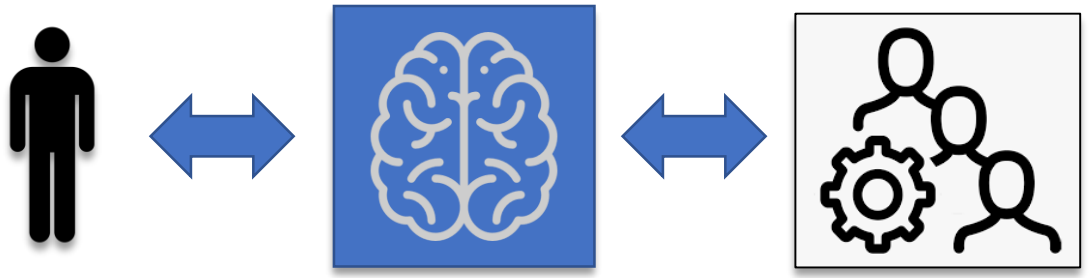
Justifying trust in *AI/ML system* using Engineering Models and Assurance Cases

Morayo ADELJOUMA[1,3], Florent CHENEVIER[1,5], Georges JAMOUS[1,4], **Eric JENN[1,2]**, Vincent MUSSOT[1,2]

IRT System-X and (2) IRT St-Exupery, (3) CEA , (4) APSYS, (5) Thales AVS

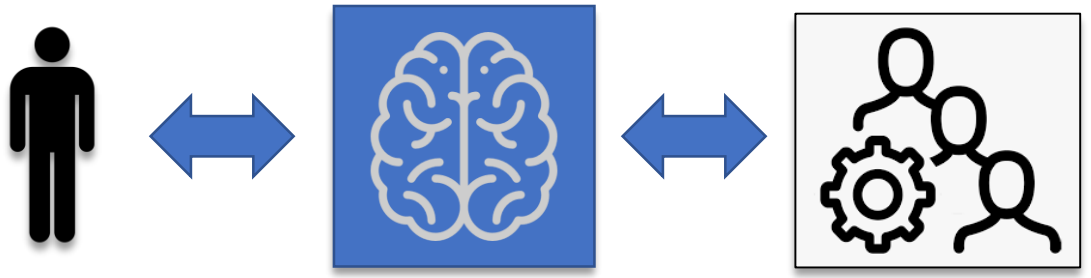
Contact: eric.jenn@irt-saintexupery.com

Objective



The system will work as required and not cause harm.

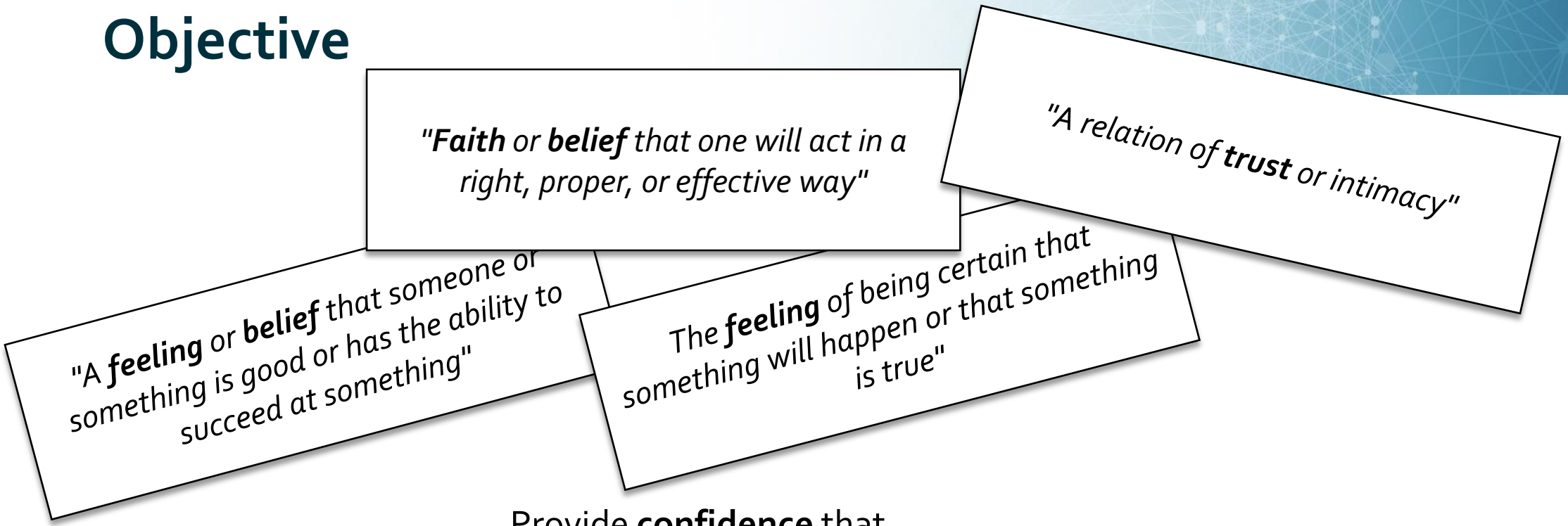
Objective



Provide **confidence** that…

The system will work as required and not cause harm.

Objective



Provide **confidence** that…

The system will work as required and not cause harm.

Objective

Confidence is based on **justified** **beliefs** about the system and its environment

Provide **confidence** that…

The system will work as required and not cause harm.

Objective

Justification can be developed and documented [...] as a **structured argument grounded on evidence**.

Confidence is based on **justified beliefs** about the system and its environment

Provide **confidence** that...

The system will work as required and not cause harm.

