



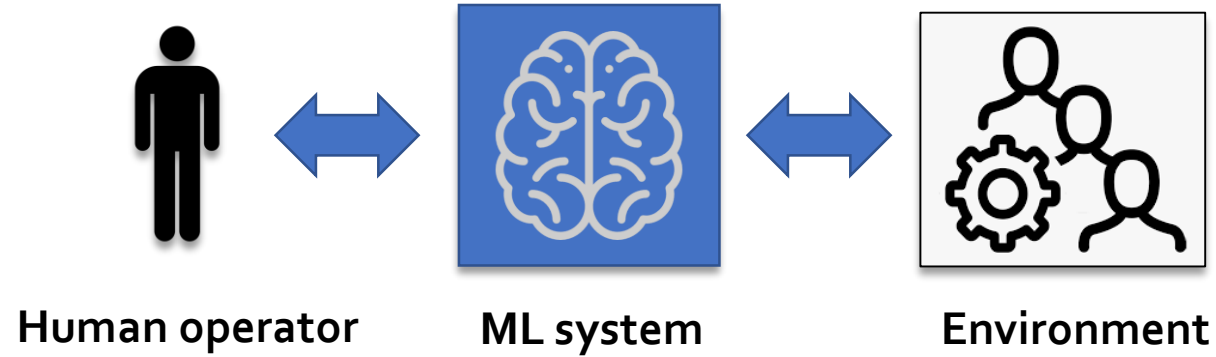
Justifying trust in *AI/ML* *system* using Engineering Models and Assurance Cases

Morayo ADELJOUA^{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

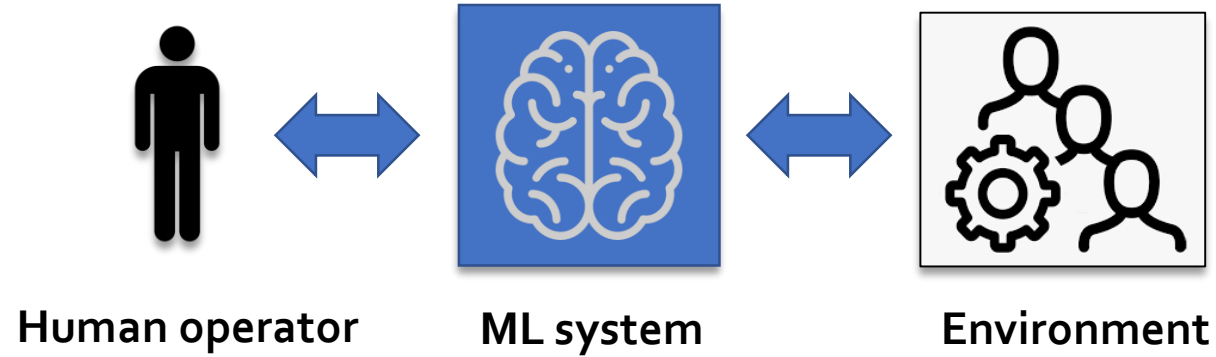
Contact: eric.jenn@irt-saintexupery.com

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

"Faith or belief that one will act in a right, proper, or effective way"

"A relation of trust or intimacy"

"A feeling or belief that someone or something is good or has the ability to succeed at something"

The feeling of being certain that something will happen or that something is true"

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

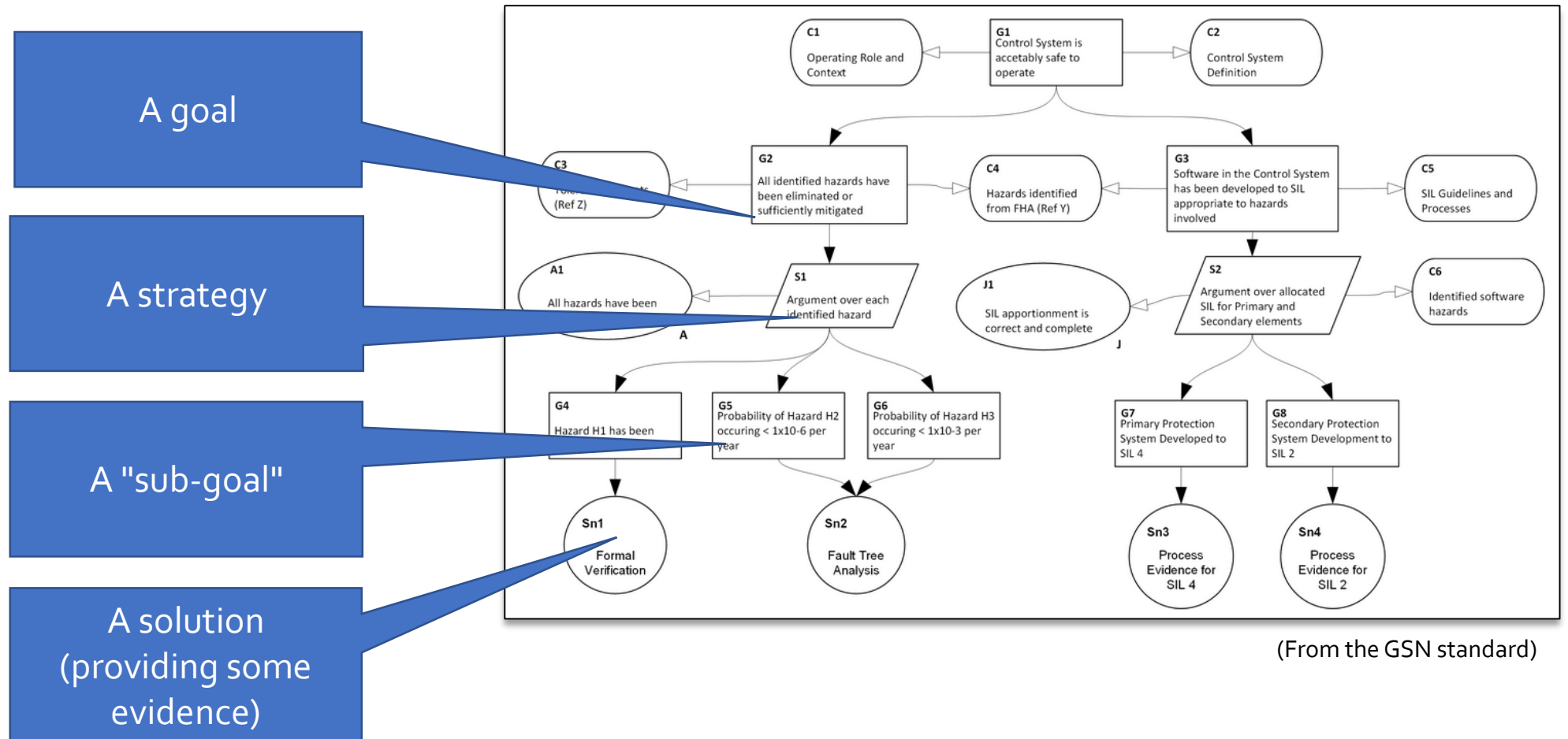
Provide **confidence** that...

