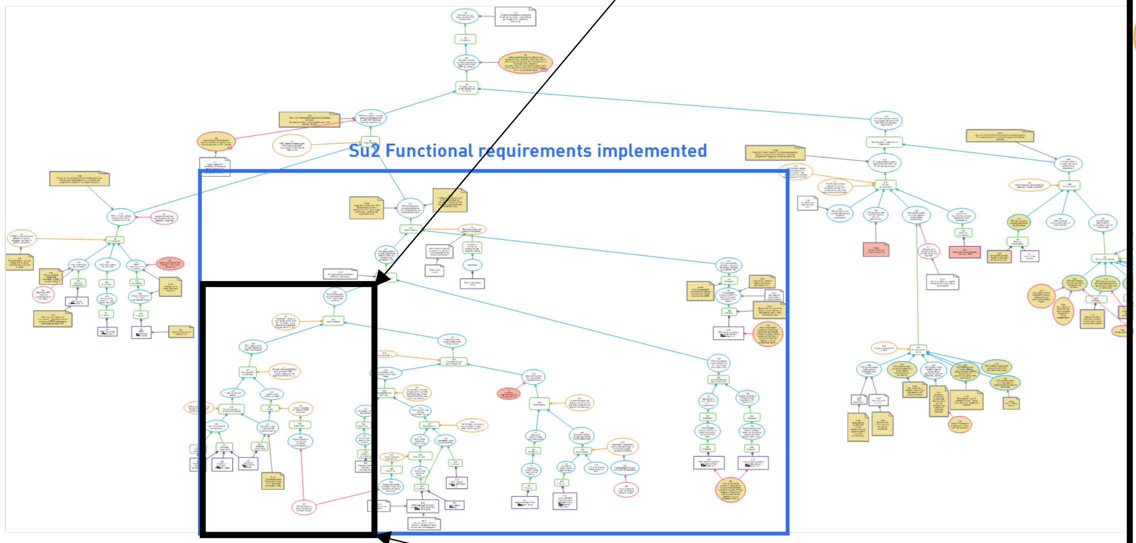


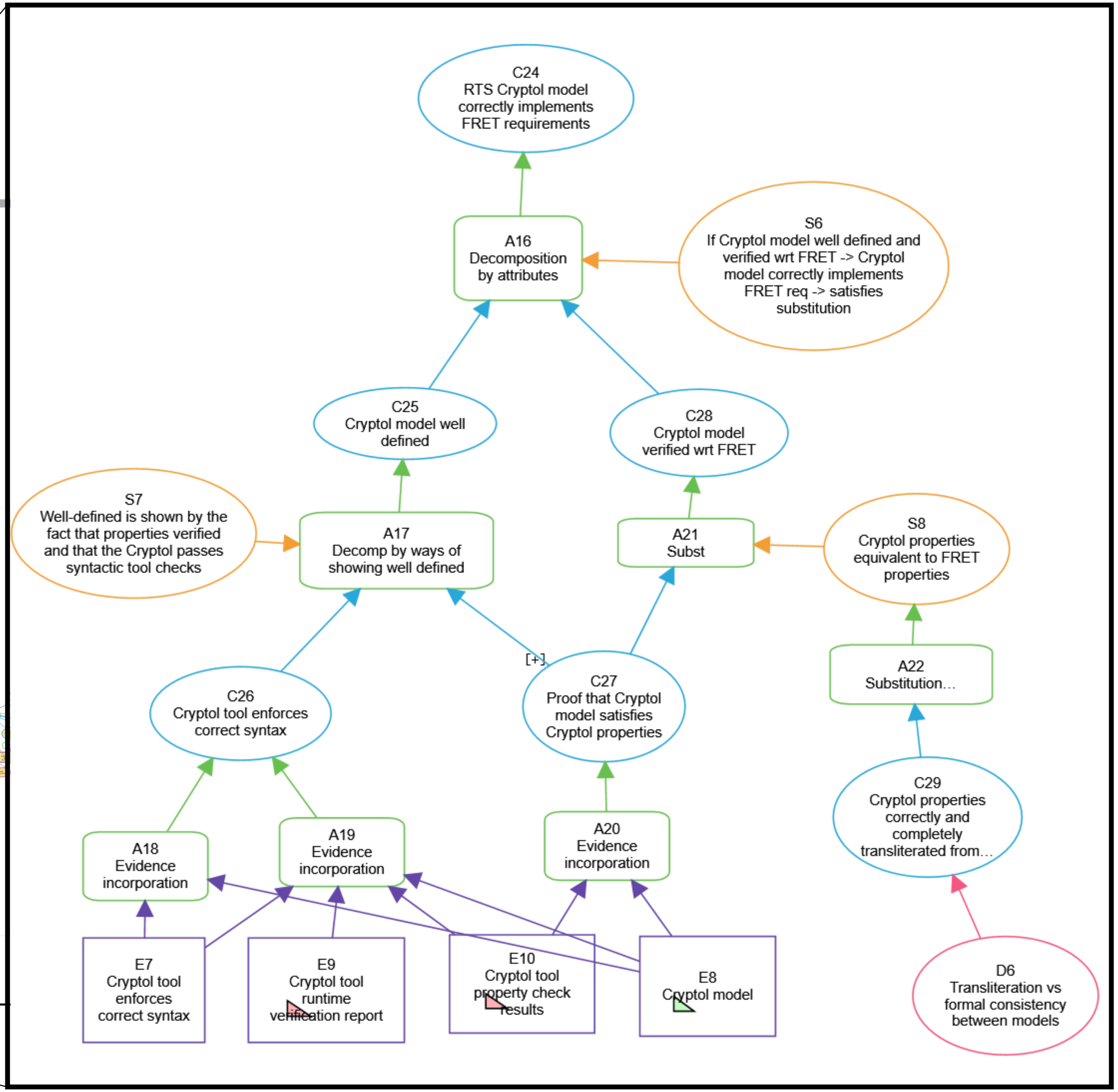
OVERVIEW

Want to produce understanding not wallpaper



DEEP DIVES – CRYPTOL MODEL





DEFEATERS AND OTHER ISSUES

- Assumptions
  - Evidence is assumed can be reconstructed if instructions to do so
  - Also marked evidence that can not be found but a potential report identified
- Defeater classes
  - Transliteration vs formal refinement proofs
  - Requirements and specification of handwritten code (GUI, self test)
  - Dealing with independence requirements
  - Traceability vs mapping FRET to RFP
- Have identified some areas of doubt
  - Some might be due to our misunderstanding
  - Others due to scope of case study
- In Clarissa terms these show how the case has been developed and assessed

HARDENS BASED RTS CASE STUDY

- To illustrate Clarissa case methodology
  - Included – based on role of *presenting* Hardens case
    - CAE Blocks, theories, doubts and defeaters,
    - Evidence integration and some narrative
    - Views
    - Theories and synthesis
  - Not included
    - Prolog export
    - Confirmation theory – use for review or by case makers
    - Confidence propagation
    - Theories linking probability of zero defects to risk
- To support NRC and our understanding of a correct by construction case
  - To provide feedback to Clarissa on how an evaluator might use a Clarissa style case
- It is **not** to assess whether Hardens would be acceptable as an RTS