J. Rushby, "Assurance and Assurance cases"

Assurance cases

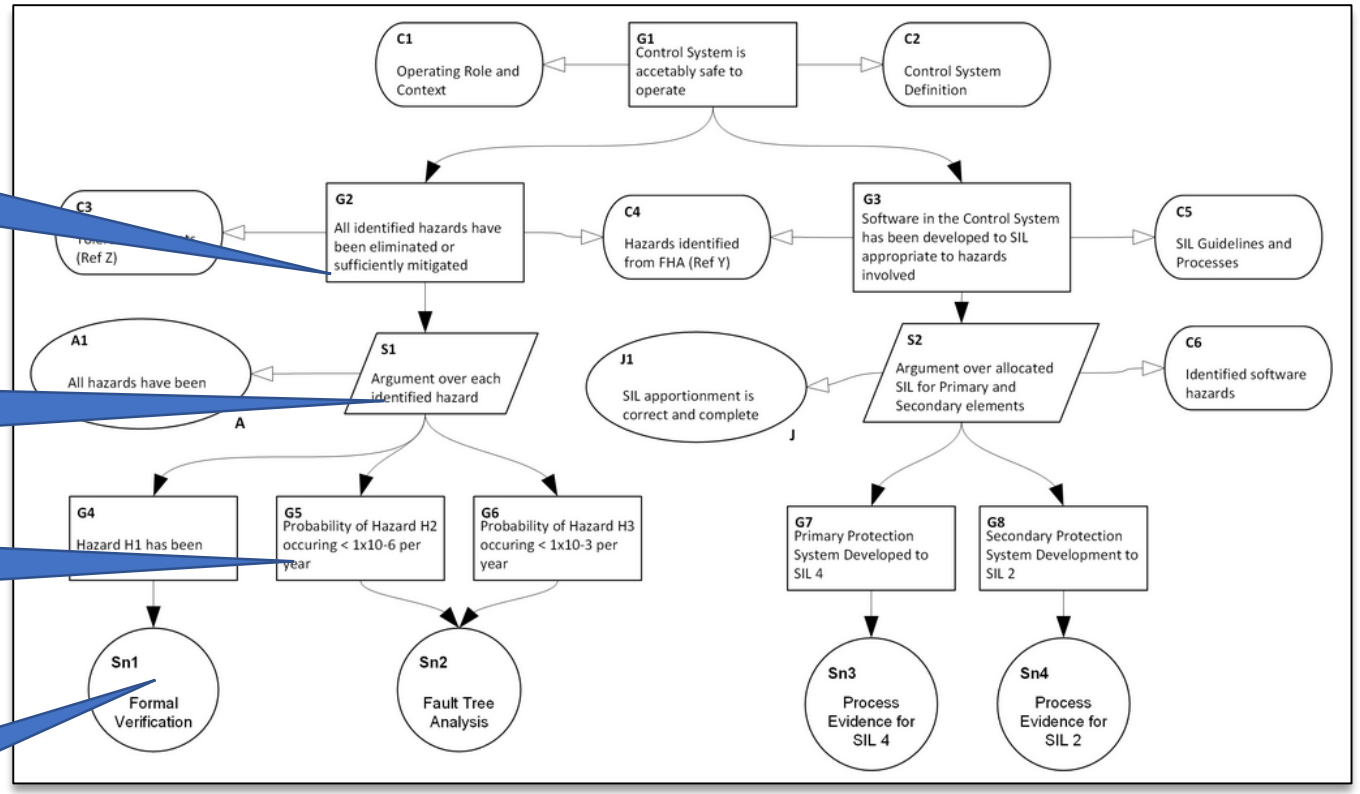
Justification can be developed and documented [...] as a **structured argument grounded on evidence**.

A goal

A strategy

A "sub-goal"

A solution (providing some evidence)



(From the GSN standard)