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**.



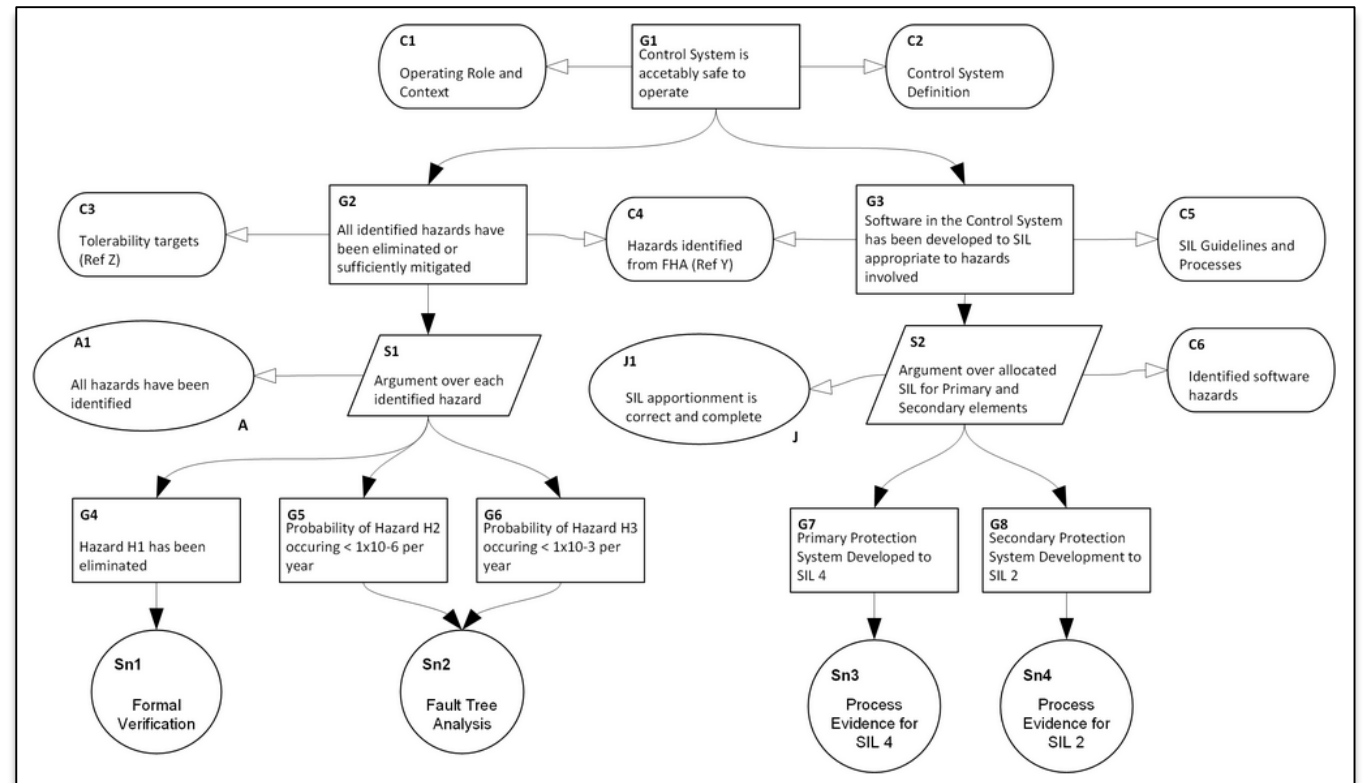
Objective

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

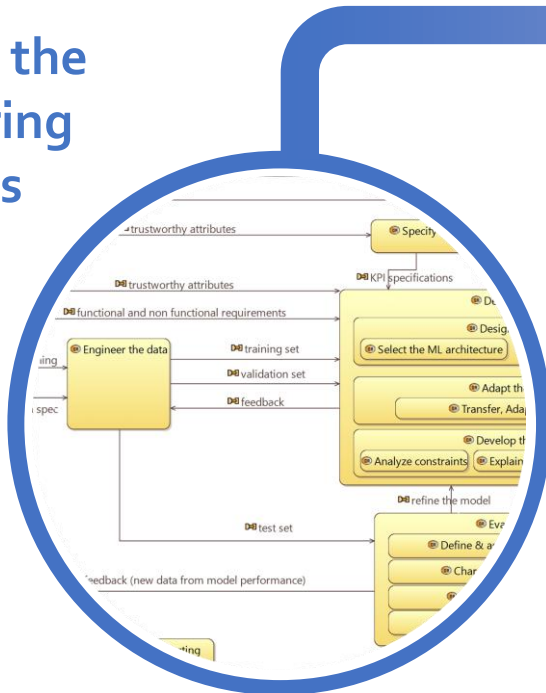


(From the GSN standard)

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

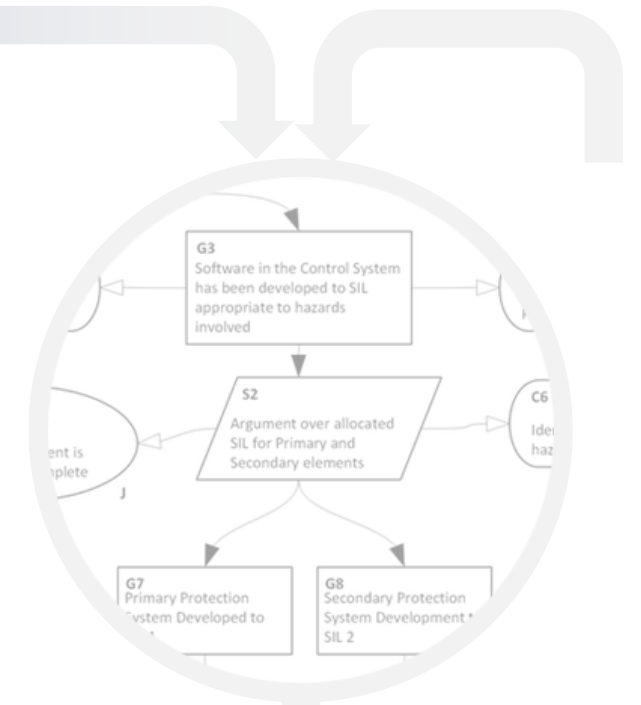
