

---

# DEPENDABLE AI-ASSISTED ENGINEERING: A FORMAL FRAMEWORK FOR AI PARTICIPATION AND ASSURANCE IN SAFETY-CRITICAL WORKFLOWS

---

A PREPRINT

**Puxue Tan**

School of Mechanical and Aerospace Engineering  
Queen's University Belfast  
50 Malone Road, Belfast, UK  
p.tan@qub.ac.uk

## ABSTRACT

Generative AI can produce engineering artefacts, but generation alone does not determine whether or how those artefacts should enter safety-critical workflows. This paper develops a formal framework for assigning AI participation and assurance at the level of individual workflow units. Each unit has a participation and assurance record covering its engineering requirement, an approved operational formalization where applicable, the applicable mechanism, fallback where applicable, evidence obligations and the applicable guarantee, plus a deployment-readiness status. The framework distinguishes deterministic verification, statistically calibrated admission, authorized human judgement supported by AI advice, authorized human adjudication of AI-produced artefacts, retained deterministic tool paths and explicit non-participation; these arrangements carry different kinds of guarantee rather than levels on a common scale. The framework also separates formalization fidelity from verifier soundness, provides a staged classification and readiness procedure, and derives conditions for comparing a gated AI-assisted unit with an incumbent process under recurring-population assumptions. We instantiate and apply the framework in an executed 17-unit wing-spar structural-analysis workflow combining deterministically gated AI-generated CAD, retained deterministic computation and human judgement. The AI-generated CAD program passed all 23 deterministic checks and was admitted at the first attempt. Favourable stress magnitudes did not suffice to pass the stress criteria where the predeclared mesh-convergence evidence was insufficient; those criteria were instead referred to engineering judgement. The case demonstrates selective AI participation and explicit evidence handling at unit level; no claim is made of workflow-level dependability, certification, structural safety or productivity.

**Keywords** AI-assisted engineering · generative AI · engineering assurance · safety-critical workflows · formalization fidelity · human-in-the-loop engineering

## 1 Introduction

### 1.1 AI participation in safety-critical engineering workflows

Generative AI systems increasingly produce engineering artefacts, from analysis scripts and parametric CAD programs to technical documentation. In safety-critical engineering, however, assurance rests on traceable and reviewable evidence relating decisions and deliverables to their requirements. Introducing AI changes who or what produces a deliverable; it does not remove the need for that evidence.

The generation process does not, by itself, establish that an output is acceptable for engineering use. Whether the generator is deterministic or stochastic, capable or error-prone, acceptance depends on what generation does not supply: the requirement the artefact must satisfy, the way that requirement is made operational for checking, the rule

by which the artefact is admitted, and the course taken when the evidence is insufficient. Different parts of the same workflow may therefore call for different forms of AI participation and assurance, and the same AI system may suit one part of a workflow but not another.

Placing human review around every AI contribution is a conservative response, but not a complete organizing principle. Authorized human judgement remains necessary where it is the appropriate acceptance mechanism. Elsewhere, a repeatable deterministic check, a calibrated statistical gate or an established deterministic tool may be more appropriate, and some parts of a workflow should involve no AI at all. Uniform review also does not, by itself, record what was checked, against which specification and by whom. The difficulty is not review effort as such, but the absence of an explicit rule for how AI participates in each unit of engineering work and which assurance mechanism governs it.

The motivating concern is dependability, the ability to deliver service that can justifiably be trusted (Avizienis et al., 2004). In engineering workflows, that trust rests on evidence attached to individual contributions. The relevant question is therefore not whether AI is globally trustworthy, but, for each unit of work, how AI participates, what assurance that participation requires, and what happens when the required evidence is insufficient. Where an AI-produced artefact is proposed for release, insufficient evidence should lead to rejection, a declared fallback or authorized adjudication rather than implicit acceptance.

## 1.2 Beyond binary trust

Framing AI adoption as a binary trust decision (whether an AI output can be trusted or not) obscures the fact that different parts of a workflow admit different forms of AI participation, evidence and acceptance. Existing work addresses complementary parts of this problem, from aviation guidance on human–AI authority and learning assurance (EASA, 2024) to assurance cases, formal verification, statistical risk control, human oversight and tool qualification (Section 2).

The closest conceptual prior work is the architectural compensation discussed by Bloomfield and Rushby (2025), in which strongly assured guards or diverse secondaries compensate for components that can be only weakly assured. The present work shares that concern with controlling weakly assured components. It formalizes how AI participation is assigned within a workflow and what assurance supports it, making the applicable mechanism, fallback, evidence obligations and readiness explicit.

In the sources reviewed, we did not identify a scheme that combines four elements for each workflow unit in an engineering workflow:

- a decision on whether AI participates, and under what acceptance mechanism an AI-produced artefact may be admitted;
- a declared course of action when it is not admitted: fallback, rejection or authorized adjudication;
- an explicit assurance state, consisting of evidence obligations with explicit evidence kinds and a readiness status kept distinct from the unit’s classification;
- explicit treatment of the step from requirement to operational formalization.

The gap concerns their integration at the level of individual workflow units, rather than any one element in isolation.

Engineering contrasts show why these elements must be decided unit by unit. An AI-generated CAD program may admit deterministic checks against an approved operational formalization of its geometry, provided the checker’s soundness is supported by evidence. Such checking does not, by itself, establish fidelity to the full informal requirement. A stress calculation may already have an established deterministic tool path, and there the question is whether AI should be inserted at all. Concept selection or the acceptance of an unresolved finding may depend on authorized engineering judgement. Where a recurring population, a fixed generator configuration and trusted reference labels exist, calibrated statistical admission becomes a further option. These are different participation and assurance arrangements, not points on a single assurance scale; what unifies them is an explicit rule for how AI participates and what assurance that participation requires. Section 1.3 states the research questions this raises and the paper’s contributions.

## 1.3 Contributions and paper organization

This paper addresses three linked questions:

- how AI participation should be allocated across the units of work in an engineering workflow;
- under what acceptance mechanism an AI-produced artefact may be admitted as a unit’s deliverable;

- how evidence that does not support acceptance should be handled.

It makes three contributions.

**1. A formal framework for AI participation and assurance.** For each relevant unit of work, the framework records the engineering requirement; an approved operational formalization of a declared scope and version, where applicable; the assigned participation and assurance regime; the applicable acceptance or execution mechanism; any declared fallback; the assurance obligations and their evidence; deployment readiness; and the regime guarantee.

The regimes, covering deterministic and statistically calibrated admission, human judgement supported by AI advice, human adjudication of AI-produced artefacts, retained deterministic tool paths and explicit non-participation, differ in participation arrangement, assurance mechanism and guarantee kind, with no scalar dependability score or total ordering. The framework also treats the translation from requirement to operational formalization as an explicit assurance problem: approving a formalization is a recorded act, not a proof of its fidelity.

**2. A staged classification and readiness procedure.** As part of the formal framework, the procedure separates participation (whether AI should take part in a unit of work at all), assurance (which assurance mechanism is actually available) and readiness (whether the applicable obligations are currently evidenced). It records the current regime, any target regime attainable in principle and its deployment readiness as distinct statuses. Deterministic and statistical admission are assigned only where the verification or calibration mechanism actually exists. Its decision predicates, such as whether AI adds justified value, whether delegation is permitted and whether an approved formalization exists, remain engineering and organizational judgements, recorded with their justification.

**3. An executed engineering demonstration.** The framework was instantiated and applied to a structural-analysis workflow for a tapered, stiffened wing-spar segment, executed with regimes, criteria and tools fixed before the run. An AI-generated CAD program entered the analysis only through deterministic admission, established deterministic tools were retained, AI-produced drafts were released only through authorized human acceptance, and engineering judgement, supported by AI advice, was applied explicitly where the evidence did not resolve a criterion. Favourable numerical values did not yield a pass where the required convergence evidence was absent, even though every unit's evidence obligations had been satisfied before the run. The study is author-led and bounded in scope; its limitations are set out in Section 6.2.

Sections 2–7 present related work, the framework, the classification procedure, the case study, the discussion and the conclusions. The Supplement (Sections S1–S7) provides the case-study evidence and provenance, together with a retrospective audit of the original case-study record, made before the controlled execution.

## 2 Background and Related Work

### 2.1 AI-generated engineering artefacts

Generative models now produce engineering artefacts directly. DeepCAD (Wu et al., 2021) generates CAD operation sequences with a Transformer and is evaluated by reconstruction accuracy and the ratio of invalid outputs; CAD-MLLM (Xu et al., 2024) conditions parametric CAD generation on text, images or point clouds and adds topology-quality metrics. Kumar et al. (2025) use a large language model to write FreeCAD scripts, refined from execution errors and evaluated mainly by whether they run, and Zhang et al. (2026) survey large language models for computer-aided design and the metrics in current use.

These studies demonstrate generation capability and measure it with geometric, topological and executability metrics. They do not, by themselves, establish the rule under which a generated artefact should be admitted into an engineering workflow, nor what should happen when it is not admitted.

### 2.2 Assurance, governance and autonomy-level approaches

Aviation guidance addresses AI primarily at the level of the application and its machine-learning constituents. EASA's AI Concept Paper Issue 02 (EASA, 2024) classifies applications as Level 1 ("assistance to human") or Level 2 ("human-AI teaming"), with sublevels distinguished by the function allocated to the AI-based system and the authority retained by the end user, and structures learning assurance as a W-shaped process. The Overarching Properties (Intent, Correctness and Innocuity) are properties of a product for which approval is sought. They emerged from an FAA-sponsored initiative, and the report presenting them is not itself official guidance (Holloway, 2019). The joint EUROCAE WG-114/SAE G-34 standard ED-324/ARP6983, still a draft, defines a machine-learning development lifecycle for ML constituents (EUROCAE WG-114 and SAE G-34). Luettig et al. (2024) classify aerospace AI applications by criticality and AI-method complexity.

At the organizational level, the NIST AI Risk Management Framework (NIST, 2023) organizes AI risk management into four functions, and ISO/IEC 42001:2023 (ISO/IEC, 2023) specifies requirements for an AI management system.

Assurance-case approaches structure arguments and evidence. AMLAS (Hawkins et al., 2021) provides a stage-based safety-case process for machine-learning components. Assurance 2.0 (Bloomfield and Rushby, 2021) strengthens claims–arguments–evidence cases. Lee et al. (2026) propose reusable safety-case templates together with taxonomies of claims, arguments and seven evidence families. Surveys organize the wider ML-safety and ML-certification literature (Mohseni et al., 2022; Tambon et al., 2022). The approaches surveyed here answer different questions at different levels, and no correspondence between them and the participation and assurance regimes of Section 3 is claimed.

**Closest prior work.** Bloomfield and Rushby (2025) observe that where only weak assurance can be provided for a component, such as a machine-learning component, the system architecture should compensate, for example with a strongly assured guard performing runtime checking or a diverse secondary. This is closely related to the present concern with how AI outputs enter an engineering process; the present framework assigns participation and assurance unit by unit within an engineering workflow (Section 1.2).

### 2.3 Deterministic verification and formalization fidelity

Formalized properties can in many settings be checked deterministically. SMT solvers such as Z3 (de Moura and Bjørner, 2008) support automated checking over formalized constraints in supported logical theories, and SAT-solver unsatisfiability verdicts, for example, can be accompanied by proof certificates checked by independent, in some cases formally verified, checkers (Cruz-Filipe et al., 2017). Two distinctions matter here.

**Checking is distinct from generating.** A deterministic checker can decide whether an artefact satisfies a formalized property irrespective of how it was produced. The strength of that decision rests on the checker’s soundness, which is supported by proof where available and by engineering evidence otherwise.

**Soundness relative to a formalization is distinct from fidelity to the requirement.** Zave and Jackson (1997) argue that a specification satisfies the requirements only in conjunction with domain knowledge, so meeting a specification does not by itself establish the requirement. The same holds for the chain from requirement to approved operational formalization to verifier: a verifier may be sound with respect to its formalization while the formalization fails to capture part of the informal requirement.

Tools and practices support fidelity without proving it: NASA’s FRET translates structured natural-language requirements into temporal logic (Giannakopoulou et al., 2020), and requirements traceability has long been recognized as a persistent problem (Gotel and Finkelstein, 1994). Formalization fidelity therefore remains an assurance and governance problem in its own right.

### 2.4 Statistical acceptance and selective risk

Statistical admission builds on selective prediction, in which a predictor may abstain; Chow (1970) characterized the trade-off between error and rejection. In selective classification, coverage is the fraction of inputs on which a classifier predicts, and selective risk is its error among the inputs it accepts (El-Yaniv and Wiener, 2010). Geifman and El-Yaniv (2017) choose an acceptance threshold so that selective risk does not exceed a target with probability at least  $1 - \delta$  over i.i.d. samples, while making coverage as large as possible.

Risk-controlling calibration generalizes this idea. Bates et al. (2021) bound the expected loss of a black-box predictor’s set-valued output at level  $\alpha$  with probability at least  $1 - \delta$ . Learn then Test (Angelopoulos et al., 2025) controls user-chosen risks, including selective error among non-abstained predictions, at stated confidence.

Such guarantees depend on stated sampling assumptions and on the predictor or generator configuration to which the calibration applies, and they carry a sample-size cost: for a fixed acceptance rule with no observed violations among independent trials, exact binomial limits give the familiar zero-numerator bound of roughly  $3/n$  at one-sided 95 % confidence for  $n$  trials (Clopper and Pearson, 1934; Hanley and Lippman-Hand, 1983).

In the sources reviewed, coverage is maximized or reported rather than guaranteed jointly with selective risk at stated confidence. The joint requirement used in Section 3, bounded selective risk together with a minimum coverage at stated confidence, is this paper’s own construction on these foundations.

Agreement among models does not itself provide calibrated risk control. Kim et al. (2025) find substantially correlated errors across more than 350 language models, with larger, more accurate models showing highly correlated errors even across providers and architectures. Goel et al. (2025) observe that model mistakes become more similar as capability

increases. Taken together, these findings motivate treating agreement among models as a signal whose relation to correctness must be calibrated against the relevant formalization, rather than as a correctness guarantee.

## 2.5 Human oversight, established tools and model adequacy

Where human oversight is used as an acceptance mechanism, its effectiveness depends on how it is exercised. Parasuraman and Riley (1997) distinguish use, misuse (over-reliance), disuse and abuse of automation. Parasuraman and Manzey (2010) review evidence that automation complacency and automation bias affect naive and expert users alike, and that automation bias produces errors of omission and commission that training alone does not prevent. Bansal et al. (2021) report that AI explanations increased acceptance of AI recommendations regardless of their correctness. Human authority over a decision matters, but recorded human acceptance does not by itself establish that the accepted artefact is correct.