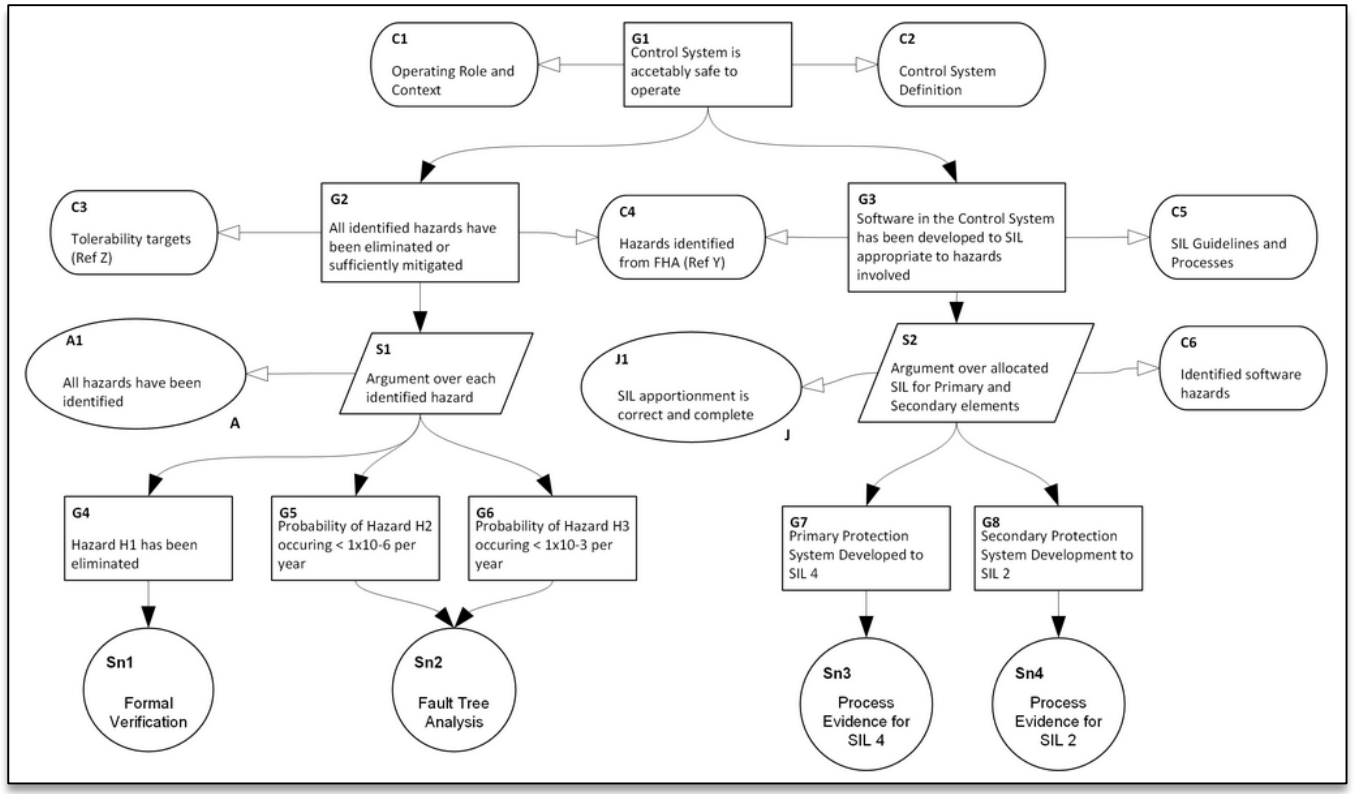
Objective

Justification can be developed and documented [...] as a **structured argument grounded on evidence**.

Is this the current practice?

Is there a specific need for IA?

- There is a need to help end-user choose appropriate (but new) techniques in regards to actual risks
- There is an opportunity
  - New issues requires new engineering practices...
  - Formalizing engineering practice

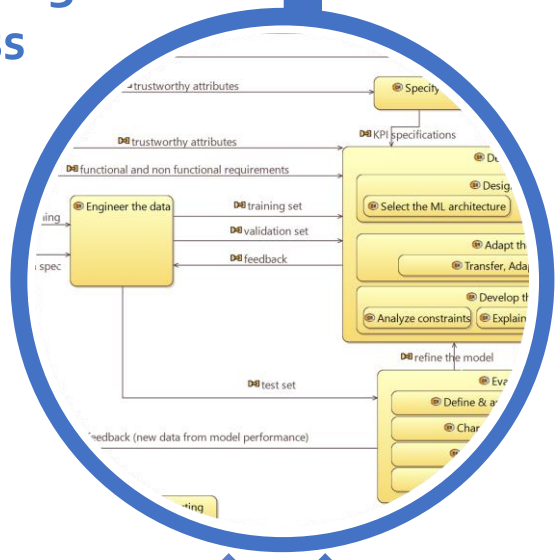


(From the GSN standard)

From the engineering process to the assurance case and vice-versa

Development activities *Development artefacts* *Properties*

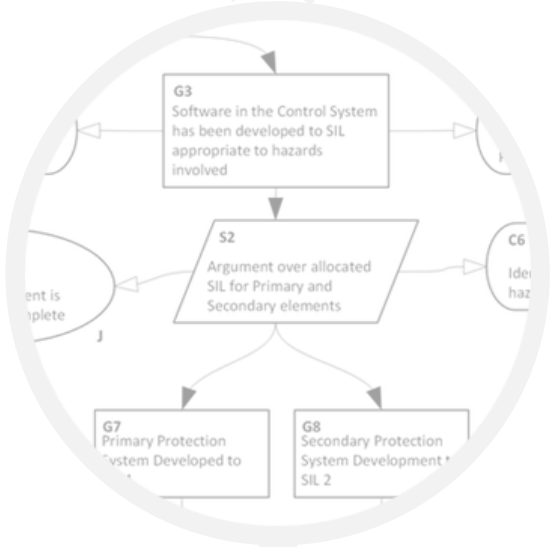
Modeling the engineering process



- What are the **activities**?
- What are the **items** consumed / produced by these activities?

?

- What are the **expected properties** of these items
- What are the "**failure modes**" of these activities ?



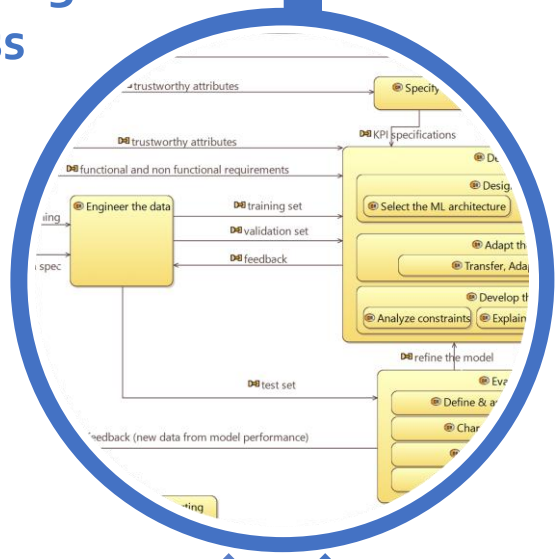
Modeling the argumentation process

Verification activities *Verification artefacts*

From the engineering process to the assurance case and vice-versa

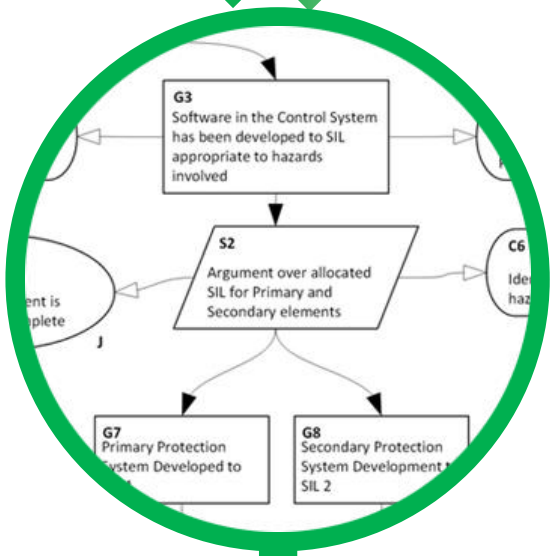
Development activities *Development artefacts* *Properties*

Modeling the engineering process



- What are the **goals**?
- What are the **strategies** to break down a goal into sub-goals?
- What are the **assumptions** ?
- What are the **solutions** to provide the **evidences** ?