Development activities Development artefacts Properties

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

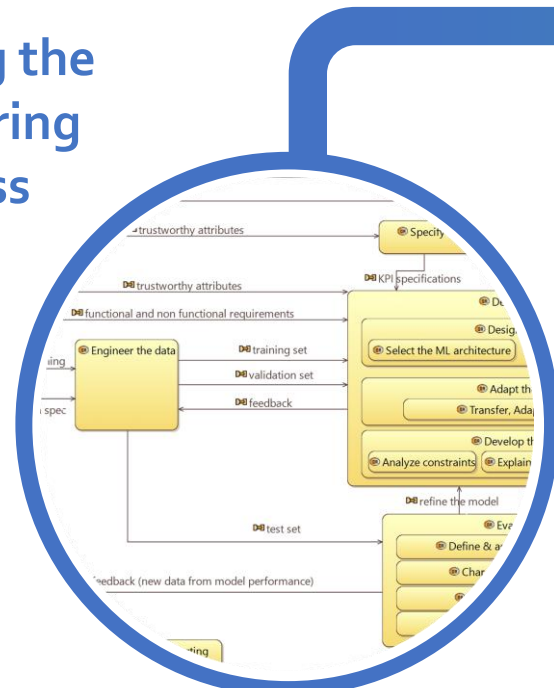
Verification activities

Verification artefacts

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

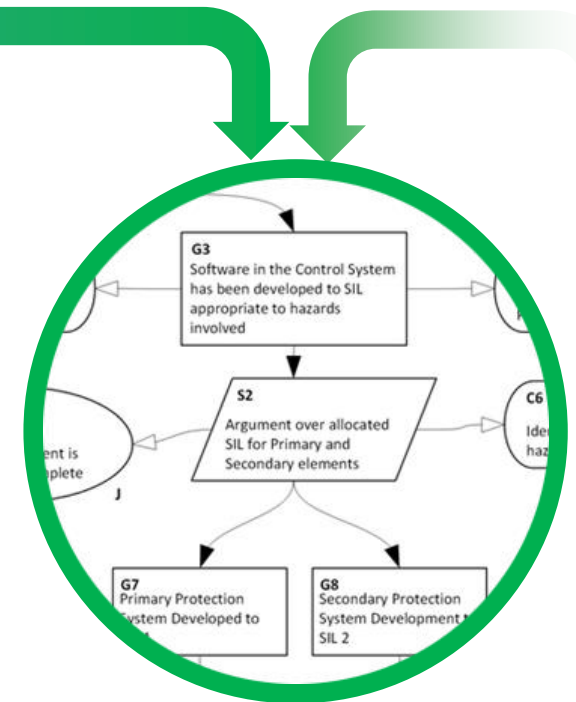
Development activities Development artefacts Properties