Established deterministic tools raise a different boundary. A tool path with fixed versions, inputs and configuration makes outputs repeatable, but repeatability is not validation, and validation evidence is not, by itself, formal qualification. In airborne software, tool qualification is a defined process (RTCA DO-330/EUROCAE ED-215 (RTCA, 2011), recognized by FAA Advisory Circular 20-115D (FAA, 2017)) whose required qualification level depends on the tool’s potential impact on the software life cycle and on the software level of the item it supports. Whether a computational model is adequate for its intended use is a separate question, addressed by verification and validation practice such as ASME V&V 10-2019 (ASME, 2019), a question that retaining an established tool does not settle.

## 2.6 Synthesis and research gap

The literature offers mature, complementary mechanisms for artefact generation, assurance of machine-learning constituents, organizational risk management, assurance cases, deterministic and proof-checked verification, statistical risk control, the human factors of oversight, and tool qualification and model verification and validation. These address different questions about different objects: the application, the organization, the assurance argument, the artefact or the tool. Read together, they also mark the limits of each mechanism: deterministic checking is sound only relative to a formalization whose fidelity it does not establish (Section 2.3); statistical admission requires stated sampling assumptions and a fixed configuration (Section 2.4); recorded human acceptance confers authority but not correctness; and a retained tool secures repeatability, not model adequacy (Section 2.5). In the sources reviewed, however, we did not identify a scheme that integrates these mechanisms, and their limits, for each workflow unit in an engineering workflow by combining the four elements set out in Section 1.2.

Dependability motivates this integration but is not formalized here as a scalar property. Section 3 formalizes the integration as a framework for AI participation and assurance.

# 3 Formal Framework

The framework’s unifying principle is **an explicit AI-participation decision for each unit of engineering work, paired with the assurance required to justify that decision**. Where AI produces an artefact for possible release (Types A, B and C[adjudicate]), an acceptance or admission mechanism is part of that assurance; under C[assist], AI is advisory and an authorized person produces the deliverable; Type D retains an established deterministic path; and Type E records justified non-participation. The regimes defined below differ in participation arrangement, assurance mechanism and guarantee kind, and share no common scalar measure; dependability (Section 1.1) is the motivating system-level property.

## 3.1 Activities, Workflow Units, Requirements and Formalizations

An engineering **workflow**  $W$  is a finite collection of **activities**: recognizable units of engineering work such as geometry definition, mesh generation or compliance checking. Activities are the units in which engineers describe their work, but not necessarily the units at which assurance can be meaningfully assigned: a single activity may contain parts whose acceptance can be decided by computation alongside parts that require professional judgement. The framework therefore classifies **workflow units**.

**Definition 1 (Workflow unit).** A *workflow unit*  $\kappa$  of an activity is a unit of work with:

- an input space  $I_\kappa$ ;
- an output (artefact) space  $O_\kappa$ ;
- a declared **valid-input domain**  $I_\kappa^{valid} \subseteq I_\kappa$ ;

such that its AI-participation and assurance characteristics are sufficiently homogeneous for a single participation and assurance regime to be meaningful. An activity is decomposed into one or more workflow units (Section 4.3).

The valid-input domain records the conditions that inputs are assumed to satisfy: units, provenance, consistency with upstream deliverables, and so on. **Every guarantee in this framework is conditional on  $i \in I_\kappa^{valid}$ .** Whether upstream units actually deliver such inputs is outside the scope of a single assurance record (Definition 6; Section 4.4).

**Definition 2 (Requirement and approved formalization).**

- The **requirement**  $R_\kappa$  of a workflow unit is the condition its deliverable must satisfy, as stated by the relevant regulation, organization, engineering practice or design intent.  $R_\kappa$  is generally expressed in natural language. It need not be computable. It need not even be bivalent: for some units, competent experts may legitimately disagree on whether a given output meets it.
- An **approved formalization** of  $R_\kappa$ , where one exists, is a predicate

$$\hat{S}_\kappa : I_\kappa \times O_\kappa \rightarrow \{0, 1\},$$

together with a declared **scope** (the part of  $R_\kappa$  it is intended to capture) and a **version identifier**, adopted through a recorded approval.

In general  $R_\kappa \neq \hat{S}_\kappa$ : the formalization is an operational interpretation of a declared scope of the requirement, adopted for the purpose of checking, not the requirement itself. Whether, for example, inequalities over computed stresses faithfully represent the governing regulation is addressed in Section 3.8.

**Definition 3 (Generator).** A *generator*  $G_\kappa$  is an AI system that, given  $i \in I_\kappa$ , produces an artefact  $o \sim G_\kappa(i)$ . The generator is treated as a black box and may be stochastic. When a guarantee depends on the generator's behaviour (Type B), the generator is identified by a **version identifier**, and the guarantee refers to that version only.

### 3.2 Participation and Assurance Records

#### Gate, fallback, and released output

**Definition 4 (Gate, admission event, fallback).**

- A *gate* for workflow unit  $\kappa$  is a rule that decides, for an input  $i$  and an AI-produced artefact  $o$ , whether  $o$  is **admitted** as the unit's deliverable. We write  $E$  for the admission event.
- The *fallback*  $F_\kappa$  is the declared path to which execution is routed when the artefact is not admitted. Examples are regeneration, manual execution, or escalation to human adjudication.
- The *released output* is

$$O_{rel}(i) = \begin{cases} o & \text{on } E, \\ F_\kappa(i) & \text{on } \neg E, \end{cases}$$

where  $F_\kappa(i)$  may be empty ( $\perp$ ) if the fallback does not produce a deliverable.

Rejection routes execution to the fallback; it does not by itself ensure that a deliverable is produced, or that the fallback's deliverable is correct. The availability and quality of the fallback are properties of the fallback path, to be evidenced separately where they matter.

#### Evidence obligations

A regime's guarantee holds only under conditions the framework cannot establish on its own, such as verifier soundness, representative calibration data, human authority or use of a tool within its validity domain. These are recorded as **assurance obligations** with explicit evidence kinds.

**Definition 5 (Assurance obligation).** An *assurance obligation* is a condition on which a regime's guarantee depends, recorded with its supporting evidence and one of four evidence kinds:

- **[Proof]**: a mathematical proof obligation, *discharged* by a proof (e.g., a checked proof certificate or a verified checker);
- **[Emp]**: an empirically validated condition, supported by data and carrying statistical uncertainty;
- **[Org]**: an organizationally authorized condition, established by a recorded decision of a person or body with the relevant authority;

- **[Eng]**: an engineering assumption supported by evidence (review, independent implementation, testing, validation history) but not mathematically proven.

Only [Proof] obligations are *discharged*; the others are *satisfied with stated evidence* (testing a verifier on examples supports but does not prove its soundness). Whether a record's obligations are satisfied determines its *deployment readiness* (Definition 6), kept separate from which regime a unit requires (Section 4.2).

### The record

**Definition 6 (Participation and assurance record).** A *participation and assurance record* (for short, *assurance record*) for workflow unit  $\kappa$ , denoted  $AR_\kappa$ , is

$$AR_\kappa = \langle R_\kappa, \hat{S}_\kappa, r_\kappa, \text{Mech}_\kappa, F_\kappa, \text{Obl}_\kappa, G_{r_\kappa} \rangle,$$

with a version identifier. Its fields are:

- $R_\kappa$ : the requirement.
- $\hat{S}_\kappa$ : the approved formalization and its scope, where one exists.
- $r_\kappa \in \{A, B, C[\text{assist}], C[\text{adjudicate}], D, E\}$ : the participation and assurance regime (Type C carries the role attribute assist or adjudicate; Section 3.5).
- $\text{Mech}_\kappa$ : the participation and assurance mechanism prescribed by the regime. It is instantiated as a deterministic verifier and admission gate (Type A); a calibrated statistical gate (Type B); authorized human acceptance or adjudication ( $C[\text{adjudicate}]$ ); authorized human production or judgement, with AI material restricted to labelled advice ( $C[\text{assist}]$ ); an established deterministic tool path (Type D); or non-AI execution (Type E).
- $F_\kappa$ : the declared fallback, where applicable.
- $\text{Obl}_\kappa$ : the assurance obligations with their evidence kinds and evidence.
- $G_{r_\kappa}$ : the regime's **guarantee clause**, defined in Sections 3.3–3.6.

**Deployment readiness.** An assurance record has a readiness status

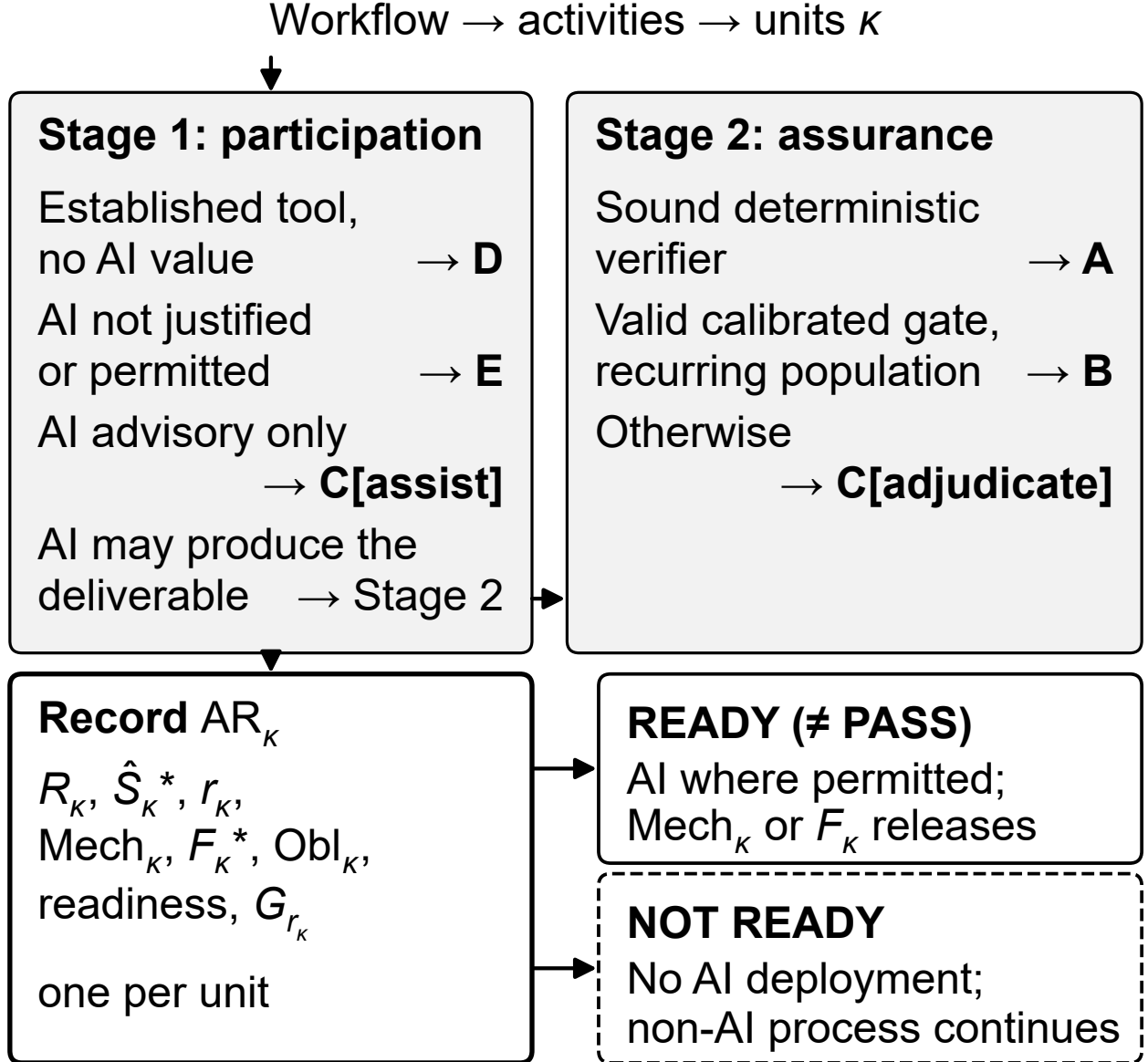
$$\text{readiness}(AR_\kappa) \in \{\text{READY}, \text{NOT READY}\}.$$

It is READY when every obligation in  $\text{Obl}_\kappa$  is satisfied with evidence of an adequate kind, and NOT READY otherwise.

- The regime  $r_\kappa$  states which participation and assurance regime the unit requires.
- Readiness states whether the obligations needed to operate that regime currently hold. It is a status, not a guarantee.
- AI participates in a unit only under a READY assurance record in a regime that permits AI participation. Where an AI-produced artefact may become the unit's deliverable, it is admitted only through the mechanism  $\text{Mech}_\kappa$  prescribed by that regime.
- A guarantee clause  $G_{r_\kappa}$  is claimed only for a READY record.

Figure 3.1 summarizes the framework: the staged assignment of regimes, the record fields of Definition 6 and their separation from deployment readiness.

The assurance record is a structured record, not an algebraic object: no refinement or composition operators are defined on it. Its **definition** (the fields above and the guarantee clause they instantiate), the **evidence** that its obligations are satisfied (which determines readiness) and the **mathematical consequences** that follow once the obligations hold are kept apart.



**Figure 3.1.** Overview of the workflow-unit participation and assurance framework. Regime classification, assurance evidence and deployment readiness are distinct; the regimes carry different guarantee kinds and are not ordered on a scalar assurance scale. Stage 1 (Section 4.2) assigns Type D, Type E or C[assist], or passes the unit to Stage 2, which selects the admission rule for units whose deliverable AI may produce. The record box shows the fields of Definition 6 (\* where applicable) together with the record’s readiness status. Under a READY record, the prescribed mechanism produces, admits or authorizes the deliverable, or a declared fallback produces it; under a NOT READY record (dashed outline), the existing non-AI process produces it (Definition 15). PASS denotes a favourable study outcome label (Section 5.6), which readiness does not imply.

### 3.3 Type A — Deterministically Gated Units

Some AI-produced artefacts, such as parametric CAD programs whose geometry can be measured, are hard to produce but deterministically checkable. Where the check’s soundness does not rest on trust in the generator, AI may participate as an untrusted generator whose artefacts are admitted only if they pass the check.