- What are the **expected properties** of these items
- What are the "**failure modes**" of these activities ?

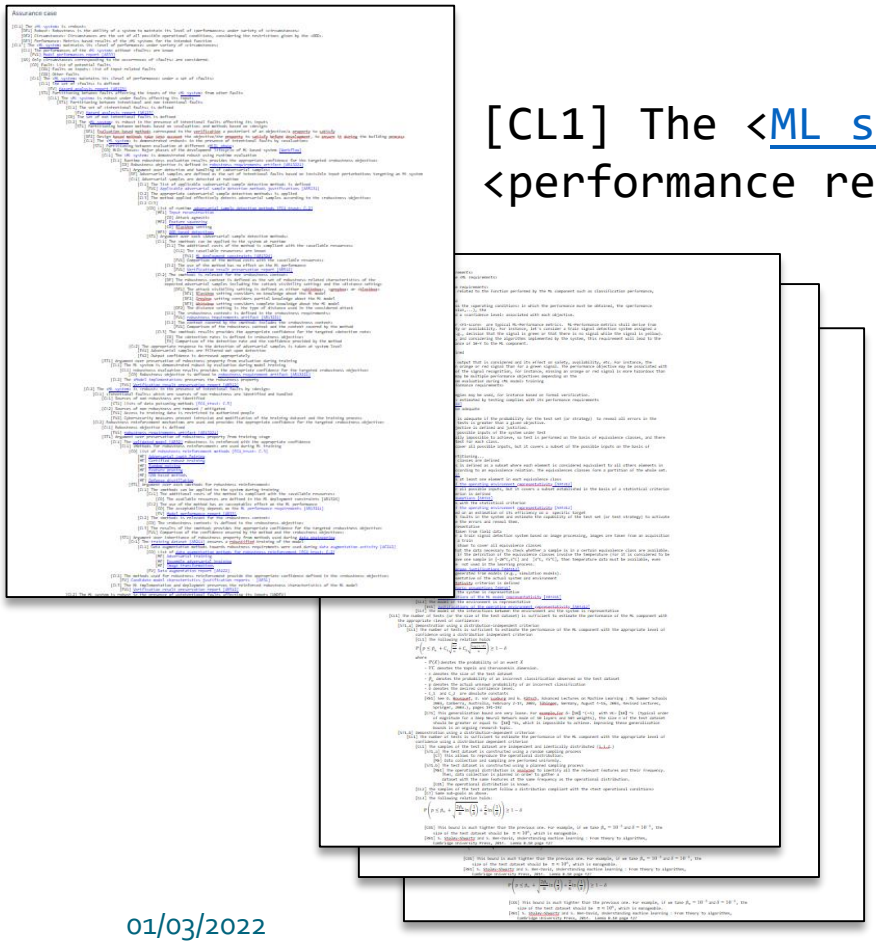


Modeling the argumentation process

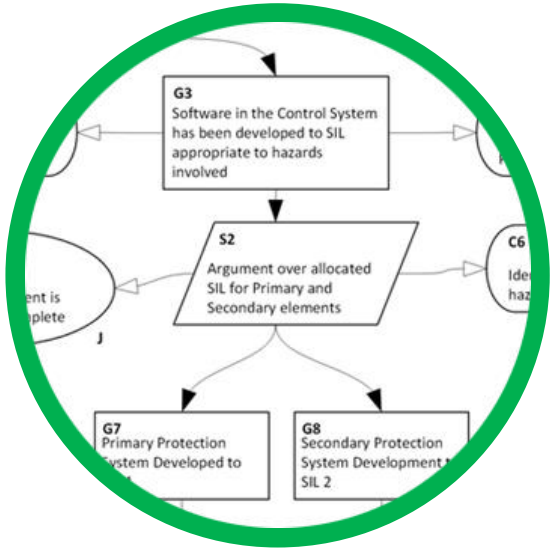
Verification activities *Verification artefacts*