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

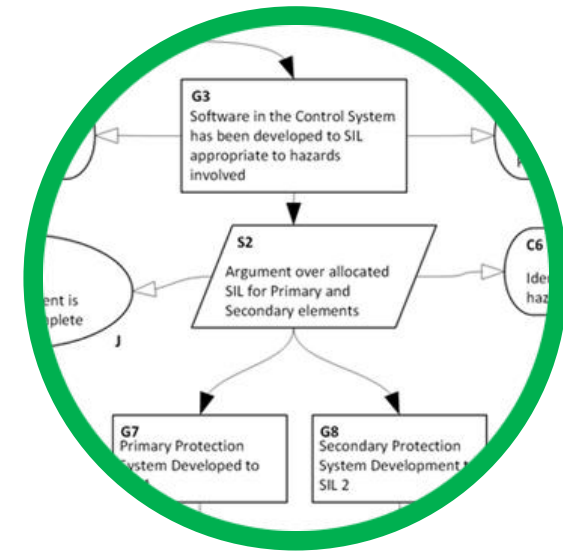
[CL1] The <ML system> is <robust>

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

```
Requirements
[REQ-001] The software shall be designed to meet the functional requirements of the system.
[REQ-002] The software shall be designed to meet the performance requirements of the system.
[REQ-003] The software shall be designed to meet the safety requirements of the system.
[REQ-004] The software shall be designed to meet the security requirements of the system.
[REQ-005] The software shall be designed to meet the maintainability requirements of the system.
[REQ-006] The software shall be designed to meet the interoperability requirements of the system.
[REQ-007] The software shall be designed to meet the compatibility requirements of the system.
[REQ-008] The software shall be designed to meet the portability requirements of the system.
[REQ-009] The software shall be designed to meet the scalability requirements of the system.
[REQ-010] The software shall be designed to meet the flexibility requirements of the system.
[REQ-011] The software shall be designed to meet the extensibility requirements of the system.
[REQ-012] The software shall be designed to meet the modifiability requirements of the system.
[REQ-013] The software shall be designed to meet the testability requirements of the system.
[REQ-014] The software shall be designed to meet the debuggability requirements of the system.
[REQ-015] The software shall be designed to meet the recoverability requirements of the system.
[REQ-016] The software shall be designed to meet the fault tolerance requirements of the system.
[REQ-017] The software shall be designed to meet the error handling requirements of the system.
[REQ-018] The software shall be designed to meet the logging requirements of the system.
[REQ-019] The software shall be designed to meet the reporting requirements of the system.
[REQ-020] The software shall be designed to meet the documentation requirements of the system.
[REQ-021] The software shall be designed to meet the configuration management requirements of the system.
[REQ-022] The software shall be designed to meet the version control requirements of the system.
[REQ-023] The software shall be designed to meet the backup requirements of the system.
[REQ-024] The software shall be designed to meet the recovery requirements of the system.
[REQ-025] The software shall be designed to meet the disaster recovery requirements of the system.
[REQ-026] The software shall be designed to meet the business continuity requirements of the system.
[REQ-027] The software shall be designed to meet the risk management requirements of the system.
[REQ-028] The software shall be designed to meet the compliance requirements of the system.
[REQ-029] The software shall be designed to meet the regulatory requirements of the system.
[REQ-030] The software shall be designed to meet the industry standards requirements of the system.
[REQ-031] The software shall be designed to meet the best practices requirements of the system.
[REQ-032] The software shall be designed to meet the state-of-the-art requirements of the system.
[REQ-033] The software shall be designed to meet the cutting-edge requirements of the system.
[REQ-034] The software shall be designed to meet the leading-edge requirements of the system.
[REQ-035] The software shall be designed to meet the next-generation requirements of the system.
[REQ-036] The software shall be designed to meet the future-proof requirements of the system.
[REQ-037] The software shall be designed to meet the long-term requirements of the system.
[REQ-038] The software shall be designed to meet the sustainable requirements of the system.
[REQ-039] The software shall be designed to meet the green requirements of the system.
[REQ-040] The software shall be designed to meet the ethical requirements of the system.
[REQ-041] The software shall be designed to meet the social requirements of the system.
[REQ-042] The software shall be designed to meet the environmental requirements of the system.
[REQ-043] The software shall be designed to meet the economic requirements of the system.
[REQ-044] The software shall be designed to meet the cultural requirements of the system.
[REQ-045] The software shall be designed to meet the political requirements of the system.
[REQ-046] The software shall be designed to meet the legal requirements of the system.
[REQ-047] The software shall be designed to meet the moral requirements of the system.
[REQ-048] The software shall be designed to meet the philosophical requirements of the system.
[REQ-049] The software shall be designed to meet the spiritual requirements of the system.
[REQ-050] The software shall be designed to meet the religious requirements of the system.
```

```

[REQ-051] The software shall be designed to meet the performance requirements of the system.
[REQ-052] The software shall be designed to meet the safety requirements of the system.
[REQ-053] The software shall be designed to meet the security requirements of the system.
[REQ-054] The software shall be designed to meet the maintainability requirements of the system.
[REQ-055] The software shall be designed to meet the interoperability requirements of the system.
[REQ-056] The software shall be designed to meet the compatibility requirements of the system.
[REQ-057] The software shall be designed to meet the portability requirements of the system.
[REQ-058] The software shall be designed to meet the scalability requirements of the system.
[REQ-059] The software shall be designed to meet the flexibility requirements of the system.
[REQ-060] The software shall be designed to meet the extensibility requirements of the system.
[REQ-061] The software shall be designed to meet the modifiability requirements of the system.
[REQ-062] The software shall be designed to meet the testability requirements of the system.
[REQ-063] The software shall be designed to meet the debuggability requirements of the system.
[REQ-064] The software shall be designed to meet the recoverability requirements of the system.
[REQ-065] The software shall be designed to meet the fault tolerance requirements of the system.
[REQ-066] The software shall be designed to meet the error handling requirements of the system.
[REQ-067] The software shall be designed to meet the logging requirements of the system.
[REQ-068] The software shall be designed to meet the reporting requirements of the system.
[REQ-069] The software shall be designed to meet the documentation requirements of the system.
[REQ-070] The software shall be designed to meet the configuration management requirements of the system.
[REQ-071] The software shall be designed to meet the version control requirements of the system.
[REQ-072] The software shall be designed to meet the backup requirements of the system.
[REQ-073] The software shall be designed to meet the recovery requirements of the system.
[REQ-074] The software shall be designed to meet the disaster recovery requirements of the system.
[REQ-075] The software shall be designed to meet the business continuity requirements of the system.
[REQ-076] The software shall be designed to meet the risk management requirements of the system.
[REQ-077] The software shall be designed to meet the compliance requirements of the system.
[REQ-078] The software shall be designed to meet the regulatory requirements of the system.
[REQ-079] The software shall be designed to meet the industry standards requirements of the system.
[REQ-080] The software shall be designed to meet the best practices requirements of the system.
[REQ-081] The software shall be designed to meet the state-of-the-art requirements of the system.
[REQ-082] The software shall be designed to meet the cutting-edge requirements of the system.
[REQ-083] The software shall be designed to meet the leading-edge requirements of the system.
[REQ-084] The software shall be designed to meet the next-generation requirements of the system.
[REQ-085] The software shall be designed to meet the future-proof requirements of the system.
[REQ-086] The software shall be designed to meet the long-term requirements of the system.
[REQ-087] The software shall be designed to meet the sustainable requirements of the system.
[REQ-088] The software shall be designed to meet the green requirements of the system.
[REQ-089] The software shall be designed to meet the ethical requirements of the system.
[REQ-090] The software shall be designed to meet the social requirements of the system.
[REQ-091] The software shall be designed to meet the environmental requirements of the system.
[REQ-092] The software shall be designed to meet the economic requirements of the system.
[REQ-093] The software shall be designed to meet the cultural requirements of the system.
[REQ-094] The software shall be designed to meet the political requirements of the system.
[REQ-095] The software shall be designed to meet the legal requirements of the system.
[REQ-096] The software shall be designed to meet the moral requirements of the system.
[REQ-097] The software shall be designed to meet the philosophical requirements of the system.
[REQ-098] The software shall be designed to meet the spiritual requirements of the system.
[REQ-099] The software shall be designed to meet the religious requirements of the system.
[REQ-100] The software shall be designed to meet the requirements of the system.
```