**Definition 7 (Type A unit and guarantee  $G_A$ ).** A workflow unit is **Type A** if:

- it has an approved formalization  $\hat{S}_\kappa$ ;
- a deterministic verifier  $V_\kappa : I_\kappa \times O_\kappa \rightarrow \{\text{accept}, \text{reject}\}$  is implemented and terminates within the unit’s time budget;
- the gate admits an artefact only if  $V_\kappa$  accepts it.

The obligations required for  $G_A$  are:

- **verifier soundness** with respect to the approved  $\hat{S}_\kappa$ :  $V_\kappa(i, o) = \text{accept} \Rightarrow \hat{S}_\kappa(i, o)$ . This is [Proof] where a proof is available; otherwise [Eng].
- **termination** within budget ([Emp]/[Eng]).

Further obligations of the record are **fidelity support** for  $\hat{S}_\kappa$  (Section 3.8), which bears on how  $G_A$  relates to  $R_\kappa$ , and a declared fallback.

The guarantee clause is **deterministic acceptance soundness relative to the approved formalization**: for every  $i \in I_\kappa^{\text{valid}}$  and every artefact  $o$ ,

$$E(i, o) \Rightarrow \hat{S}_\kappa(i, o), \quad (1)$$

conditional on verifier soundness.

Clause (1) is a per-instance statement with five boundaries:

- **Completeness is not required**: a sound verifier may reject correct artefacts, at a cost in efficiency, not assurance.
- **Availability is not guaranteed**: rejections route to the fallback; the generator–verifier acceptance rate should be reported.
- **Generator reliability does not enter  $G_A$** : conditional on verifier soundness, only artefacts satisfying the approved formalization  $\hat{S}_\kappa$  are admitted.
- **Fidelity is not established**: (1) concerns  $\hat{S}_\kappa$ , not  $R_\kappa$  (Section 3.8).
- **Verifier soundness is itself an obligation** on which the guarantee is conditional.

Where soundness cannot be proven, independence between verifier and generator (separate development, diverse tooling, independent review) strengthens the [Eng] evidence for it. Independence is good practice, not a condition of  $G_A$ : if soundness is proven, (1) holds regardless of who produced the verifier.

**Generation–verification asymmetry.** AI participation can be useful under Type A where an asymmetry holds: generation is difficult or costly relative to a tractable deterministic verification step. This concerns the value of AI participation, not the validity of (1); it is not part of  $G_A$ , and Definition 7 does not require it. Where the verifier merely reproduces the unit’s own computation, such as evaluating inequalities whose result is the deliverable, an AI generator adds no justified value, and the unit is more accurately treated as Type D (Section 3.6) than as Type A.

*Illustration.* In the wing-spar study (Section 5.3), an AI-generated CAD program is admitted through a deterministic geometric gate that establishes conformance only within the formalization’s declared scope; residual engineering adequacy is a separate human-adjudicated unit (Sections 3.5 and 4.3).

### 3.4 Type B — Statistically Gated Units

For some workflow units, an approved formalization exists but can be evaluated only on a sample (by expert review or an expensive reference computation), not per instance at deployment. Computable signals (tests, consistency checks, quality metrics, model agreement) can then be combined into a score whose threshold is calibrated against reference evaluations of  $\hat{S}_\kappa$  on a sample from the unit’s population. The resulting guarantee is statistical and refers to that population.

Let  $D_\kappa$  be a **declared deployment population** of unit instances, and  $G_\kappa$  a **fixed generator version**. For a gate  $g$  with admission event  $E_g$ , define its **selective acceptance error** and **coverage**:

$$\eta(g) = P_{D_\kappa, G_\kappa}(\neg \hat{S}_\kappa(i, o) \mid E_g), \quad \text{cov}(g) = P_{D_\kappa, G_\kappa}(E_g).$$

( $\eta(g)$  is defined only when  $\text{cov}(g) > 0$ ; in (2) below, the event is taken to fail when  $\text{cov}(\hat{g}) = 0$ .)

**Definition 8 (Calibration procedure).** A calibration procedure  $\mathcal{K}$  maps a calibration sample  $\{(i_j, o_j, y_j)\}_{j=1}^m$  to a gate  $\hat{g} = \mathcal{K}(\cdot)$ . Here  $i_j$  is drawn from  $D_\kappa$ ,  $o_j \sim G_\kappa(i_j)$ , and  $y_j = \hat{S}_\kappa(i_j, o_j)$  is obtained from a **trusted reference labelling procedure**.  $\mathcal{K}$  is *valid for targets*  $(\alpha, c, \delta)$  under stated sampling assumptions (for example, independent draws from  $D_\kappa$ ) if, under those assumptions,

$$P_{\text{cal}}(\eta(\hat{g}) \leq \alpha \text{ and } \text{cov}(\hat{g}) \geq c) \geq 1 - \delta, \quad (2)$$

where  $P_{\text{cal}}$  is taken over the calibration sample. Here:

- $\alpha \in (0, 1)$  is the **selective-risk target**;
- $c \in (0, 1]$  is the **coverage target**;
- $\delta \in (0, 1)$  is the **calibration confidence parameter**.

**Definition 9 (Type B unit and guarantee  $G_B$ ).** A workflow unit is **Type B** if it has an approved formalization  $\hat{S}_\kappa$  and its gate is produced by a calibration procedure valid for stated targets  $(\alpha, c, \delta)$ , with  $c > 0$ . The obligations are:

- a declared deployment population  $D_\kappa$  ([Eng]);
- a fixed generator version ([Org]);
- a trusted reference labelling procedure ([Eng]/[Org]);
- validity of the calibration procedure under stated assumptions (the procedure’s property is [Proof], typically imported from the statistical literature);
- satisfaction of those sampling assumptions by the calibration and deployment data ([Emp]/[Eng]);
- monitoring for population shift ([Emp]);
- reporting of the observed acceptance rate.

The guarantee clause is (2): **with confidence at least  $1 - \delta$  over calibration, the calibrated gate has selective acceptance error at most  $\alpha$  and coverage at least  $c$  on  $D_\kappa$  for generator version  $G_\kappa$ .**

Several features of this definition are deliberate.

- **Selective risk with coverage.** A bound on  $P(E \wedge \neg \hat{S}_\kappa)$  is satisfied trivially by a gate that admits nothing. Clause (2) instead bounds the error *among admitted artefacts*, the quantity relevant to anyone relying on one, and the coverage requirement makes the guarantee non-vacuous and keeps  $\eta$  well defined.
- **Population- and generator-specific.** The *regime* is determined by the unit and its available assurance mechanism, but a *calibrated Type B guarantee instance* refers to a specific population and generator version, does not transfer, and must be recalibrated after a material change in either.
- **Population-level, not per-instance.** Clause (2) does not establish the correctness of any individual artefact.
- **One-off units.** If no meaningful recurring population can be declared, the calibration predicate  $P_{\text{calib}}$  (Section 4.1) is false and the unit cannot be Type B; the framework does not introduce subjective probabilities to make a statistical guarantee apply to a one-off decision. Types A, C, D and E do not depend on such a population.
- **Agreement is not a guarantee.** Model agreement, tests and metrics may feed the score but are not a statistical guarantee: model errors can be correlated (Kim et al., 2025; Goel et al., 2025), and agreement says nothing quantitative about correctness unless calibrated against  $\hat{S}_\kappa$ . Correlated errors among the combined signals do not by themselves invalidate a valid calibration, but they may reduce the coverage attainable at a given risk target  $\alpha$ .

**Acceptable calibration methods.** The framework prescribes no particular procedure. Acceptable classes, provided (2) holds under assumptions valid for the unit, include selective classification with guaranteed risk (Geifman and El-Yaniv, 2017), finite-sample binomial confidence procedures applied to a gate fixed in advance (Clopper and Pearson, 1934), Learn-then-Test-type risk control (Angelopoulos et al., 2025) extended with a coverage constraint, and other statistically valid schemes.

**Remark (illustrative calibration effort).** In the simplest special case, the gate is fixed in advance (not selected using the calibration data), the admitted calibration instances are independent and each violates  $\hat{S}_\kappa$  with probability  $\eta$ , and none of the  $k$  admitted instances violates  $\hat{S}_\kappa$ . A one-sided exact binomial bound at confidence  $1 - \delta$  then gives  $\eta \leq 1 - \delta^{1/k} \approx \ln(1/\delta)/k$ , so establishing  $\eta \leq \alpha$  requires about  $k \approx \ln(1/\delta)/\alpha$  admitted instances: for  $\delta = 0.05$ ,  $k = 299$  for  $\alpha = 10^{-2}$  and  $k = 2995$  for  $\alpha = 10^{-3}$ . The total sample is larger (roughly  $k$  divided by the coverage), the coverage clause needs its own confidence bound, and a threshold selected on the same data changes the requirement. The example is not a general requirement, but **rigorous Type B assurance requires a reusable operational population and sufficient labelled calibration evidence, and more stringent risk targets generally require larger calibration sets.**

*Illustration (hypothetical).* AI-drafted summaries of standardized inspection records, recurring across many parts, are checked against a formalized list of mandatory fields and value ranges, some confirmable only by an engineer. Engineer review of a calibration sample provides reference labels; the calibrated gate admits summaries scoring above its threshold and routes the rest to review. No Type B instance is claimed for the wing-spar study.

### 3.5 Type C — Human Judgement and Adjudication

For many workflow units, the deliverable or its acceptance cannot be delegated to computation: the requirement may not be formalizable or may admit legitimate expert disagreement, or regulation and professional responsibility may require a named person to decide. AI may still participate, by supplying advisory material or by drafting the artefact; Type C covers both arrangements through one regime with a role attribute.

**Definition 10 (Type C unit and guarantee  $G_C$ ).** A workflow unit is **Type C** if its deliverable or decision is released only through a recorded act of an identified, authorized person. The role attribute is one of:

- **C[assist]:** AI supplies advisory material (analysis, options, summaries, flagged concerns), and the authorized person produces the deliverable or decision. AI material is retained only as labelled advisory material and is never itself admitted as the deliverable.
- **C[adjudicate]:** AI produces a draft artefact, and the draft is released as the deliverable only upon the authorized adjudicator’s explicit acceptance, possibly after revision.

The obligations are:

- the identity and authority of the authorized person ([Org]);
- a recorded act: acceptance of the AI-produced artefact (C[adjudicate]), or production of the deliverable or decision (C[assist]) ([Org]);
- recorded AI provenance of every contributing artefact or advisory item ([Org]);
- a recorded rationale and the evidence consulted ([Org]/[Eng]).

These obligations concern readiness, not classification: a unit requiring human judgement or adjudication is Type C even before an authorized process exists, but its record is then NOT READY, and no AI-produced artefact or advisory material is used for it.

The guarantee clause is **authorized human release**: under C[adjudicate], no AI-produced artefact of the unit is released as its deliverable without the recorded acceptance of an authorized adjudicator; under C[assist], the deliverable or decision is produced and recorded by an authorized person, with AI material present only as labelled advice. In both cases, all AI contributions are identifiable.

$G_C$  is procedural. It makes no claim about the probability that the released deliverable is correct, or how that probability compares with unaided adjudication. AI assistance may improve decisions or degrade them, for example through automation bias (Parasuraman and Manzey, 2010) or anchoring on AI recommendations (Bansal et al., 2021); which occurs is an empirical question outside the formal guarantee. Mitigations such as independent-first judgement can be recorded as [Eng] obligations, but  $G_C$  does not depend on them. Human authority does not imply human correctness; the framework guarantees the former only.

### 3.6 Types D and E — Retained Deterministic Paths and Non-Participation

Types D and E are participation decisions in which AI-produced artefacts do not contribute to the deliverable.

**Definition 11 (Type D unit and guarantee  $G_D$ ).** A workflow unit is **Type D** if two conditions hold:

- an **established deterministic tool path** already produces its deliverable, with stated validation or assurance evidence and a declared validity domain;
- AI participation would add no justified value beyond that path.

Type D is thus a decision to retain an established deterministic path, whether or not AI could assist the unit; it is not a form of verified AI.

The obligations are:

- the stated validation evidence for the tool path ([Eng]; plus [Org] where a formal qualification record exists);
- operation within the declared validity domain ([Eng]; the domain check may itself be a separate Type A or D unit).

The guarantee clause is **retention of the established path**: the deliverable is produced by the established tool path on inputs within its validity domain, and no AI-produced artefact substitutes for its output.

Four distinct things are easily conflated:

1. **Deterministic solver correctness**: the tool computes what it is specified to compute (e.g., a finite-element solver solves the discretized equations to tolerance).
2. **Validation evidence**: benchmarks, comparisons with analytical or experimental results, and operational history supporting (1).
3. **Formal tool qualification**: a recorded qualification under an applicable standard (e.g., RTCA DO-330/EUROCAE ED-215), existing only where actually carried out.
4. **Engineering model adequacy**: whether discretization, idealization, boundary conditions and analysis type suit the engineering question.

$G_D$  retains the established tool’s deterministic correctness, supported by the stated validation evidence and, where available, formal tool qualification. It does **not** establish engineering-model adequacy; mesh adequacy, idealization and boundary conditions are workflow units in their own right, with their own records.

**Definition 12 (Type E unit and guarantee  $G_E$ )**. A workflow unit is **Type E** if AI does not participate in it and the retained-tool rationale of Type D does not apply. Typically, this means the unit is performed by people. The guarantee clause is **non-delegation**: no AI-produced artefact contributes to the unit’s deliverable. The obligation is an auditable record that no AI artefact contributed ([Org]).

Type E is explicit, justified non-participation, defined narrowly: where AI supplies useful advisory material but a human produces the deliverable, the unit is C[assist], not Type E.

### 3.7 Relations Among the Participation and Assurance Regimes

The five guarantee clauses are of different kinds:

- $G_A$  is a per-instance logical implication;
- $G_B$  is a population-level statistical statement with a coverage requirement;
- $G_C$  is a procedural property of release;
- $G_D$  and  $G_E$  are participation boundaries.

They are not points on a single scale, and **no total ordering among the regimes is claimed**.

One comparison is meaningful. For units with the **same** approved formalization  $\hat{S}$ , a Type A gate with a sound verifier admits only artefacts satisfying  $\hat{S}$ . Its selective acceptance error is therefore zero on every population within  $I^{valid}$  on which it admits anything:

$$P(\neg \hat{S} \mid E) = 0.$$

Hence it satisfies the risk part of  $G_B$  for every  $\alpha$ , but not necessarily the coverage part, since a sound verifier may reject most artefacts. The comparison does not extend across different formalizations (a Type A guarantee over a narrow  $\hat{S}$  is not stronger than a Type B guarantee over a broad one), nor to Type C, whose guarantee concerns authority and traceability rather than error rates. Types D and E are boundaries of AI participation, not levels of assurance.