SAFETY ASSURANCE CASE FRAMEWORK (SAC) PROJECT OBJECTIVES

"to improve the efficiency and flexibility of Nuclear Regulatory Commission (NRC)'s licensing reviews of Digital Instrumentation & Controls (I&C) by enabling consistent evaluation and documentation of performance based (outcome oriented), safety focused, risk informed digital I&C licensing applications through a safety assurance case (SAC) approach"

PROJECT APPROACH

The assurance case approach will build on Assurance 2.0

The work will build on the approach developed within the DARPA Clarissa project part of the Arcos program (Automated Rapid Certification Of Software)

The focus of the work is on digital I&C safety systems of the highest criticality

A Hardens-based case study will be used throughout to support the understanding of the approach and to provide concrete examples

ADELARD part of nccgroup

CONCLUSIONS

- Assuring transformative technologies
  - Tempo, scope and focus on behavior
- Assurance cases
  - Not just pictures, narrative and justification, understanding and communication
- Assurance 2.0
  - Updated and more rigorous approach
  - Supports synthesis
  - Formal methods example
- Transition project with NRC
  - Formal methods based assurance
    - higher assurance at lower cost in less time, less uncertainty?

Prof Robin E Bloomfield FREng

Adelard (part of NCC Group)

and City, University of London

r.e.bloomfield@city.ac.uk

robin.bloomfield@nccgroup.com

Assurance 2.0 joint work with John Rushby, SRI

ADELARD