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

[CL1] The <[ML system](#)> is <robust>



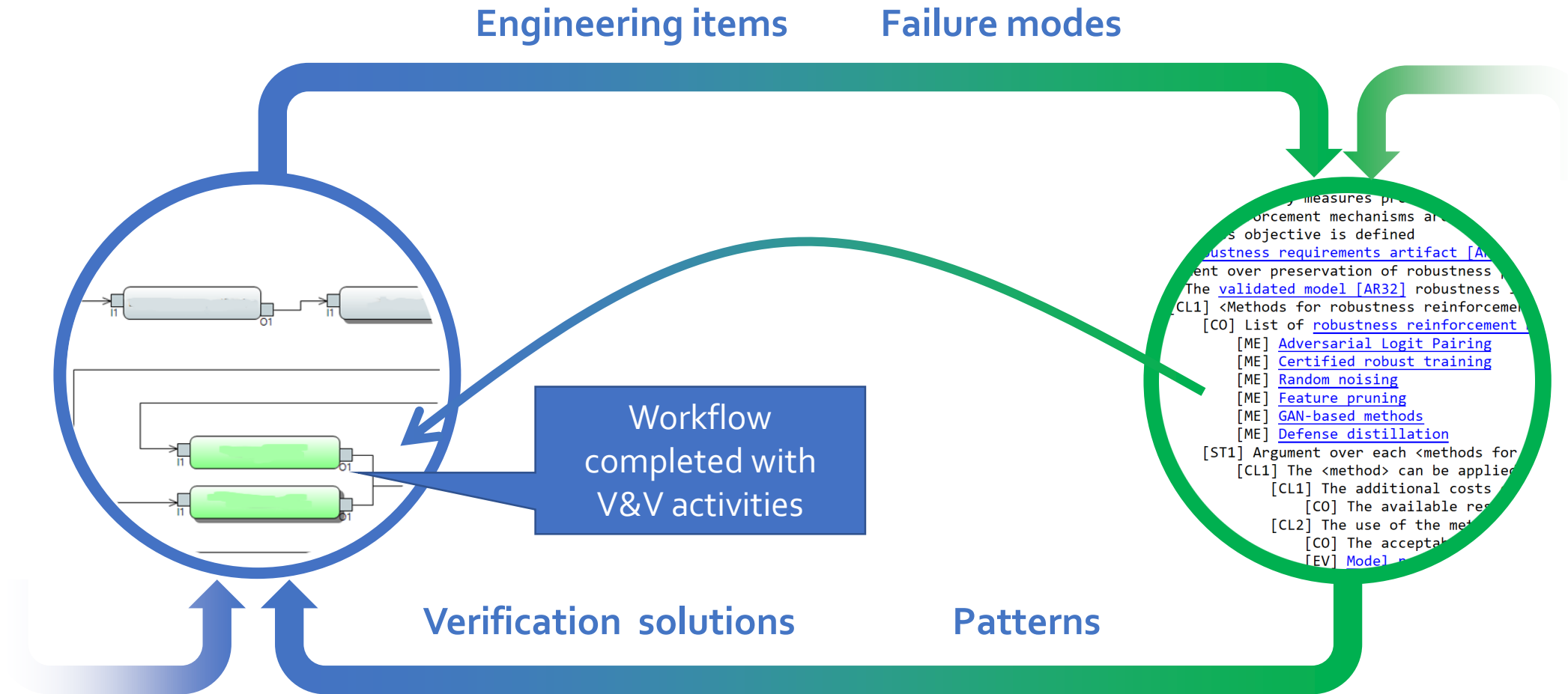
[CL1] The <[ML system](#)> maintains its <level of performance> under a set of <faults>
 [CL1] The set of <faults> is defined
 [EV] [Hazard analysis report \[AR123\]](#)
 [ST1] Partitioning between faults affecting the inputs of the <[ML system](#)> from other faults
 [CL1] The <[ML system](#)> is robust 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 <