### 3.8 Formalization Fidelity

The guarantees of Types A and B are stated relative to an approved formalization. Their bearing on the real requirement depends on two chains that meet only at admission:

$$\begin{array}{c} R_\kappa \xrightarrow{\text{formalized as}} \hat{S}_\kappa \xrightarrow{\text{implemented by}} V_\kappa, \\ i \xrightarrow{G_\kappa} o \xrightarrow{\text{evaluated by } V_\kappa} E \text{ (admit) or } \neg E \text{ (reject)}. \end{array}$$

The **requirement**  $R_\kappa$  is interpreted as  $\hat{S}_\kappa$ ;  $\hat{S}_\kappa$  is implemented as a verifier  $V_\kappa$  (Type A), or through the reference labelling against which a gate is calibrated (Type B, where the calibrated gate takes the place of  $V_\kappa$ ). Separately, the generator produces the artefact  $o$ , which that mechanism evaluates.

Two distinct gaps separate the artefact’s admission from the requirement:

1. **Requirement  $\rightarrow$  formalization.** Is  $\hat{S}_\kappa$  a sufficiently faithful operational interpretation of the declared scope of  $R_\kappa$ ? Regulatory text may be ambiguous, implicit conventions (such as the load level at which a criterion applies) omitted, a threshold mis-transcribed, or the scope narrower than intended.
2. **Formalization  $\rightarrow$  implementation.** Does  $V_\kappa$  soundly implement  $\hat{S}_\kappa$ ? This is the verifier-soundness obligation of Type A. For Type B, it is the correctness of the reference labelling.

The second gap can sometimes be closed by proof. The status of the first depends on the form of the authoritative requirement.

- **Informal requirements.** When  $R_\kappa$  is regulatory text, design intent or engineering judgement expressed in natural language, the fidelity of  $\hat{S}_\kappa$  cannot, in general, be established by proof in the sense in which verifier soundness can. It is supported by engineering evidence and authorized interpretation. This is the common case in regulated engineering, and it is where formalization requires explicit assurance.
- **Requirements that are already formal.** When the authoritative source requirement is itself formal or machine-checkable (for example, a normative data schema or a formally specified interface constraint), the fidelity gap may collapse if  $\hat{S}_\kappa$  is that requirement, or it may admit formal refinement or equivalence evidence, which would be a [Proof] obligation.

For informal requirements, the framework distinguishes the **approval** of a formalization (a governance and engineering-review act, recorded as an [Org] obligation and supported by [Eng] evidence such as clause-by-clause traceability, test cases and independent re-derivation) from its **actual fidelity**, which may remain imperfect after approval. **Approval of  $\hat{S}_\kappa$  is not proof that  $\hat{S}_\kappa$  represents  $R_\kappa$ ,** even when the requirement is formal, unless formal equivalence or refinement evidence is actually provided.

**Definition 13 (Formalization task).** For each workflow unit for which a formalization is produced, the *formalization task*  $\tau_{\text{form}}(\kappa)$  is defined by:

- input:  $R_\kappa$  and its relevant context;
- output: a formalization  $\hat{S}_\kappa$  with declared scope (and, where applicable, a verifier  $V_\kappa$ );
- requirement: that  $\hat{S}_\kappa$  adequately represents the declared scope of  $R_\kappa$ .

The formalization task is itself classified by the procedure of Section 4. For natural-language regulatory or engineering requirements, final acceptance of fidelity rests on authorized human judgement (C[adjudicate] where an AI-drafted formalization may be adopted, otherwise C[assist] or Type E), since no computation can decide it. Deterministic sub-checks (well-formedness of an encoding, consistency of units, agreement of a threshold with a cited table, coverage of a finite clause list) may be separated and classified as Type A or D where justified. AI-assisted formalization is thus an explicit, classified task, not silently absorbed into a Type A gate.

A change to an approved formalization, including one decided in a downstream unit, is a change to the output of  $\tau_{\text{form}}(\kappa)$ . It produces a new version of  $\hat{S}_\kappa$ , and every gate that depends on it is re-evaluated.

For the comparison of Section 3.9, the residual effect of imperfect fidelity is captured by the **specification-fidelity residual**

$$\gamma = P(\hat{S}_\kappa(i, o) \wedge \neg R_\kappa(i, o) \mid E),$$

the probability that an admitted artefact satisfies the formalization but not the requirement. It is defined only where  $R_\kappa$  is bivalent, may be difficult or impossible to estimate exactly, and is used only in the optional comparison of Section 3.9. Review and approval of  $\hat{S}_\kappa$  do not establish  $\gamma = 0$ ; the case  $\gamma = 0$  is an ideal conditional assumption.

### 3.9 Gated-Fallback Comparison with an Incumbent Process

The guarantees of Sections 3.3–3.6 make no reference to how the unit was performed before AI was introduced. This section asks when a gated AI-assisted process is at least as likely to produce a deliverable meeting the requirement as the process it supplements, and gives a lower bound with the exact identity behind it. The result is optional, applies only under the conditions stated, and concerns a single workflow unit, not a workflow.

The comparison requires a meaningful operational probability model: a declared population of recurring instances over which the probabilities below are defined. It does not apply automatically to a one-off engineering decision, and the framework does not introduce subjective probabilities to make it apply.

**Setting.** Consider a workflow unit  $\kappa$  (subscripts omitted below) under the following conditions:

- **(S1)** The requirement  $R$  is bivalent:  $R : I \times (O \cup \{\perp\}) \rightarrow \{0, 1\}$ , with  $R(i, \perp) = 0$  (no deliverable does not meet the requirement).
- **(S2)** A probability space is defined on which the following are jointly defined: an instance  $i \sim D$  from a declared operational population; an AI artefact  $o \sim G(i)$ ; the admission event  $E$  of the gate; the output  $F(i)$  of the actual fallback; and the output  $M(i)$  of an **incumbent process**  $M$ , meaning the manual or tool-based process that would perform the unit without AI.  $M(i)$  is a counterfactual: it is the output the incumbent would have produced on instance  $i$ .
- **(S3)**  $P(E) > 0$ .
- **(S4)** Rejected instances are routed to the fallback:  $O_{rel} = o$  on  $E$  and  $O_{rel} = F(i)$  on  $\neg E$ .

The actual fallback  $F$  and the counterfactual incumbent  $M$  are distinct objects. They are linked only by the following assumption.

**Assumption (fallback-performance equivalence).**

$$P(R(i, F(i)) \wedge \neg E) = P(R(i, M(i)) \wedge \neg E).$$

When  $P(\neg E) > 0$ , this is equivalent to  $P(R(i, F(i)) \mid \neg E) = P(R(i, M(i)) \mid \neg E)$ : on the rejected instances, the actual fallback produces a deliverable meeting the requirement with the same probability as the incumbent would have unaided.

This is exactly the assumption needed to cancel the rejected-region terms, and nothing stronger is assumed. It concerns correctness performance on the rejected region, not causal non-interference. For a human fallback, independent-first review or withholding the rejected artefact can provide supporting evidence ([Org]/[Eng]) but does not prove the absence of anchoring; the framework does not model human–AI causal interaction.

**Quantities.** Define:

$$d_{gate} = P(R(i, O_{rel})), \quad d_M = P(R(i, M(i))),$$

$$\eta = P(\neg \hat{S}(i, o) \mid E), \quad \gamma = P(\hat{S}(i, o) \wedge \neg R(i, o) \mid E), \quad q_M(E) = P(\neg R(i, M(i)) \mid E).$$

- $\eta$  is the gate’s selective acceptance error.
- $\gamma$  is the specification-fidelity residual of Section 3.8.
- $q_M(E)$  is the **incumbent’s error rate on the sub-population of instances the gate admits**. It is not the incumbent’s overall error rate.
- $\omega = P(\neg \hat{S}(i, o) \wedge R(i, o) \mid E)$  is the **over-restriction residual**: the probability that an admitted artefact fails the formalization yet meets the requirement. Since  $\neg \hat{S} \wedge R$  implies  $\neg \hat{S}$ ,  $0 \leq \omega \leq \eta$ .

**Proposition 1 (Gated-fallback comparison).** Under (S1)–(S4) and fallback-performance equivalence,

$$d_{gate} - d_M \geq P(E) [q_M(E) - \eta - \gamma]. \quad (3)$$

*Proof.* See Appendix A.

The proof establishes the exact identity

$$d_{gate} - d_M = P(E) [q_M(E) - \eta - \gamma + \omega], \quad (4)$$

from which (3) follows because  $\omega \geq 0$  and  $P(E) > 0$ .

Any claim that “AI-assisted execution is no worse than the incumbent” thus depends on the gate’s error on admitted artefacts ( $\eta$ ), the fidelity of the formalization ( $\gamma$ ) and the incumbent’s performance **on the admitted instances** ( $q_M(E)$ ). The last matters: a gate that admits only easy instances must be compared with the incumbent’s error on those instances, which may be much smaller than its overall error.

**Corollary 1 (Type A).** Suppose the verifier-soundness obligation of a Type A gate holds and the declared population  $D$  lies within  $I^{valid}$ . Then  $\eta = 0$ , hence  $\omega = 0$  (since  $0 \leq \omega \leq \eta$ ), and by (4)

$$d_{gate} - d_M = P(E) [q_M(E) - \gamma].$$

Under the additional ideal assumption  $\gamma = 0$ ,

$$d_{gate} - d_M = P(E) q_M(E) = P(E \wedge \neg R(i, M(i))) \geq 0.$$

Under the ideal assumption  $\gamma = 0$ , the non-degradation conclusion  $d_{gate} \geq d_M$  requires no estimate of the incumbent’s error rate or of the generator’s reliability; the size of the improvement,  $P(E) q_M(E)$ , still depends on the admission probability and on  $q_M(E)$ . The conclusion is conditional on perfect fidelity, which approval does not establish; the primary Type A guarantee remains  $G_A$ .

**Corollary 2 (Type B).** For a Type B gate, with  $D = D_\kappa$  and  $G$  the calibrated generator version, consider the event over calibration (of probability at least  $1 - \delta$ ) on which  $\eta \leq \alpha$  and  $P(E) \geq c$ . On that event, by (3),

$$d_{gate} - d_M \geq P(E) [q_M(E) - \alpha - \gamma].$$

Consequently, if  $\alpha + \gamma \leq q_M(E)$ , then  $d_{gate} \geq d_M$  and moreover

$$d_{gate} - d_M \geq c [q_M(E) - \alpha - \gamma] \geq 0.$$

Here  $q_M(E)$  refers to the sub-population admitted by the calibrated gate, and it must be evidenced for that gate if the comparison is to be claimed.

**Corollary 3 (Types C, D, E).** The framework provides no comparison of this form for Types C, D and E.

For Types D and E, and for C[assist], no AI artefact is admitted, so  $P(E) = 0$  and (S3) fails; for C[adjudicate], (S1) often fails, and otherwise no record clause bounds  $\eta$  for a human gate and fallback-performance equivalence cannot be presumed, because the adjudicator has seen the AI material.

## 4 Staged Classification and Readiness Procedure

The framework’s staged classification and readiness procedure assigns a regime to each workflow unit and records its readiness, in four stages: decomposition and specification (Stage 0); participation, whether and how AI contributes (Stage 1); assurance, the admission rule where AI may produce the deliverable (Stage 2); and record and readiness (Stage 3).

### 4.1 Participation and Assurance Predicates

**Definition 14 (Predicates).** For a workflow unit  $\kappa$ :

*Participation predicates (Stage 1):*

- $P_{assist}(\kappa)$  — AI can materially contribute an artefact or information to the unit, and such contribution is permitted.
- $P_{incumbent}(\kappa)$  — an established deterministic tool path already produces the unit’s deliverable, with stated validation or assurance evidence and a declared validity domain.
- $P_{value}(\kappa)$  — AI participation would provide meaningful, justified value beyond the incumbent path (capability, efficiency or cost), documented with its reasons.
- $P_{deleg}(\kappa)$  — it is technically meaningful and organizationally permitted for an AI-produced artefact to become the unit’s deliverable after admission.

*Assurance predicates (Stage 2):*

- $P_{\text{formal}}(\kappa)$  — an approved formalization  $\hat{S}_\kappa$  exists for a declared scope, and its fidelity obligation is satisfied with stated evidence (Section 3.8).
- $P_{\text{sound}}(\kappa)$  — a deterministic verifier for  $\hat{S}_\kappa$  is implemented and terminates within budget, and its soundness obligation is satisfied with stated evidence (Definition 7). Independence of the verifier from the generator strengthens [Eng] evidence of soundness but is not required.
- $P_{\text{calib}}(\kappa)$  — a meaningful recurring deployment population is declared, and for it and a fixed generator version, a gate has been produced by a calibration procedure valid for stated targets  $(\alpha, c, \delta)$  with  $c > 0$ , using trusted reference labels (Definitions 8–9).

#### Kinds of predicate.

- **Participation predicates** are engineering and organizational judgements.  $P_{\text{value}}$  depends on available AI capabilities, is recorded with its justification, and is evaluated only when  $P_{\text{incumbent}}$  holds.
- **Assurance predicates** concern the unit’s approved formalization and available assurance mechanisms; the regime they select does not depend on generator capability, and only a Type B *guarantee instance* refers to a generator version.
- $P_{\text{sound}}$  and  $P_{\text{calib}}$  are **mechanism-availability** predicates: true only when the mechanism actually exists and its defining property (soundness, or calibration validity) is supported, not when it merely could exist. They determine classification; the record’s remaining obligations (shift monitoring, acceptance-rate reporting, fallback declaration, provenance, human authority) are assessed under readiness (Section 4.2).



available; otherwise Stage 2 falls through to  $C[\text{adjudicate}]$ . The procedure therefore never classifies a unit into A or B on the strength of a mechanism that does not exist.

2. **Target regime:** a regime attainable in principle but not yet available, for instance a verifier that could be written but has not been implemented and supported by evidence. It records an intended improvement and carries no guarantee.
3. **Deployment readiness**  $\text{readiness}(\text{AR}_{\kappa}) \in \{\text{READY}, \text{NOT READY}\}$ : whether every obligation of the assigned assurance record is currently satisfied with adequate evidence (Definition 6).