From the engineering process to the assurance case and vice-versa

[CL1] The <ML system> is <robust>



[CL1] The <ML system> complies with its <performance requirements>



From the engineering process to the assurance case and vice-versa

[CL1] The <ML system> is <robust>

From the engineering process to the assurance case and vice-versa

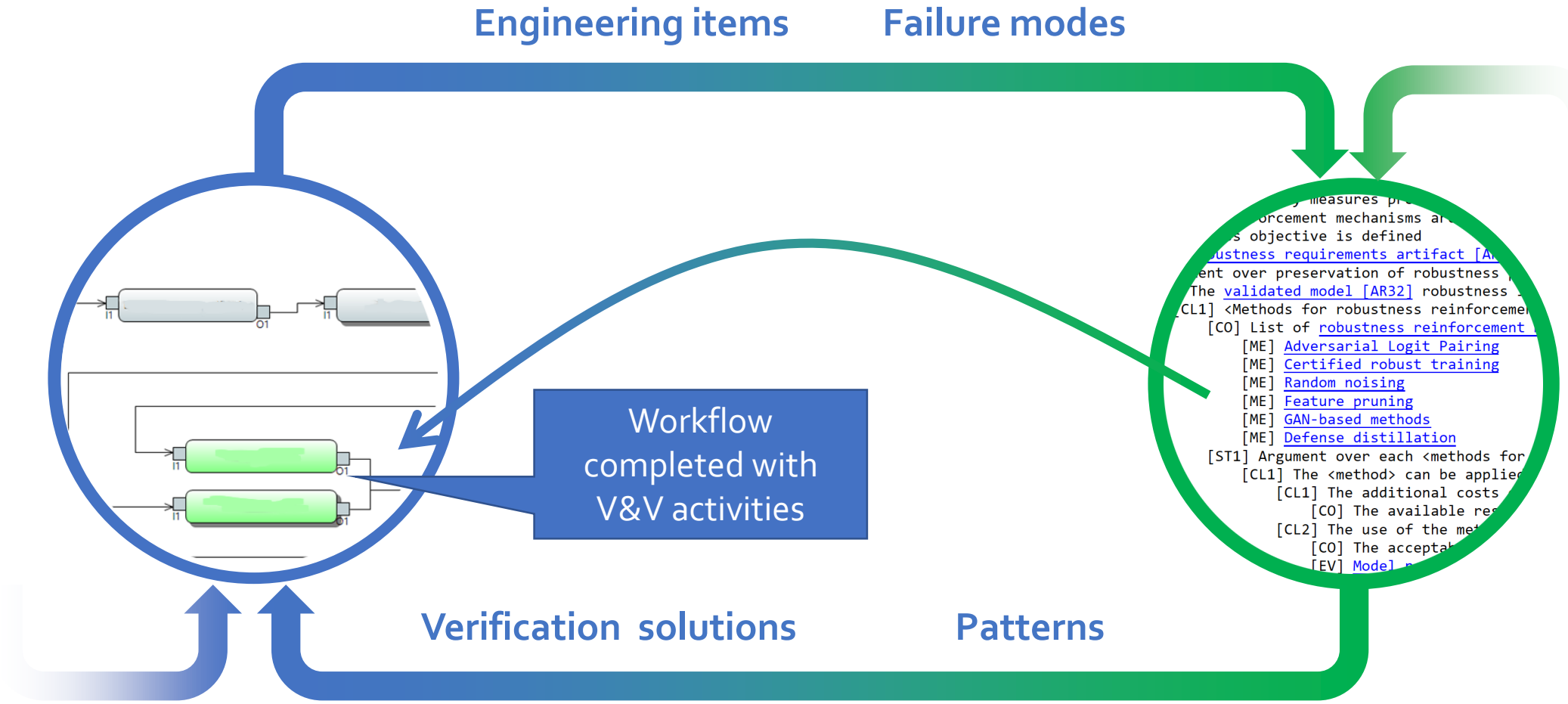
[CL1] The <ML system> is <robust>

```
[CL1] The <ML system> maintains its <level of performance> under a set of <faults>
    [CL1] The set of <faults> is defined
        [EV] Hazard analysis report [AR123]
    [ST1] Partitioning between faults affecting the inputs of the <ML system> from other faults
        [CL1] The <ML system> is robust under faults affecting its inputs
            [ST1] Partitioning between intentional and non-intentional faults
                [CL1] The set of <intentional faults> is defined
                    [EV] Hazard analysis report [AR123]
                [CO] The set of non intentional faults is defined
                [CL2] The <ML system> is robust in the presence of intentional faults affecting its
                    [ST1] Partitioning between methods based on <evaluation> and methods based on <d
```

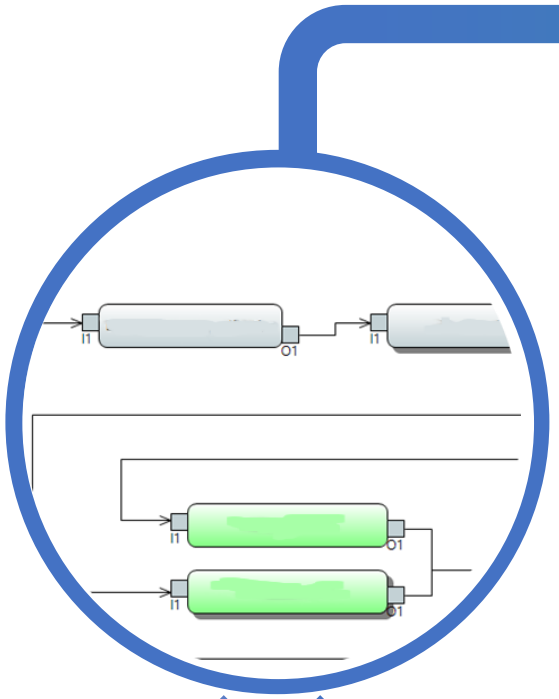
```
[CL2] The <ML system> is <robust> in the presence of intentional faults by
    [CL1] <Intentional faults> which are sources of non-robustness are iden
        [CL1] Sources of non-robustness are identified
            [CT1] Lists of data poisoning methods [EC4 trust: C.5]
        [CL2] Sources of non-robustness are removed / mitigated
            [EV1] Access to training data is restricted to authorized peopl
            [EV2] Cybersecurity measures prevent intrusion and modification
    [CL2] Robustness reinforcement mechanisms are used and provides the app
        [CL1] Robustness objective is defined
            [EV1] robustness requirements artifact [AR13221]
        [ST1] Argument over preservation of robustness property from traini
            [CL1] The validated model [AR32] robustness is reinforced with
                [CL1] <Methods for robustness reinforcement> are used durin
                    [CO] List of robustness reinforcement methods [EC4_trus
                        [ME] Adversarial Logit Pairing
                        [ME] Certified robust training
                        [ME] Random noising
                        [ME] Feature pruning
                        [ME] GAN-based methods
                        [ME] Defense distillation
```