[CL2] The <[ML system](#)> is <robust> in the presence of intentional faults by
 [CL1] <Intentional faults> which are sources of non-robustness are identified
 [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 people
 [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 training
 [CL1] The [validated model \[AR32\]](#) robustness is reinforced with
 [CL1] <Methods for robustness reinforcement> are used during
 [CO] List of [robustness reinforcement methods \[EC4 trust\]](#)

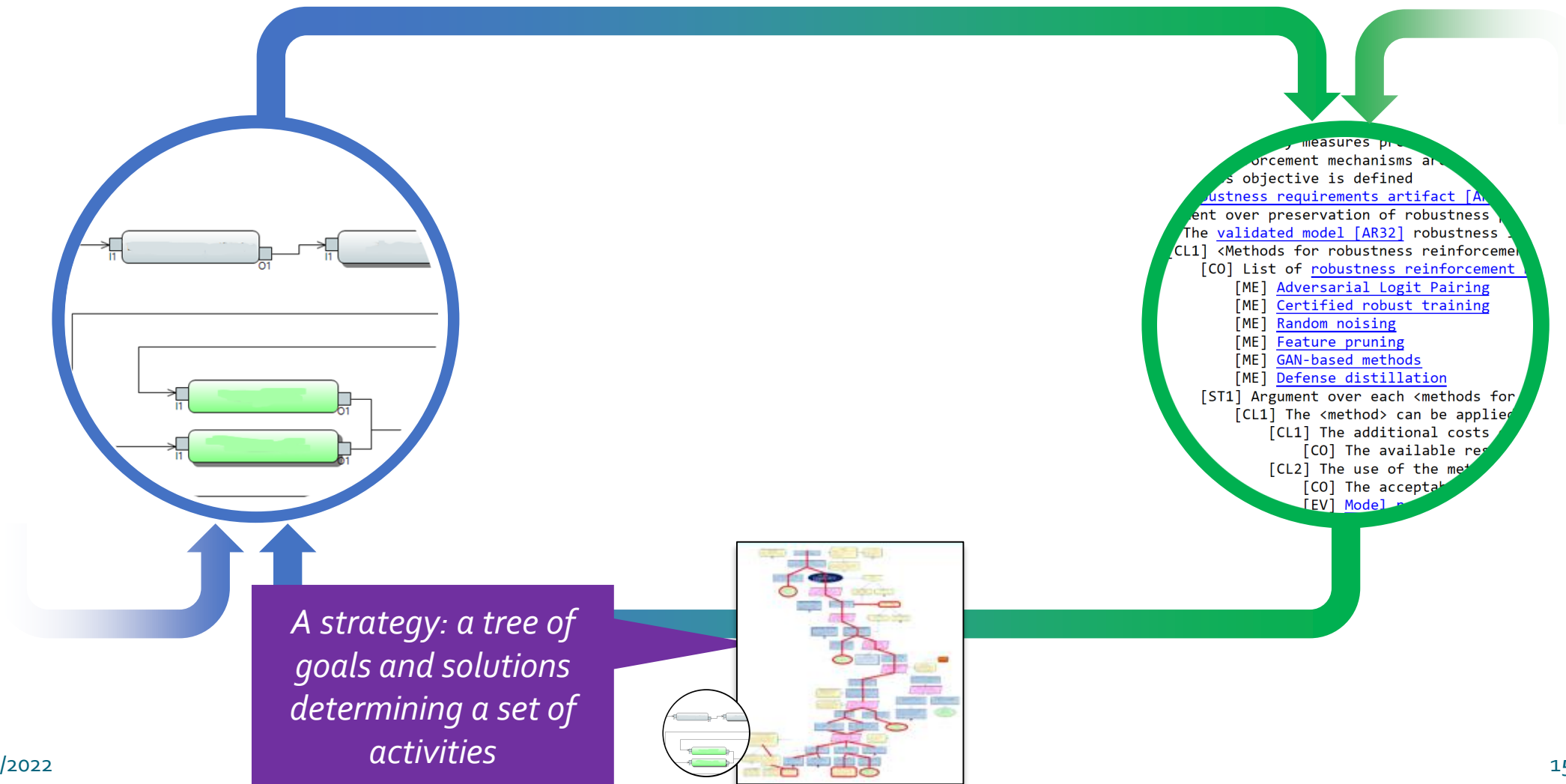
Solutions

- [ME] [Adversarial Logit Pairing](#)
- [ME] [Certified robust training](#)
- [ME] [Random noising](#)
- [ME] [Feature pruning](#)
- [ME] [GAN-based methods](#)
- [ME] [Defense distillation](#)