A unit is not deployed with AI unless its record is READY. A  $C[\text{adjudicate}]$  or  $C[\text{assist}]$  unit without an established authorized judgement or adjudication process remains Type C, is NOT READY for AI deployment, continues under its existing non-AI process, and has no guarantee clause claimed. Readiness also applies to Types D and E: a Type D record whose tool is operated outside its validity domain is NOT READY, and  $G_D$  is not claimed. No additional regime is introduced for unready records. This policy concerns the honesty of assurance claims, not a presumption that more human involvement improves outcomes (Section 3.5).

### 4.3 Composite Activities and Coverage

An activity whose requirement has formalizable and non-formalizable parts is decomposed into workflow units, each classified separately: a report, for example, into a unit whose numerical content is checked deterministically against upstream sources and a unit whose completeness and clarity are adjudicated by an engineer.

**Definition 15 (Composite activity and coverage obligation).** Let an activity  $\tau$  be decomposed into workflow units  $\kappa_1, \dots, \kappa_k$ . Its *composite record* is the tuple  $(\text{AR}_{\kappa_1}, \dots, \text{AR}_{\kappa_k})$ , together with:

- a **release rule**: the activity’s deliverable is released only if, for every workflow unit, either its assurance record is READY and the prescribed participation and assurance mechanism has produced, admitted or authorized the unit’s deliverable, as applicable, or a declared fallback has produced it; or its record is NOT READY and the unit’s existing non-AI process has produced it. Under the prescribed mechanism, for Types A and B the gate admits the AI-produced deliverable; for  $C[\text{adjudicate}]$  the authorized human accepts the AI-produced artefact for release, possibly after revision; for  $C[\text{assist}]$  the authorized human produces the deliverable or decision, with AI material retained as labelled advice; for Type D the established deterministic path produces the deliverable; and for Type E non-AI execution produces it;
- a **coverage obligation**: evidence that the decomposition covers the declared scope of the activity’s requirement. Conceptually, with  $J_{\text{formal}} = \{j : \text{AR}_{\kappa_j} \text{ contains an approved } \hat{S}_{\kappa_j}\}$ ,

$$R_{\tau} \Leftarrow \bigwedge_{j \in J_{\text{formal}}} \hat{S}_{\kappa_j} \wedge R_{\text{res}},$$

where  $R_{\text{res}}$  denotes the residual part of the declared scope of  $R_{\tau}$  that lies within the declared scope of no  $\hat{S}_{\kappa_j}$  and is assigned to the units without an approved formalization (typically human judgement or adjudication); an empty conjunction or an empty residual is read as true.

The coverage obligation is itself a fidelity claim about the decomposition. It is supported by engineering evidence and recorded approval ( $[\text{Eng}]/[\text{Org}]$ ), typically through  $C[\text{adjudicate}]$ , and is not in general mathematically provable. Review narrowing illustrates its importance: when a reviewer is directed to the non-formalized residual only, any part of the requirement that falls neither within some  $\hat{S}_{\kappa_j}$  ( $j \in J_{\text{formal}}$ ) nor within  $R_{\text{res}}$  is checked by no one.

The composite record carries no scalar combined guarantee. The guarantees of its units are of different kinds (Section 3.7) and remain unit-level statements; the probability that all units are correct is not, in general, the minimum of their probabilities of correctness.

### 4.4 Reclassification and the Workflow Boundary

**Reclassification and versioning.** A unit’s regime may change when its requirement or approved formalization changes, a verifier becomes available with supported soundness, calibration becomes feasible for a declared population, AI capabilities change (altering  $P_{\text{value}}$  or  $P_{\text{deleg}}$ ), organizational permissions change, or the population shifts. Assurance records carry version identifiers, and a regime assignment and readiness status refer to a specific record version. A change may alter readiness without changing the regime: a lapsed human authorization or an unmonitored population shift makes a record NOT READY, and a Type B guarantee instance must be recalibrated after a material change in generator version or population.

**Workflow boundary.** The framework establishes **local assurance records for individual workflow units**, each conditional on the unit’s inputs lying in the declared valid-input domain. It does not derive:

- workflow-level dependability;
- non-degradation at the level of the workflow;
- independence between units;
- bounds on error propagation;
- the safety of the engineered system.

Local assurance does not imply workflow-level correctness. A record’s guarantee concerns the unit’s deliverable *for the inputs received* and is claimed only for inputs in  $I_{\kappa}^{valid}$ : a deterministic compliance check fed wrong upstream values may execute correctly on the values it receives, but where those values violate the declared upstream assumptions or lie outside  $I_{\kappa}^{valid}$ , the record’s guarantee is not claimed, and in no case does the check establish anything about the physical structure. Conversely, a downstream gate may detect upstream errors. Degradation of an individual unit is therefore neither necessary nor sufficient for degradation of the workflow.

Each record states its input assumptions. This makes it possible, in future work, to check whether the guarantees of upstream units support the input assumptions of downstream ones. A composition theory of this kind, including iterative loops in which a downstream decision changes an upstream formalization, is outside the scope of this paper.

## 5 Case Study: AI-Assisted Structural Analysis of a Wing-Spar Segment

**Case-study conduct.** This section reports the controlled execution of the wing-spar workflow; the regime classifications describe AI participation within that executed workflow. A retrospective audit of the original case-study record, which preceded the controlled execution, is reported separately in Supplement S7, and full provenance in Supplement S1.

### 5.1 Case-study objective and workflow

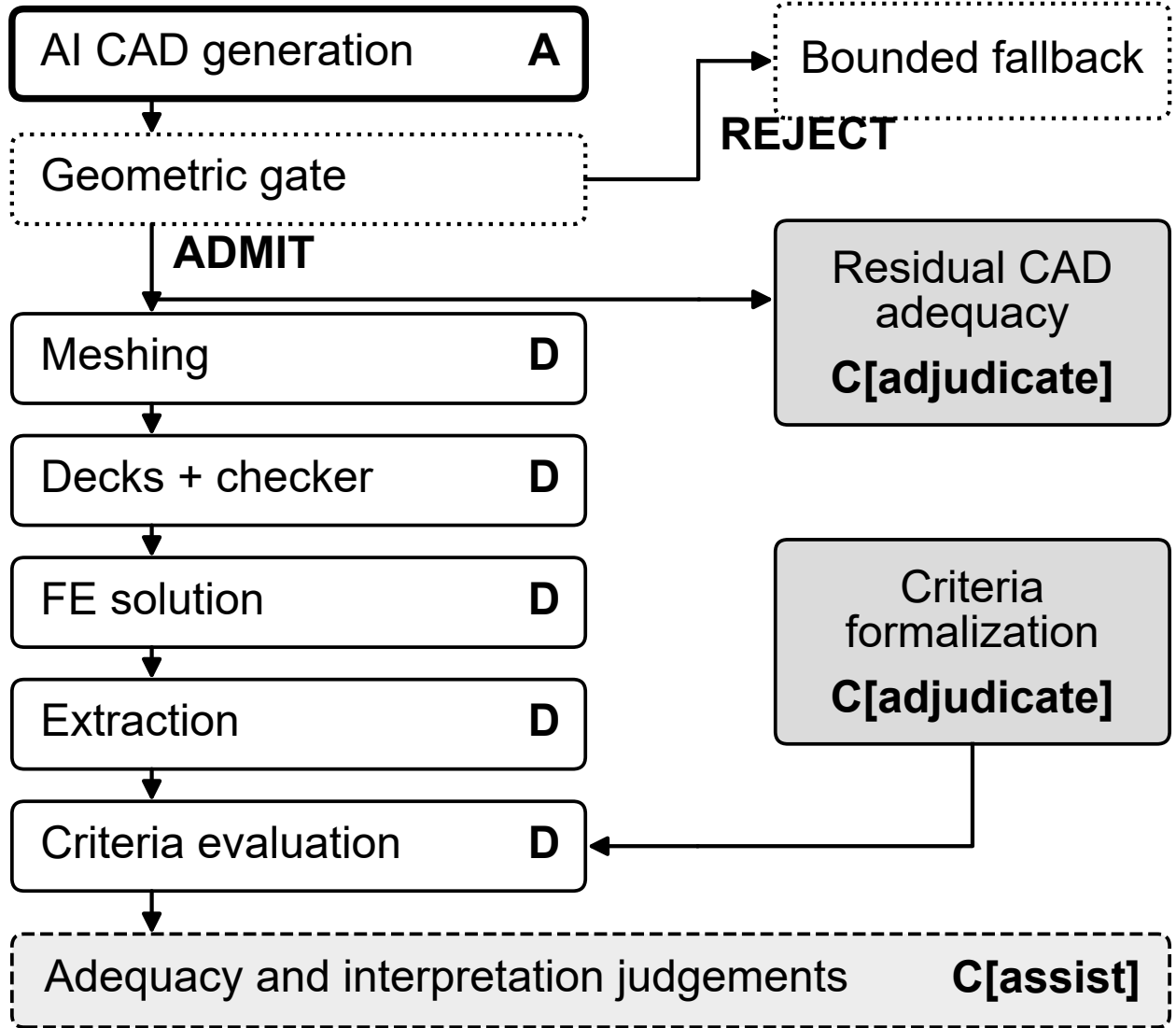
The case study asks whether the framework can be instantiated and applied to a mixed engineering workflow in which AI participates selectively and its outputs are controlled before they propagate downstream. A structural-analysis workflow suits this question because its workflow units differ in kind:

- generating a parametric CAD program is open-ended generative work whose product can nevertheless be measured;
- meshing, finite-element solution and result extraction are already performed by established deterministic tools;
- the adequacy of idealizations and the treatment of unresolved results remain matters of engineering judgement.

The engineering object is a tapered, stiffened, cantilevered aluminium wing-spar segment with lightening holes, idealized as a mid-surface shell model. It is evaluated against project-defined stress and buckling criteria under positive and negative tip loading. Tip deflection and a comparison between linear and geometrically nonlinear analyses are reported and used diagnostically. In the implemented workflow (Figure 5.1), AI-generated CAD enters only through a deterministic gate with a declared fallback, downstream computation uses retained deterministic tools, and engineering judgement is applied explicitly.

The participation and assurance records, requirements, acceptance criteria, tool versions and acceptance rules were fixed before generation began. The workflow was then executed once, end to end.

The study is an author-led research case study, not a certification or independent assurance exercise; its scope limits are stated in Section 6.2.



**Figure 5.1.** Executed workflow and regime allocation. AI-generated CAD (Type A) enters only through a deterministic gate (ADMIT/REJECT) with a predeclared bounded fallback; meshing, deck handling, solution, extraction and evaluation use deterministic tools (Type D). Residual CAD adequacy and criteria formalization are C[adjudicate]; adequacy and interpretation judgements are C[assist]. Representative units are shown; the executed set comprises 17 units (A 1, D 6, C[adjudicate] 4, C[assist] 6; no B or E), all READY. Dotted outlines mark mechanisms, not workflow units. The predeclared fallback allowed at most three attempts through a bounded repair loop, then engineer-authored CAD through the same gate, then a recorded halt; it was not invoked.

## 5.2 Instantiating the classification

Procedure 1 decomposes the workflow, which as a whole receives no regime, into a controlled executed set of 17 workflow units. Each unit has a participation and assurance record (regime, mechanism, fallback where applicable, obligations and evidence) recorded before execution, and all 17 assurance records were READY before the run. In this executed set, one unit is Type A, six are Type D, four are C[adjudicate] and six are C[assist]. These counts refer only to the 17-unit executed set, not to the retrospective audit of the original case-study record reported in Supplement S7. Table 5.1 lists representative workflow units, which together cover every regime instantiated.

**Table 5.1.** Representative workflow units of the wing-spar workflow and their regimes.

| Workflow unit                                        | AI role in this execution                                | Regime        | Acceptance / execution mechanism                                                                                                                          |
|------------------------------------------------------|----------------------------------------------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAD program generation                               | AI generated the program                                 | A             | Deterministic gate measures the produced geometry against an approved operational formalization; admission only if every clause is met; declared fallback |
| Residual CAD adequacy                                | AI-generated geometry under review; AI compiled evidence | C[adjudicate] | Study author accepts or rejects the artefact on aspects outside the formalized scope                                                                      |
| Mesh generation                                      | None                                                     | D             | Scripted mesher with frozen quality rules                                                                                                                 |
| Solver input-deck generation                         | None                                                     | D             | Deterministic deck writer and separate deterministic deck checker                                                                                         |
| Finite-element solution                              | None                                                     | D             | Pinned solver within a declared validity domain                                                                                                           |
| Result extraction and numerical checks               | None                                                     | D             | Deterministic parsers, equilibrium checks, convergence metrics                                                                                            |
| Formalization of acceptance criteria                 | AI-assisted draft of the criteria                        | C[adjudicate] | Study author approves and versions the criteria before any result exists                                                                                  |
| Evaluation of approved criteria                      | None                                                     | D             | Deterministic evaluator                                                                                                                                   |
| Adequacy and interpretation judgements (three units) | Advisory: AI compiled evidence and proposed findings     | C[assist]     | Study author produces the judgement                                                                                                                       |

*Note.* The AI-role column states the AI contribution recorded in each unit’s frozen review or provenance record; “None” means that no AI-produced material entered the unit’s execution.

**Type A.** CAD program generation is the single Type A unit: the program is admitted only through a deterministic geometric gate against an approved operational formalization (Section 3.3). Because that formalization covers only specified geometric properties, residual CAD adequacy (whether the geometry suits the analysis and the design intent) is a separate C[adjudicate] unit. Together, the two units are intended to cover the geometric requirement; that coverage is a fidelity claim supported by the recorded approval, not a proved property (Section 4.3).

**Type D.** Meshing, deck generation, finite-element solution, result extraction, evaluation of the approved criteria and engineering report assembly retain established deterministic paths, whose case-specific validation evidence is summarized in Supplement S4. Evaluation is Type D rather than Type A because the deterministic evaluation is the task itself (Sections 3.3 and 3.6).

**Type C.** C[adjudicate] covers the geometry specification, residual CAD adequacy, the formalization of the acceptance criteria, whose fidelity was not treated as mechanically decidable (Section 3.8), and the case-study engineering report narrative. C[assist] covers load-case and idealization decisions, discretization and analysis adequacy, the interpretation of results, the response to unresolved criteria and the study-level review (Section 3.5).

**No Type B or Type E.** Neither is instantiated; Type B would require a recurring deployment population, which a one-off study lacks (Section 3.4).

The allocation is therefore selective: deterministically admitted AI generation is confined to the one unit where deterministic verification is tractable; elsewhere, AI-produced drafts are released only through authorized human acceptance (C[adjudicate]), AI advice informs human judgement (C[assist]), and established deterministic tools are retained (Type D).