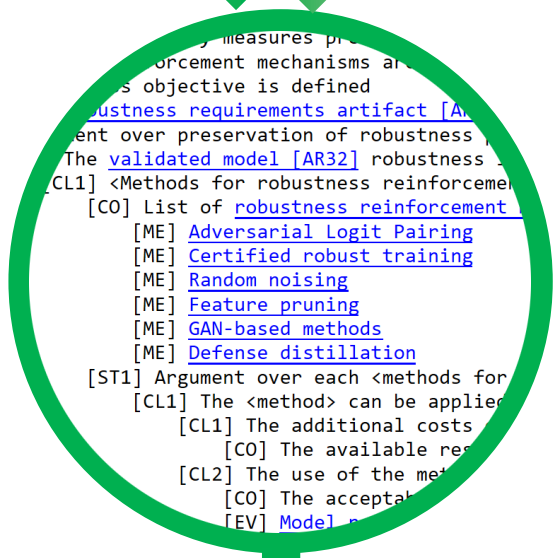
Solutions

From the engineering process to the assurance case and vice-versa

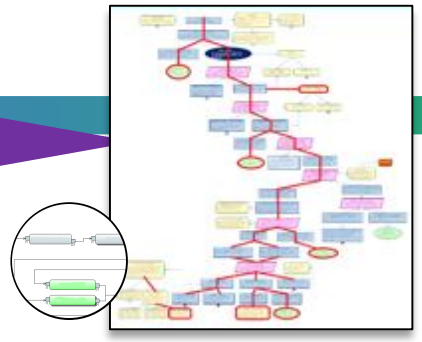


From the engineering process to the assurance case and vice-versa

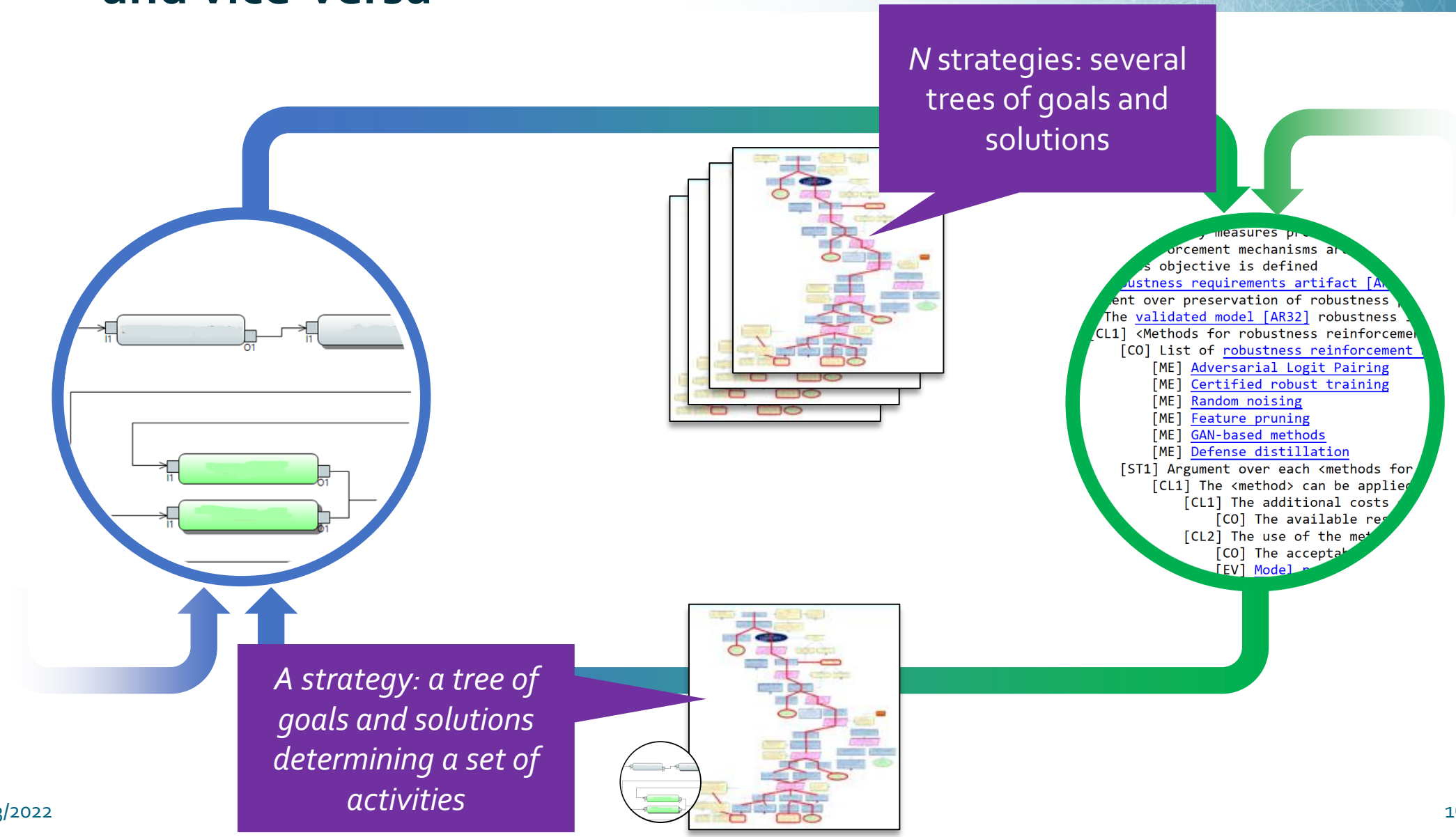




A strategy: a tree of goals and solutions determining a set of activities

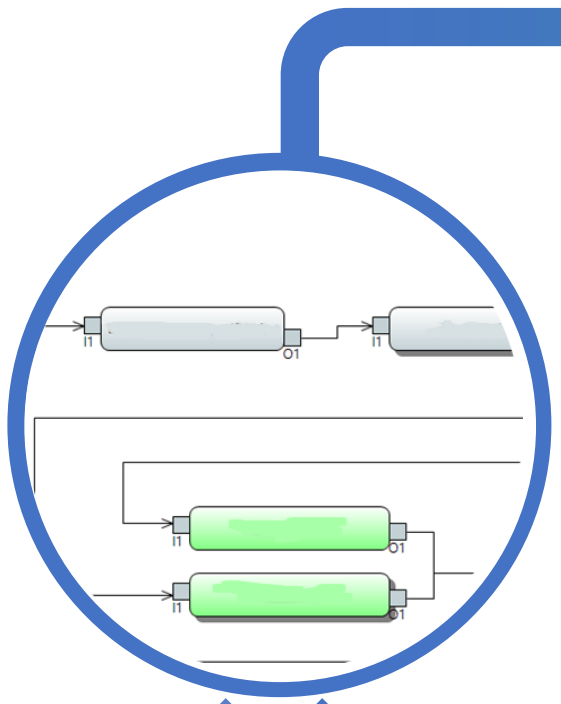


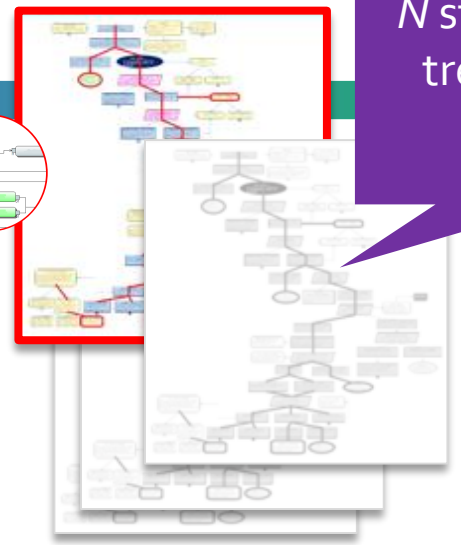
From the engineering process to the assurance case and vice-versa

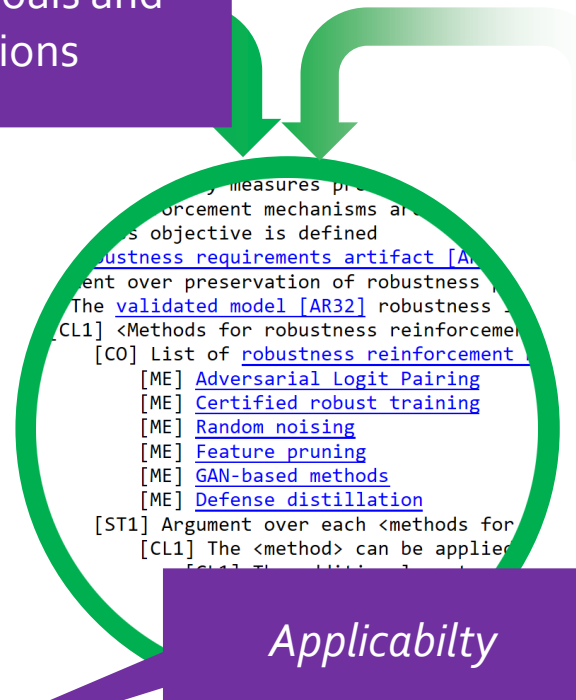


From the engineering process to the assurance case and vice-versa

N strategies: several trees of goals and solutions







Applicabilty

Method	Reference	Method description / type	Method specificities	Method quality
Adversarial training	EC4_trust: C.2	Train model with dataset augmented with adversarial samples	Concerns in priority L_inf norm-based perturbations setting Requires crafting of adversarial examples Requires lots of adversarial examples Sensible to transferable attacks	EC4_trust: Tables C.2 and C.3: Accuracy which drops down from >99% to ~ 65% can be restored to ~99% using these defences.
Ensemble adv training	EC4_trust: C.2	Train model with dataset augmented with adversarial samples from multiple sources (increases diversity of adversarial samples)	Requires the craft of adversarial samples Requires lots of adversarial examples	Increase robustness against transferred adversarial samples Increase robustness against black-box attacks