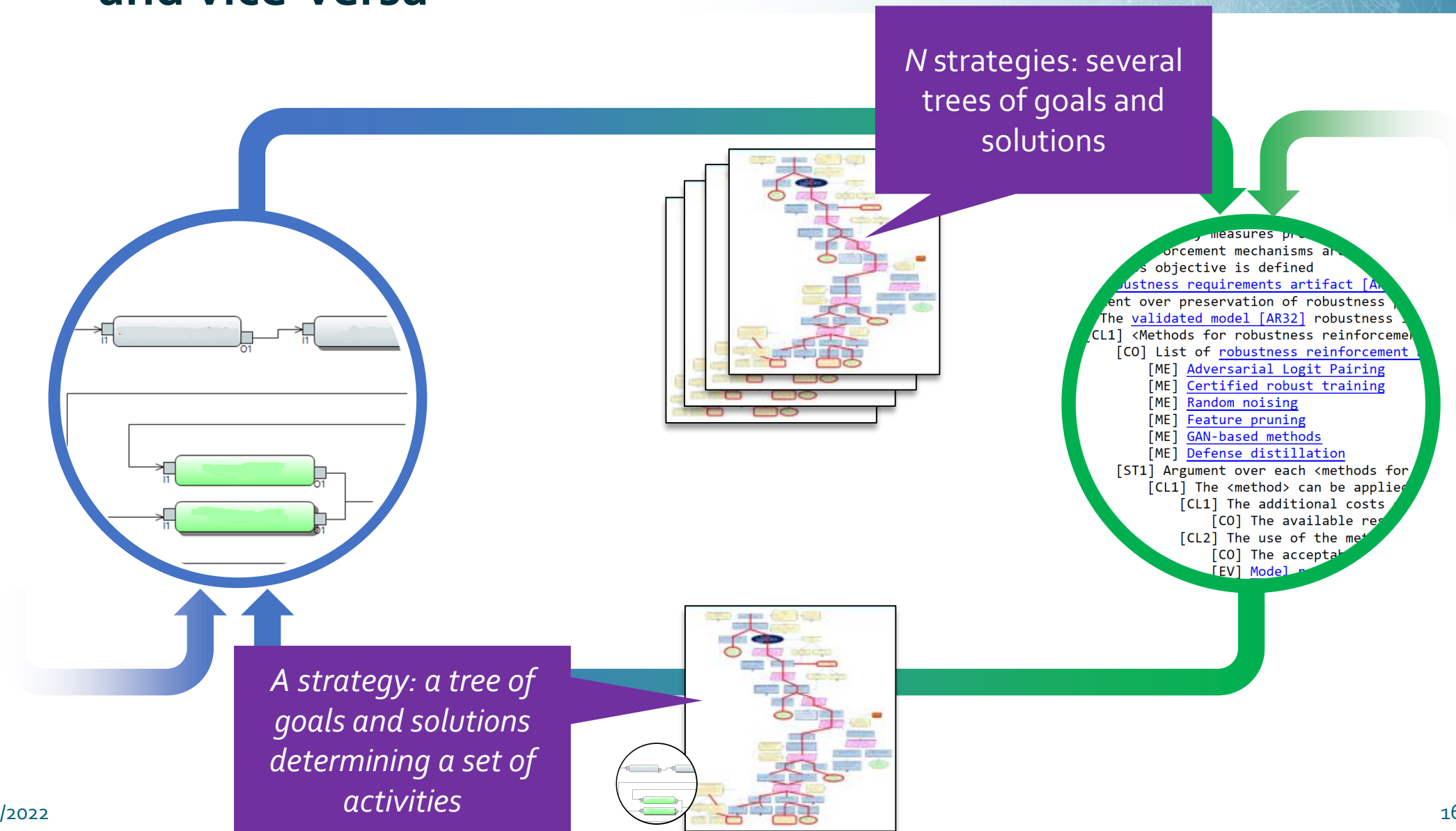
From the engineering process to the assurance case and vice-versa