### 5.3 AI generation with deterministic admission

In this Type A unit, the generator does not establish the acceptability of its own output. The candidate program was generated by Claude Opus 5.5 in a fresh context with no conversation history. The generator received the frozen generation prompt verbatim. It had no access to the engineer-adopted reference program, the verifier implementation or any downstream result. This is a procedural separation of generation from acceptance, not independence in a formal assurance sense. The model, provider, prompt and generation context of the generation call were recorded (Supplement S1.4). The execution plan specified a “High” reasoning-effort setting, but the model-invocation interface used for the run did not allow that setting to be enforced. The deterministic gate was unaffected, and the run should not be interpreted as evidence of generator performance at a specified reasoning-effort setting.

**The deterministic gate.** The gate implements the approved geometric formalization. It executes the candidate program within a fixed time budget under execution-containment monitoring, requires exactly one STEP file declaring metre units, and re-imports that file in a fresh process. The re-imported geometry is checked for:

- topology and face classification;
- geometric planes and outlines;
- hole and stiffener geometry;
- areas and overall extents.

Each check has a declared tolerance; the full clause list is in Supplement S3. The gate admits the artefact only if every clause of the formalization is satisfied, including all measured checks.

**Evidence for the gate.** The gate’s soundness is supported by engineering evidence, not by formal proof. Before the study, a pre-run suite of 32 cases reproduced every expected decision. It included controlled mutations of an engineer-adopted reference program, which the gate had to reject, and was complemented by logic-level tests and an author-led code review.

**Predeclared bounded fallback.** At most three sequential attempts were allowed through a bounded repair loop. After three rejections, an engineer-authored program would have had to pass the same gate; failing that, the unit would halt and the halt would be recorded.

**Outcome.** The first candidate executed within its budget, produced a single metre-unit STEP file and passed all 23 geometric and validity checks, with no containment violations detected (Table 5.2). It was admitted at the first attempt and used unmodified; the repair loop and the fallback were not invoked. A single admission says nothing about the generator’s reliability or its admission rate across instances. The assurance comes from the gate, not from the outcome.

**Residual scope.** Admission establishes conformance to the approved geometric formalization within tolerance, conditional on verifier soundness. It does not establish every engineering property of the artefact. In the separate C[adjudicate] review, the study author judged the geometric representation adequate for the study and recorded, rather than repaired, defensive fallback branches unnecessary in the pinned environment, since the predeclared protocol allowed modification only through the bounded repair loop.

### 5.4 Deterministic downstream execution and reproducibility

After admission, no further AI-generated engineering artefacts entered the computational chain. The admitted geometry passed to scripted meshing, deterministic deck generation and checking, a pinned solver (CalculiX 2.23) and deterministic extraction with numerical checks; the 21 decks cover linear static, linear buckling and one geometrically nonlinear analysis (Table 5.2).

**Forward-carrying controls.** Three controls carry the admission decision forward:

- the mesher verifies that its input geometry was admitted by the gate and produced by the admitted program;
- the solver runs only when every deck has passed a separate deterministic checker, which reconstructs the tip load-introduction constraints and confirms the completeness of loads and supports;
- the evaluator accepts only approved criteria and material data.

**Table 5.2.** Execution and integrity evidence.

| Stage               | Control / evidence                                                                                                                        | Observed outcome                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| CAD gate (Type A)   | Bounded execution under containment monitoring; 23 geometric and validity checks against the approved formalization; predeclared fallback | ADMIT at attempt 1; 23/23 checks; no containment violations detected; fallback unused; candidate unmodified <sup>a</sup>           |
| Mesh                | Lineage check to gate-admitted geometry; frozen quality rules; geometry-conformity check; repeated meshing                                | 3 levels of quadratic triangular shell elements (3,218 / 12,152 / 47,046 elements); all pass; same-host canonical hashes identical |
| Solver input decks  | Separate deterministic deck checker; regeneration comparison                                                                              | 21/21 pass; byte-identical regeneration                                                                                            |
| Solver              | Pinned binary hash; completion and error/warning record                                                                                   | 21/21 complete; no errors or warnings recorded                                                                                     |
| Numerical integrity | Precision-aware force and moment equilibrium; output-consistency conditions                                                               | 12/12 linear static jobs pass; all conditions satisfied                                                                            |
| Frozen inputs       | Hash manifest and solver pin re-verified after the run                                                                                    | 61/61 manifest entries and solver binary unchanged (contemporaneous report; later re-hash: Supplement S1.5)                        |

<sup>a</sup> An earlier invocation failed before the candidate executed (operator-side path error) and consumed no generation attempt (Supplement S3.5).

The controls are independent of the geometry’s source by design, although the single run did not test this.

**Reproducibility.** A 61-entry hash manifest fixed before generation pinned the specifications, criteria, material data, analysis plans, core tools and solver binary. The solver analyses were executed once and not independently repeated. This evidence establishes repeatability of the implemented process, not physical validity of the results, adequacy of the engineering model or formal tool qualification.

### 5.5 When a favourable numerical result is not sufficient for PASS

**The convergence rule.** The stress clauses require a quantity both to be converged under the frozen mesh-convergence rule and to meet its threshold. Convergence is judged between the medium and fine meshes, with tolerances fixed before the run:

- 1 % for displacement and buckling factors;
- 5 % for the stress metric.

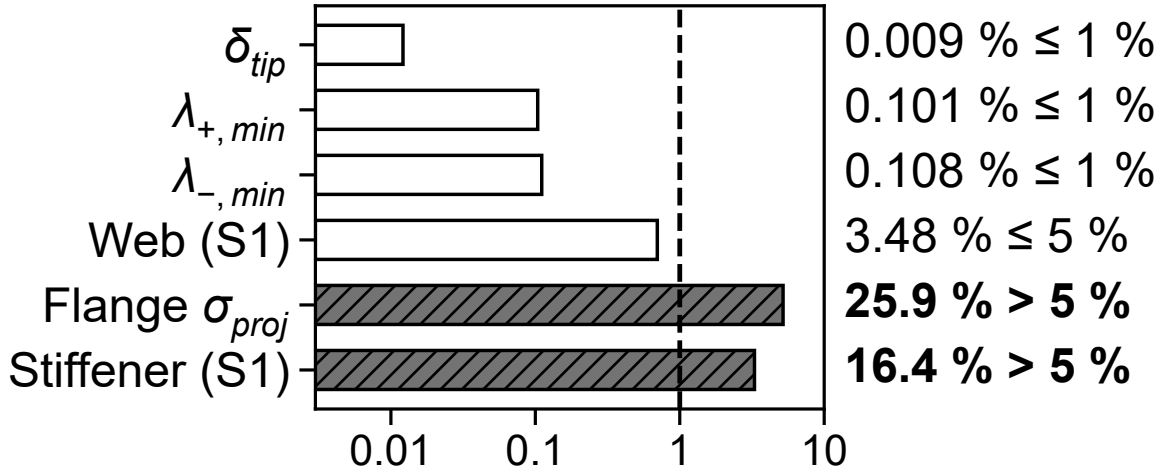
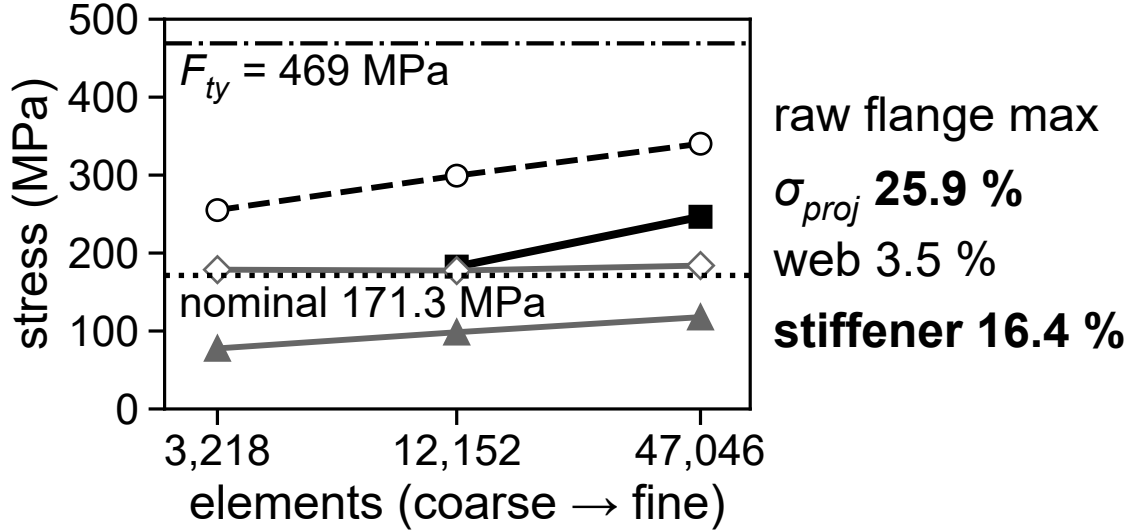
Tip displacement and both buckling factors converged well within these tolerances. The stress results did not converge uniformly (Figure 5.2).

**Three distinct quantities.** The assessment keeps three quantities distinct:

- the raw finite-element maximum in each component group;
- the project metric  $\sigma_{proj}$  that enters the criteria;
- a nominal beam-theory reference at the root.

*Flange.* At positive ultimate load, the raw flange maximum sits at the clamped root corner and rose with refinement (Figure 5.2). The frozen rule anticipated this case. Such a peak is reported separately as a non-converged local clamp peak, not removed as a singularity, and  $\sigma_{proj}$  is taken over the flange outside a predeclared root exclusion zone, bounded below by the nominal reference.

In the executed run, however, the maximum outside the exclusion zone ( $\sigma_{proj}$ ) moved to the opposite end of the spar: the tip corner where the load is introduced through the rigid tip constraint, a location the rule had not anticipated. That peak also grew with refinement, from 182.6 MPa to 246.5 MPa (25.9 %), so  $\sigma_{proj}$  was NOT\_CONVERGED.

**(a) Medium-to-fine change ÷ tolerance****(b) Stress at positive ultimate load**

**Figure 5.2.** Mesh-convergence evidence under the frozen rule. (a) Medium-to-fine relative change of the tip displacement  $\delta_{tip}$ , the smallest positive buckling factors  $\lambda_{+,min}$  and  $\lambda_{-,min}$ , and the web, flange ( $\sigma_{proj}$ ) and stiffener stress metrics (“S1” in the figure denotes the group-maximum stress rule of the frozen engineering plans; Supplement S5.1), divided by the frozen tolerance (1 % for displacement and buckling factors, 5 % for the stress metrics; dashed line: tolerance). Hatched bars and bold values exceed the tolerance (NOT\_CONVERGED): the flange project metric  $\sigma_{proj}$  (25.9 %) and the stiffener stress (16.4 %). (b) Stresses at positive ultimate load versus mesh level (3,218 / 12,152 / 47,046 elements): raw flange root-corner maximum (255.1 / 299.3 / 340.2 MPa), flange project metric  $\sigma_{proj}$  (medium 182.6, fine 246.5 MPa), stiffener and web peaks, the nominal root reference (171.3 MPa) and  $F_{ty} = 469$  MPa; bold labels mark NOT\_CONVERGED quantities. The post-hoc interior-flange diagnostic is not plotted and was not a verdict input. Full values: Supplement S5.

*Web and stiffener.* The web converged (177.5 MPa to 183.9 MPa, 3.5 %). The stiffener peak lies at a stiffener–flange junction; junctions are included in the metric by design. It changed from 98.4 MPa to 117.8 MPa (16.4 %) and was NOT\_CONVERGED. The evidence characterizes the root, tip and junction peaks as mesh-sensitive, unresolved local responses; the study does not establish their mathematical character.

**Why a low stress ratio did not yield PASS.** For the flange at positive ultimate load,  $\sigma_{proj} = 246.5$  MPa against a yield allowable of  $F_{ty} = 469$  MPa (MIL-HDBK-5J, 2003) gives a ratio of 0.526; the other groups and load levels give lower ratios. By magnitude alone this suggests ample margin. But the criterion applies to a converged metric, and the metric was not converged. The evaluator therefore returned NOT\_CONVERGED for all four stress clauses, at limit and ultimate load in both senses, and the study’s final disposition was ADJUDICATE.

ADJUDICATE is the study-level disposition used when an evaluated criterion cannot be concluded as PASS under its predeclared rule and requires engineering judgement. Final dispositions (PASS, ADJUDICATE, Not applicable), like the unit outcomes of Section 5.6, are study-level outcome labels, distinct from the evaluator’s clause verdicts, from deployment readiness and from the regime guarantees. ADJUDICATE is also distinct from the C[adjudicate] participation regime, in which a human accepts or rejects an AI-produced artefact. An ADJUDICATE disposition establishes neither structural failure nor structural safety; it records that the available evidence is insufficient for a conclusion under the predeclared rule.

**Post-hoc diagnostic (not a verdict input).** Only after the verdicts were recorded did a diagnostic examine the interior flange,  $0.20 \text{ m} \leq x \leq 2.20 \text{ m}$  along the span, which excludes 0.20 m at each end.

- **Flange:** the interior stress changed from 179.4 MPa to 179.7 MPa (about 0.2 %), suggesting that the flange non-convergence is confined to the end regions.
- **Stiffener:** the stiffener was not resolved. After excluding stiffener elements within 0.05 m of the flanges, its medium-to-fine change remained about 13 %.

The diagnostic was not part of the predeclared rule, was not passed to the evaluator and changed no verdict. It helps explain the flange outcome; it does not establish compliance.

**A limitation of the approved formalization.** The predeclared stress rule represented the root-clamp region but not the tip load-introduction region revealed by the run. This is the requirement-to-formalization gap of Section 3.8, observed in execution: approval did not ensure that the rule represented the requirement in every region the analysis produced. The outcome was recorded as a limitation, and no criterion was changed. Any refinement of the load introduction, local modelling or stress metric would be a new version of the formalization, to be fixed before a future rerun (Section 4.4), not adopted retrospectively. Because the assurance record fixed the metric, the tolerance and the exclusion zone in advance, the gap surfaced as adjudication instead of being tuned away after the result was seen.

## 5.6 Final verdicts and engineering adjudication

Table 5.3 lists, for each clause, the deterministic evaluator verdict and the study’s final disposition: NOT\_CONVERGED (Q-1, Q-2) becomes ADJUDICATE; PASS (Q-3) remains PASS; the reporting-only verdict REPORTED (Q-4) has no disposition (Not applicable); and SUPPORTED (Q-5) becomes ADJUDICATE, as predeclared.