Image transformations	EC4_trust: C.2	Transformations of inputs to mitigate the effect of adversarial samples.	Less efficient for grey-box and white-box contexts Targeted at robustness enhancement for a specific attack in a specific setting	
Adversarial logit pairing	EC4_trust: C.2	Training loss modification	Less amenable to gradient descent	
Certified robust training	EC4_trust: C.3	Guarantee robustness by specific training methods	Concerns in priority L_p norm-based perturbations setting	FORMAL proof: Network is guaranteed robust to adversarial attacks within a certain
Reluplex	EC3_robustness: D.5	Method that rely on SMT-solvers to prove if the negation of a specific property is satisfied or if it can find a counter-example.	Restricted to piecewise linear activation functions	Complete method
GAN-based methods	EC4_trust: C.1.3	Modification of input image through Generator, to reduce the potential adversarial perturbations	Requires strong and expressive GAN (High training cost) Sensible to gradient masking attack or C&W white-box attacks	
Defense distillation	EC4_trust: C.1.1	Leverage distillation techniques to hides the gradient between the pre-softmax layer (logits) and softmax outputs	[EC4_trust: Carlini2017b] can be bypassed for any norm (0-inf) Sensible to transferable attacks	
Input reconstruction	EC4_trust: C.6.2	Adversarial sample detection method and reconstruction of normal input		Model-agnostic (can be combined with other defenses) Attack-agnostic
Feature squeezing	EC4_trust: C.6.1	Adversarial sample detection method, based on color alteration and spatial smoothing.	Specific methods (EAD and l2-norm C&W) can bypass this detection	