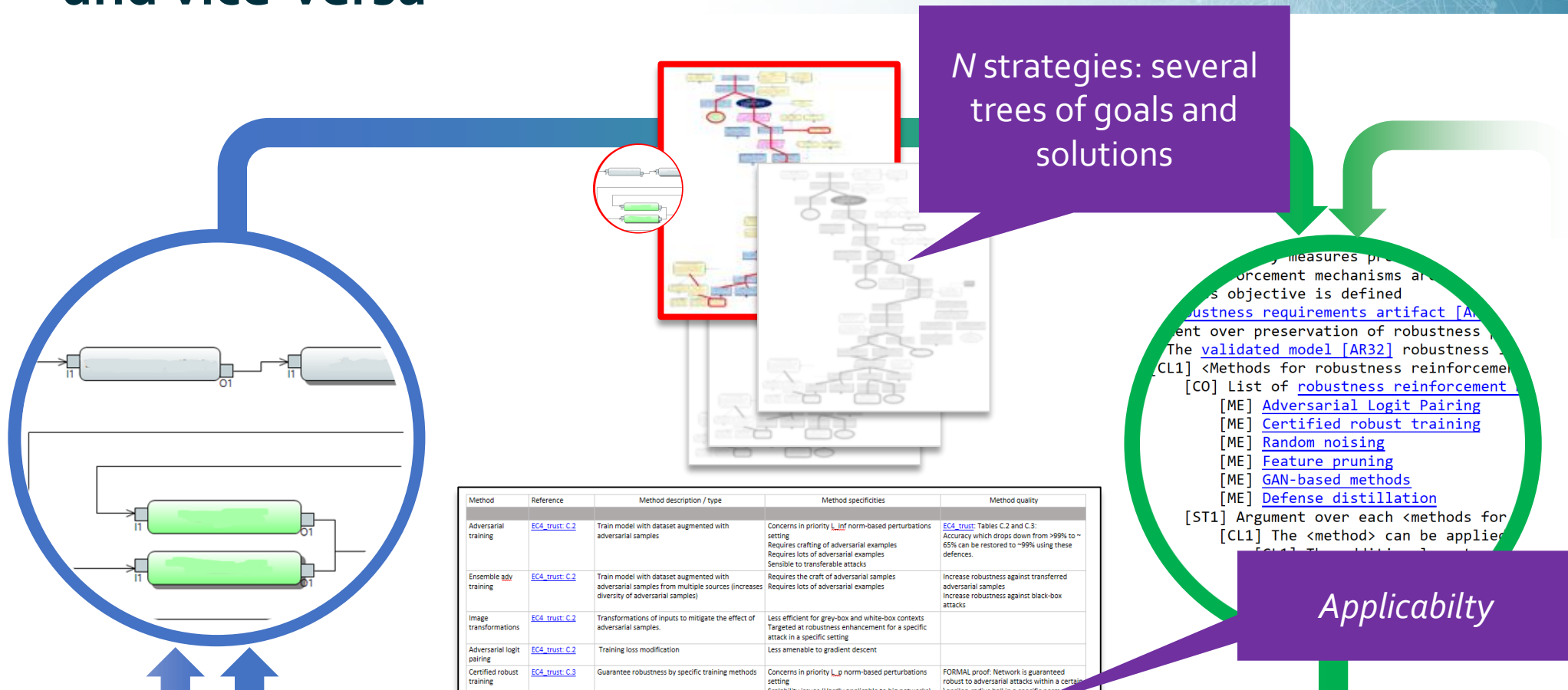
From the engineering process to the assurance case and vice-versa



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



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



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
----------	-------------------------------------	---	---	-----------------

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_{∞} 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 defenses.
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_{∞} norm-based perturbations setting Formal proof of robustness against adversarial attacks within a certain domain	FORMAL proof. Network is guaranteed robust to adversarial attacks within a certain domain

GAN-based methods	EC4_trust: C.4.3	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	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 (l_{∞}) 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.6.3	Adversarial sample detection method, based on color alteration and spatial smoothing.	Specific methods (EAD and l2-norm C&W) can bypass this detection	
Adversarial	EC4_trust: C.1.6	Filter inputs considered adversarial, leverage		

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

The screenshot shows the Confiance.ai Workbench interface, which is divided into several main sections:

- Left Panel:** Contains a tree view of assurance cases and engineering items. Callouts include:
 - "Display the assurance case associated with the property" (pointing to the AC1-AC3 list).
 - "Display the initial workflow activities and engineering items" (pointing to the AC211-AC213 and AR2102 items).
 - "Display properties applicable to an item" (pointing to the Properties table at the bottom).
- Center Panel:** Titled "Assurance cases", it features a filter dropdown set to "Constraint" and a list of attack settings (Visibility context, Distance setting, etc.). Below this is a "Strategie" section with three tabs (Strategie 1, 2, 3). A central diagram visualizes the model, with a callout: "Display applicable constraints". To the right of the diagram is a "Characteristics" table:

Metric	Value
Cost	
Confidence	0.75
...	

 Callouts include: "Provide indicators to support the selection of an argumentation strategy" (pointing to the attack settings) and "Display the workflow enriched with activities related to the argumentation strategy" (pointing to the diagram).
- Right Panel:** Mirrors the left panel's tree view but includes a "Produce V&V plan" button. Callouts include: "Display the workflow enriched with activities related to the argumentation strategy" (pointing to the AC211-AC213 items) and "Display the argumentation strategy" (pointing to the AV21 item).
- Bottom Panel:** A table with tabs for "Properties", "Reference documents", and "Definitions". The "Properties" tab is active, showing:

Property	Item	Description	Higher-level property	Reference documents
Dataset robustness	AR2102 - Augmented dataset	Characteristic of a dataset resulting from data augmentation using adversarial samples	System robustness	
System robustness	AR5X - ML-based System	Ability of a system to maintain its performance under a ... (safety ?)		EC2_Taxonomie

 Callouts include: "Display the argumentation strategy" (pointing to the AV21 item) and "Navigate through the model (parent properties, activities, engineering items, etc)" (pointing to the table).

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