**Buckling: PASS.** The smallest positive buckling factor was  $\lambda \approx 4.835$  under positive load and  $\lambda \approx 12.086$  under negative load, against a project threshold of  $\lambda \geq 1.5$ . The evaluator’s rule requires a complete run with a usable positive factor, convergence of that factor within 1 % and the converged factor to meet the threshold. All three held (medium-to-fine change about  $10^{-3}$ ), so the evaluator returned PASS. The same evaluation that withheld PASS from the stress clauses granted it here, because the evidence the rule required was present.

The C[assist] interpretation identified the first mode in both senses as a global lateral(-torsional) mode of the laterally unrestrained cantilever, so the factors depend on the idealization of tip and lateral restraint. The result is a perfect-geometry linear-bifurcation outcome under the project criteria, not certification-level buckling assurance.

**Table 5.3.** Clause verdicts and final dispositions.

| Clause                                                      | Evaluator verdict | Final disposition        |
|-------------------------------------------------------------|-------------------|--------------------------|
| Q-1+ / Q-1- — stress at limit load (positive / negative)    | NOT_CONVERGED     | ADJUDICATE               |
| Q-2+ / Q-2- — stress at ultimate load (positive / negative) | NOT_CONVERGED     | ADJUDICATE               |
| Q-3+ / Q-3- — linear buckling (positive / negative)         | PASS              | PASS                     |
| Q-4 — tip deflection (reporting only)                       | REPORTED          | Not applicable           |
| Q-5 — linear vs geometrically nonlinear diagnostic          | SUPPORTED         | ADJUDICATE (predeclared) |

*Note.* Assurance-record readiness: 17/17 READY. Post-run **unit outcomes** (study-level labels for each workflow unit, distinct from readiness, regime guarantees and the clause verdicts above): PASS 12, ADJUDICATE 3, Not applicable 2, FAIL 0.

**Q-5: a favourable cross-check under a predeclared scope limit.** Q-5 compares the linear response at positive ultimate load with a geometrically nonlinear analysis (CalculiX NLGEOM). All relative differences were within the frozen 5 % threshold, and the evaluator returned SUPPORTED. The flange comparison used the non-converged project metric of Section 5.5, so it was treated only as a diagnostic comparison.

The disposition is nevertheless ADJUDICATE. Before any analysis, the study had declared that Q-5 requires adjudication, because the finite-rotation validity of the linearized tip-load relation had not been established. The results support that caution:

- the tip-deflection ratio at ultimate load ( $\delta/L = 0.0131$ , with span  $L = 2.4$  m) exceeded the 0.01 escalation threshold;
- the nonlinear run passed every integrity check but showed a small moment-equilibrium deficit (relative error about  $2.01 \times 10^{-4}$ ), consistent with the known limitation of the linearized relation. It was treated as diagnostic, not as release-supporting evidence.

A favourable numerical cross-check does not override a predeclared scope limitation.

**The human role.** Human judgement remained explicit: the study author, not AI advice, decided what could be concluded where deterministic evidence stopped short, and no criterion was modified to force acceptance; the guarantee is procedural, not one of correct judgement (Section 3.5).

**READY is not PASS.** All 17 assurance records were READY, meaning that every obligation was satisfied with adequate evidence (Definition 6), yet readiness did not predetermine the outcome: of the 17 units with READY records, 12 ended PASS, 3 ADJUDICATE and 2 Not applicable (note to Table 5.3). Unit outcomes record each unit’s post-run result against its own requirement, and they are distinct from clause verdicts. The evaluation unit, for example, recorded PASS because it executed the approved criteria as specified, even though several of the resulting dispositions were ADJUDICATE.

**What the case study shows.** Within this single, author-led study, the AI-generated CAD artefact entered the analysis only through deterministic admission under a declared fallback, established deterministic tools were retained, and human judgement handled residual and unresolved evidence. The predeclared evidence requirements, not favourable numerical values, controlled the outcome, and the computational process can be inspected and re-executed within the demonstrated scope. These findings concern the implementability and behaviour of the framework in one workflow.

## 6 Discussion

### 6.1 Transferability beyond the case study

The framework’s objects (workflow units and their requirements, approved operational formalizations, participation decisions, participation and assurance mechanisms with their fallbacks, evidence obligations and readiness, and the validity domains of established tools) are not specific to structural engineering, and its regimes are defined by the participation and assurance mechanism available for a workflow unit, not by engineering discipline or by a level of

AI autonomy. The framework structure is therefore transferable in principle, but each application requires a fresh decomposition, formalization, evidence assessment and validity-domain decision.

Brief illustrations show how the regime semantics carry over; they are not claims of validated deployment.

- **Type A** fits workflow units in which AI generates a non-trivial artefact that a sound deterministic verifier checks against an approved formalization. Examples are machine-generated geometry measured against an approved dimensional specification, or a generated configuration checked against a machine-readable interface and rule set.
- **Type D.** A deterministically checkable output does not by itself make a unit Type A. Evaluating a computed stress, thermal margin or clearance against an approved limit is Type D when an established deterministic tool path already produces that evaluation (Section 3.6).
- **Types B, C and E** retain the conditions of Sections 3.4–3.6: Type B requires a genuinely recurring, calibrated setting; Type C retains authorized human judgement or acceptance; and Type E records justified non-participation.

The case for cross-domain applicability therefore rests on the domain-independent structure of the participation and assurance records and the classification procedure. It is an argument from structure, not empirical validation across domains. The case study shows only that the framework can be instantiated and applied in one executed structural-engineering workflow, in which Types A and D, C[adjudicate] and C[assist] were instantiated but neither Type B nor Type E (Section 5.2). It does not demonstrate readiness, AI reliability or engineering effectiveness in other domains, nor that any particular workflow will yield a given allocation. Section 6.4 identifies the evidence that would strengthen the case for transferability.

## 6.2 Limitations

### Framework-level limitations

Predicates such as  $P_{value}$ ,  $P_{deleg}$  and  $P_{formal}$  record engineering and organizational judgements that the framework does not establish mechanically. Different decompositions may yield different allocations, and coverage of a requirement by a decomposition is itself an assurance obligation, supported by evidence rather than proved (Section 4.3).

Approving an operational formalization does not prove its fidelity to the underlying requirement. For natural-language requirements, fidelity rests on engineering evidence and authorized interpretation. Refinement or equivalence evidence is possible only where the authoritative requirement is itself formal (Section 3.8).

### Each guarantee carries its own conditions.

- **Type A** is conditional on verifier soundness relative to the approved formalization. Soundness is often supported by testing, mutation testing and review rather than proof; verifier independence may strengthen that evidence but does not establish soundness.
- **Type B** is specific to a declared recurring population, a fixed generator version, reference labels and stated calibration assumptions, and is unavailable for one-off units.
- **Type C** is procedural: authorized human acceptance or judgement does not ensure correctness, and automation bias, anchoring and weak review remain possible.
- **Type D** retains an established path only within its declared validity domain and stated evidence, and does not cover model adequacy.

A READY assurance record has its obligations satisfied with adequate evidence, but judging adequacy is itself a judgement. Readiness neither replaces the regime’s guarantee nor implies a PASS (Section 5.6).

Assurance records are local to workflow units: the framework derives no workflow-level dependability, error-propagation bound or system-safety claim (Section 4.4). Proposition 1 (Section 3.9) concerns a single workflow unit under its stated recurring-probability setting and assumptions, including fallback-performance equivalence. It is not a workflow result and does not apply to one-off decisions such as the wing-spar case.

Requirements, formalizations, verifiers, tools, generator versions and populations change, so assurance records must be versioned and re-evaluated. Type B gates must be recalibrated after a material change in population or generator version, and changed verifier or tool evidence may require renewed assurance for Type A and Type D.

### Case-study limitations

The study was author-led. Its engineering reviews and adjudications were decisions of the study author, not independent third-party assurance.

The structural model is idealized:

- a mid-surface shell representation;
- a linearized rigid tip-load relation without established finite-rotation validity;
- perfect-geometry linear buckling;
- project criteria motivated by, but not equivalent to, certification requirements.

The results are interpretable only within these modelling choices.

The material allowables did not govern any disposition: the stress-clause dispositions were governed by non-convergence rather than by exceedance of an allowable. The allowables were taken from MIL-HDBK-5J (2003), Table 3.7.6.0(b1), for 7075-T651 plate, A-basis. The governing  $F_{ty}$  values are 469 MPa for the 18 mm flange and 462 MPa for the 8 mm web and stiffeners, using the minimum of the listed L and LT values for the corresponding thickness ranges.

### Execution and reproducibility.

- OS-level isolation of the AI-generated code execution was not demonstrated.
- Mesh determinism was shown on the same host only.
- The solver analyses were not independently repeated, and cross-platform reproducibility was not tested.
- Reproducibility of the implemented process is not evidence of physical model validity.

No comparable manual baseline exists, so no reduction in engineering time, effort or cost is claimed.

Within these boundaries, the case study demonstrates that the framework can be instantiated and applied in an executed engineering workflow with selective AI participation, deterministic admission of the AI-generated artefact, inspectable evidence handling and explicit routing of unresolved evidence to an ADJUDICATE disposition and engineering judgement. It does not establish structural safety, certification or regulatory compliance, independent assurance, general AI reliability, cross-platform reproducibility, workflow-level non-degradation, system-level dependability or productivity improvement.

## 6.3 Dependability, certification and accountability

Dependability motivates the framework, but the framework addresses it at a specific level: how AI participates in individual workflow units, and what assurance mechanism and evidence justify that participation.

Certification is a different matter. It is a determination made under applicable standards, organizational processes, independence requirements and the authority of the relevant regulator. Participation and assurance records can support assurance and certification activities by making the applicable mechanisms, evidence, fallbacks, readiness and formalization versions explicit and inspectable. The framework does not itself establish certification, regulatory compliance or the safety of the engineered system.

The unit-level perspective complements frameworks that characterize AI applications by the relationship between human and machine authority, such as EASA's Level 1 and Level 2 framework for machine-learning applications. Such levels concern how authority and autonomy are distributed between human and AI. The present framework asks, unit by unit, how AI participates and under what assurance mechanism an artefact or decision may enter the engineering workflow. A single application could be described in both ways, and the two descriptions answer different questions; no mapping or equivalence between EASA levels and the participation and assurance regimes is claimed.

For governance purposes, the same unit-level structure identifies, for each regime, where participation, acceptance or execution decisions are taken and where the associated evidence is held (Table 6.1). This is a description of governance, not a theory of legal liability.

Formalization approval is an accountable governance act, not proof of fidelity (Section 3.8). The framework does not determine who holds legal duties, what level of independence a regulator requires, which standards apply, or whether certification is granted; these remain external to it.

**Table 6.1.** Decision or execution locus and evidence/governance focus by regime.

| Regime        | Decision or execution locus                                         | Evidence and governance focus                                                                                                                         |
|---------------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| A             | Deterministic verifier against an approved formalization            | Approval of the formalization; evidence for verifier soundness; governance of admission and fallback; versioning of verifier and formalization        |
| B             | Calibrated statistical gate                                         | Declared recurring population; fixed generator version; reference labels; calibration assumptions; monitoring and recalibration after material change |
| C[adjudicate] | Authorized human acceptance or rejection of an AI-produced artefact | Identified, authorized adjudicator; recorded acceptance, rationale and AI provenance                                                                  |
| C[assist]     | Authorized human judgement, with AI material as labelled advice     | Identified, authorized person producing the judgement or deliverable; recorded rationale and provenance of AI advisory material                       |
| D             | Established deterministic tool path                                 | Stated validation evidence; operation within the declared validity domain                                                                             |
| E             | Non-AI execution                                                    | Governance as in the non-AI activity; record that no AI artefact contributed                                                                          |

## 6.4 Future work

Four directions follow from Sections 5 and 6.2.

### Stronger assurance evidence.

- The link from requirement to approved formalization would benefit from systematic traceability, independent re-derivation and review, and structured change control.
- Where feasible, Type A verifier soundness could be strengthened by formal proof or proof-producing verification. This would strengthen Type A, not become a requirement for every Type A unit.
- Execution assurance would benefit from OS-level isolation of AI-generated code, and reproducibility evidence from independent reruns and cross-platform repetition.

### Broader empirical evidence.

- A genuinely recurring deployment satisfying the population, calibration and monitoring conditions of Section 3.4 is needed to exercise Type B empirically.
- An independent second engineering case, with independent decomposition, classification and formalization, would add evidence of transferability without by itself establishing it across domains.
- Type C units raise empirical questions of automation bias, anchoring, the design of advisory material, reviewer independence and the quality of recorded rationale.
- Any efficiency study would require a predefined comparative baseline and measurement protocol, and its outcome cannot be presumed.

**Case-specific refinement.** The wing-spar stress rule anticipated the root clamp region but not the tip load-introduction region (Section 5.5). Any improved metric, exclusion rule or local model would be a new formalization version, fixed before a new run, not tuned retrospectively to convert the present ADJUDICATE outcome into PASS.

**Architecture and theory.** Assurance records remain local. A composition theory remains open (Section 4.4). It would need to cover:

- the propagation of rejected outputs and fallback deliverables;
- iterative workflows in which a downstream decision revises an upstream formalization;
- dependencies among assurance records.

Recurring workflow-unit patterns might support reusable assets: assurance-record templates, formalization and evidence patterns, verifier and tool-path records, and classification support. Their reuse would remain subject to checks

of scope, version and validity domain. Any automated classifier would itself be a workflow unit to be classified and assured under the framework.

## 7 Conclusion

Engineering workflows contain units of work that call for different forms of AI participation and assurance. This paper has developed a formal framework that pairs an explicit AI-participation decision for each workflow unit with the assurance required to justify it. Each unit’s participation and assurance record identifies the requirement, any approved operational formalization, the applicable mechanism, any declared fallback, evidence obligations with explicit evidence kinds and the regime guarantee, and carries a deployment-readiness status. The regimes (deterministic or statistically calibrated admission, authorized human judgement or adjudication, retention of an established deterministic path, and explicit non-participation) carry different kinds of guarantee and admit no scalar ordering. The framework also separates formalization fidelity from verifier soundness, keeps classification, target regime and deployment readiness distinct through a staged procedure, and states conditions under which a gated unit can be compared with an incumbent process on a recurring population.

The executed wing-spar case showed that the framework can be instantiated and applied in an engineering workflow with selective AI participation. The AI-generated CAD artefact entered the analysis only through deterministic admission, established deterministic tools were retained, and human judgement and adjudication handled residual and unresolved evidence. The predeclared evidence requirements, not favourable numerical values, controlled the outcome: favourable stress values did not yield PASS without the required convergence evidence, and READY assurance records did not imply PASS.