Limitations

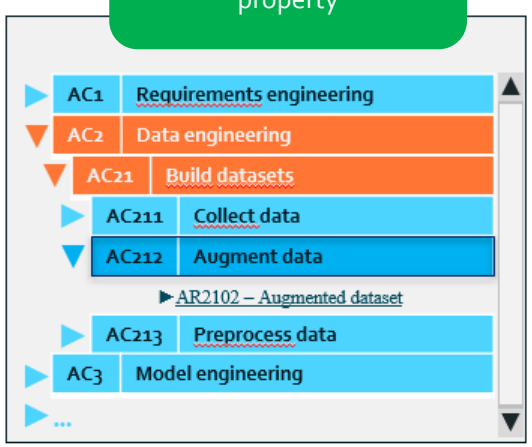
An "artist's view" of the Confiance.ai Workbench

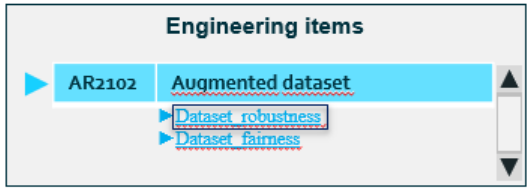
Display the **assurance case** associated with the property

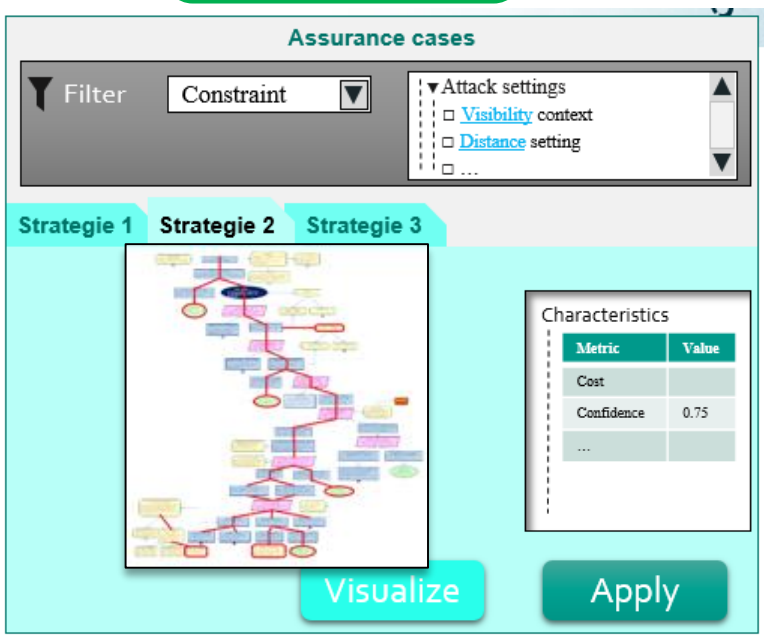
Display **applicable constraints**

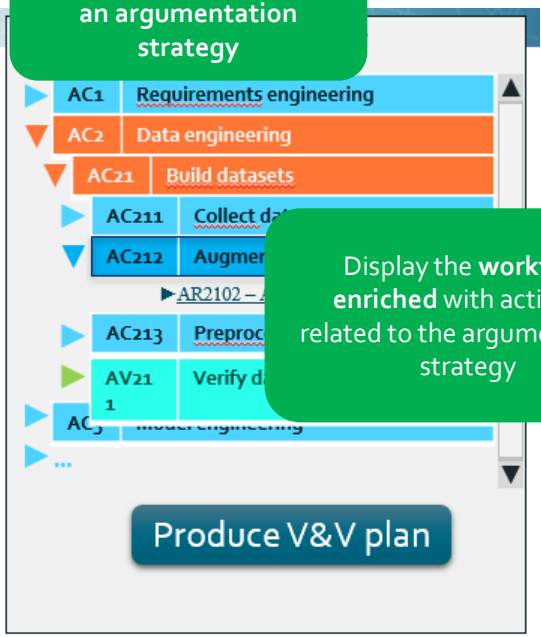
Provide indicators to support the **selection of an argumentation strategy**

Display the initial **workflow activities** and engineering items



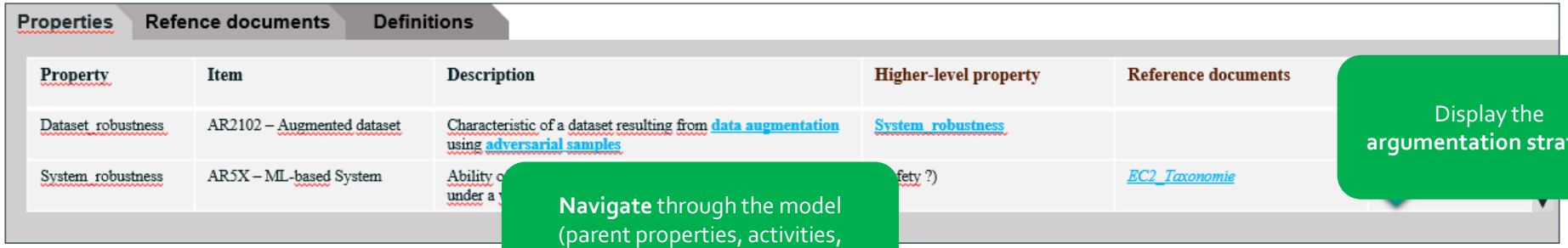






Display the **workflow enriched** with activities related to the argumentation strategy

Display **properties** applicable to an item



Display the **argumentation strategy**

Navigate through the model (parent properties, activities, engineering items, etc)

Next steps

- On-going
  - Systematic modeling of engineering activities and workflows
  - Systematic building of the AC in relation with the workflows
  - Demonstration of a V&V strategy design environment
- Future
  - Automation of V&V activities in relation with the workflow and strategies