These conclusions concern local, unit-level assurance in a single author-led case and are bounded by the limitations stated in Section 6.2; validating the framework beyond this case will require independent engineering applications.

## References

- Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004. doi: 10.1109/TDSC.2004.2.
- European Union Aviation Safety Agency. EASA concept paper: Guidance for level 1 & 2 machine learning applications. Concept Paper Issue 02, European Union Aviation Safety Agency (EASA), March 2024. URL <https://www.easa.europa.eu/en/downloads/139504/en>. Published 6 March 2024; Issue 03 exists only as a proposed issue (consultation June–August 2026).
- Robin Bloomfield and John Rushby. Where AI assurance might go wrong: Initial lessons from engineering of critical systems, 2025. Presented at the UK AISI Conference on Frontier AI Safety Frameworks (FAISC 24), Berkeley, CA, November 2024.
- Rundi Wu, Chang Xiao, and Changxi Zheng. DeepCAD: A deep generative network for computer-aided design models. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6752–6762. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00670.
- Jingwei Xu, Chenyu Wang, Zibo Zhao, Wen Liu, Yi Ma, and Shenghua Gao. CAD-MLLM: Unifying multimodality-conditioned CAD generation with MLLM, 2024. arXiv preprint; v3, August 2025.
- Sumit Kumar, Sarthak Kapoor, Harsh Vardhan, and Yao Zhao. Generative AI for CAD automation: Leveraging large language models for 3D modelling, 2025. arXiv preprint.
- Licheng Zhang, Bach Le, Naveed Akhtar, Siew-Kei Lam, and Duc Ngo. Large language models for computer-aided design: A survey. *ACM Computing Surveys*, 58(9):1–39, 2026. doi: 10.1145/3787499. Fifth author listed as Duc Ngo in the ACM/Crossref record and as Tuan Ngo on arXiv:2505.08137.
- C. Michael Holloway. Understanding the overarching properties. NASA Technical Memorandum NASA/TM-2019-220292, NASA Langley Research Center, July 2019. URL <https://ntrs.nasa.gov/citations/20190029284>.
- EUROCAE WG-114 and SAE G-34. ED-324/ARP6983: Development and assurance considerations for aeronautical systems and equipment implemented with machine learning. Draft standard, EUROCAE Working Group 114

- jointly with SAE Committee G-34. URL <https://www.eurocae.net/working-group/wg-114/>. Not yet published; the EUROCAE WG-114 page lists status “Draft” with target publication date 31 December 2026 (accessed 2 October 2026). Cited without a publication year because it is unpublished.
- Bastian Luettig, Yassine Akhiat, and Zamira Daw. ML meets aerospace: challenges of certifying airborne AI. *Frontiers in Aerospace Engineering*, 3:1475139, 2024. doi: 10.3389/fpace.2024.1475139.
- National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1, National Institute of Standards and Technology, January 2023.
- ISO/IEC. ISO/IEC 42001:2023 information technology — artificial intelligence — management system. International standard, Edition 1, International Organization for Standardization and International Electrotechnical Commission, Geneva, December 2023. URL <https://www.iso.org/standard/81230.html>.
- Richard Hawkins, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu, and Ibrahim Habli. Guidance on the assurance of machine learning in autonomous systems (AMLAS). Guidance Version 1.1, Assuring Autonomy International Programme (AAIP), University of York, March 2021. URL <https://www.york.ac.uk/media/assuring-autonomy/documents/AMLASv1.1.pdf>. Version 1 also available as arXiv:2102.01564.
- Robin Bloomfield and John Rushby. Assurance 2.0: A manifesto. In Mike Parsons and Mark Nicholson, editors, *Systems and Covid-19: Proceedings of the 29th Safety-Critical Systems Symposium (SSS’21)*, pages 85–108, York, UK, 2021. Safety-Critical Systems Club.
- Sung Une Lee, Liming Zhu, Md Shamsujjoha, Liming Dong, Qinghua Lu, Jieshan Chen, and Lionel Briand. A structured approach to safety case construction for AI systems, 2026. arXiv preprint, version 3 (6 March 2026); v1 titled “Constructing Safety Cases for AI Systems: A Reusable Template Framework”.
- Sina Mohseni, Haotao Wang, Chaowei Xiao, Zhiding Yu, Zhangyang Wang, and Jay Yadawa. Taxonomy of machine learning safety: A survey and primer. *ACM Computing Surveys*, 55(8):1–38, 2022. doi: 10.1145/3551385. Published online 23 December 2022; issue dated August 2023.
- Florian Tambon, Gabriel Laberge, Le An, Amin Nikanjam, Paulina Stevia Nouwou Mindom, Yann Pequignot, Foutse Khomh, Giulio Antoniol, Ettore Merlo, and François Laviolette. How to certify machine learning based safety-critical systems? a systematic literature review. *Automated Software Engineering*, 29(2), 2022. doi: 10.1007/s10515-022-00337-x.
- Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2008)*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008. doi: 10.1007/978-3-540-78800-3\_24.
- Luís Cruz-Filipe, Marijn J. H. Heule, Warren A. Hunt, Jr., Matt Kaufmann, and Peter Schneider-Kamp. Efficient certified RAT verification. In *Automated Deduction – CADE 26*, volume 10395 of *Lecture Notes in Computer Science*, pages 220–236. Springer, 2017. doi: 10.1007/978-3-319-63046-5\_14.
- Pamela Zave and Michael Jackson. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6(1):1–30, 1997. doi: 10.1145/237432.237434.
- Dimitra Giannakopoulou, Thomas Pressburger, Anastasia Mavridou, and Johann Schumann. Generation of formal requirements from structured natural language. In *Requirements Engineering: Foundation for Software Quality (REFSQ 2020)*, volume 12045 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2020. doi: 10.1007/978-3-030-44429-7\_2.
- Orlena C. Z. Gotel and Anthony C. W. Finkelstein. An analysis of the requirements traceability problem. In *Proceedings of the IEEE International Conference on Requirements Engineering (ICRE ’94)*, pages 94–101, Colorado Springs, CO, USA, 1994. IEEE Computer Society Press. doi: 10.1109/ICRE.1994.292398.
- C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1): 41–46, 1970. doi: 10.1109/TIT.1970.1054406.
- Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010. URL <https://jmlr.org/papers/v11/el-yaniv10a.html>.
- Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 4878–4887. Curran Associates, 2017.

- Stephen Bates, Anastasios N. Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021. doi: 10.1145/3478535.
- Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.
- C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. doi: 10.1093/biomet/26.4.404.
- James A. Hanley and Abby Lippman-Hand. If nothing goes wrong, is everything all right? interpreting zero numerators. *JAMA*, 249(13):1743–1745, 1983. doi: 10.1001/jama.1983.03330370053031.
- Elliot Myunghoon Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267 of *Proceedings of Machine Learning Research*, pages 30038–30066. PMLR, 2025. URL <https://proceedings.mlr.press/v267/kim25e.html>.
- Shashwat Goel, Joschka Strüder, Ilze Amanda Auzina, Karuna K. Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. Great models think alike and this undermines AI oversight. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267 of *Proceedings of Machine Learning Research*, pages 19621–19678. PMLR, 2025. URL <https://proceedings.mlr.press/v267/goel25b.html>.
- Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2): 230–253, 1997. doi: 10.1518/001872097778543886.
- Raja Parasuraman and Dietrich H. Manzey. Complacency and bias in human use of automation: An attentional integration. *Human Factors*, 52(3):381–410, 2010. doi: 10.1177/0018720810376055.
- Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel S. Weld. Does the whole exceed its parts? the effect of AI explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*, pages 1–16, Yokohama, Japan, 2021. ACM. doi: 10.1145/3411764.3445717.
- RTCA. DO-330: Software tool qualification considerations. RTCA, Inc., Washington, DC, December 2011. Issued 13 December 2011 (SC-205). Equivalent document: EUROCAE ED-215, Software Tool Qualification Considerations, January 2012 (WG-71; Corrigendum 1, February 2021).
- Federal Aviation Administration. Airborne software development assurance using EUROCAE ED-12( ) and RTCA DO-178( ). Advisory Circular AC 20-115D, U.S. Department of Transportation, Federal Aviation Administration, July 2017. URL [https://www.faa.gov/documentLibrary/media/Advisory\\_Circular/AC\\_20-115D.pdf](https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-115D.pdf). Issued 21 July 2017; status Active (checked 2 October 2026).
- ASME. V&V 10-2019: Standard for verification and validation in computational solid mechanics. American Society of Mechanical Engineers, New York, 2019. Published 2020; reaffirmed 2025 (R2025).
- U.S. Department of Defense. MIL-HDBK-5J: Metallic materials and elements for aerospace vehicle structures. Department of Defense Handbook, Distribution Statement A, January 2003. URL [https://quicksearch.dla.mil/qsDocDetails.aspx?ident\\_number=53876](https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=53876). Dated 31 January 2003; canceled (Notice 2, 24 March 2006) with DOT/FAA/AR-MMPDS named as replacement.

## Appendix A. Proof and derivational details for Proposition 1

Notation, the setting (S1)–(S4), fallback-performance equivalence and the quantities  $d_{gate}$ ,  $d_M$ ,  $\eta$ ,  $\gamma$ ,  $q_M(E)$  and  $\omega$  are as defined in Section 3.9.

### A.1 Proof of Proposition 1

*Proof.* Recall the over-restriction residual  $\omega = P(\neg\hat{S}(i, o) \wedge R(i, o) \mid E)$  of Section 3.9.

1. By (S4) and the law of total probability,

$$d_{gate} = P(R(i, o) \wedge E) + P(R(i, F(i)) \wedge \neg E), \quad d_M = P(R(i, M(i)) \wedge E) + P(R(i, M(i)) \wedge \neg E).$$

2. By fallback-performance equivalence the second terms coincide, so

$$d_{gate} - d_M = P(E)[P(R(i, o) \mid E) - P(R(i, M(i)) \mid E)].$$

3. The second conditional probability is  $1 - q_M(E)$ .

4. For the first, partition  $\neg R(i, o)$  according to  $\hat{S}(i, o)$ :

$$P(\neg R \mid E) = P(\hat{S} \wedge \neg R \mid E) + P(\neg\hat{S} \wedge \neg R \mid E) = \gamma + (\eta - \omega),$$

because  $\eta = P(\neg\hat{S} \wedge R \mid E) + P(\neg\hat{S} \wedge \neg R \mid E) = \omega + P(\neg\hat{S} \wedge \neg R \mid E)$ . Hence  $P(R(i, o) \mid E) = 1 - \eta - \gamma + \omega$ .

5. Substituting gives the exact identity

$$d_{gate} - d_M = P(E)[q_M(E) - \eta - \gamma + \omega].$$

6. Since  $\omega \geq 0$  and  $P(E) > 0$ , (3) follows.  $\square$

### A.2 Remark on the over-restriction residual

The over-restriction residual  $\omega$  is required for the exact decomposition (4). It is not needed for (3), or for the sufficient conditions for non-degradation derived from (3).

### A.3 Corollary 1 (Type A): further detail

Under  $\gamma = 0$ , the gain is strictly positive exactly when the incumbent errs with positive probability on instances the gate admits. Its size is limited by the admission probability, so the practical benefit depends on how often the generator produces artefacts that the gate admits.

### A.4 Corollary 2 (Type B): further detail

If  $\alpha + \gamma > q_M(E)$ , the bracket is negative, and the coverage bound does not yield a lower bound in that direction.

### A.5 Corollary 3 (Types C, D, E): derivational detail

- **Types D and E, and C[assist]:** no AI artefact is admitted, so  $P(E) = 0$  and (S3) fails. Under Types D and E the workflow unit is executed without AI; under C[assist] AI material is advisory only.
- **C[adjudicate]:**  $R$  is often not bivalent, so (S1) fails. Where it is bivalent, (3) and (4) remain algebraically valid. However, no record clause bounds  $\eta$  for a human gate, and fallback-performance equivalence cannot be presumed, because the person who adjudicates, and who may then produce the deliverable, has seen the AI material. Any comparison would rest entirely on empirical human-factors evidence, outside the framework's guarantees.

## Supplementary Material

**Nature of the case study.** This is an author-led research case study showing how the framework is instantiated and applied in an executed mixed engineering workflow: decomposition into workflow units, unit-by-unit classification of AI participation, operation of the regimes, the points at which engineering judgement remains necessary, and routing of insufficient evidence to an ADJUDICATE disposition rather than PASS. It is not a certification, airworthiness or independent assurance exercise; its scope limits are those stated in Section 6.2.

**Responsibility.** Puxue Tan was the study lead, domain engineer and author reviewer, and retained responsibility for the engineering judgements and approvals required by the executed study. Deterministic steps relied on their defined mechanisms, and their recorded tool outputs are the evidence. Human review was used where the assigned regime required engineering judgement (S6).

Except for the explicitly identified later re-hash of 2 October 2026 (S1.5), the reported study values are taken from the frozen execution records; no engineering analysis was rerun for the Supplement.

## S1 Executed workflow and study setup

### S1.1 Engineering problem

The object is a tapered, stiffened, cantilevered aluminium wing-spar segment (span  $L = 2.4$  m) with four web lightening holes and three one-sided stiffeners. It is idealized as a mid-surface shell model with an 18 mm flange and an 8 mm web and stiffeners, made of 7075-T651. The load is a tip load in positive and negative senses at limit and ultimate levels, with load factors 1.0, 1.5,  $-0.4$  and  $-0.6$  relative to the positive limit load.

The analyses are:

- linear static analysis;
- two-sense linear buckling;
- one geometrically nonlinear (NLGEOM) cross-check at positive ultimate load.

They are evaluated against project-defined stress and buckling criteria (S5).

### S1.2 Workflow decomposition and regime allocation

Procedure 1 decomposed the workflow into **17 workflow units**; the workflow as a whole receives no regime.

| Regime         | Units                                                                                                                                                                      | Count |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Type A         | CAD program generation                                                                                                                                                     | 1     |
| Type D         | Meshing; deck generation and checking; finite-element (FE) solution; extraction and numerical checks; evaluation of approved criteria; engineering report assembly         | 6     |
| C[adjudicate]  | Geometry specification; residual CAD adequacy; formalization of acceptance criteria; case-study engineering report narrative                                               | 4     |
| C[assist]      | Discretization adequacy; load-case and idealization decisions; analysis adequacy; result interpretation; response to unresolved criteria (conditional); study-level review | 6     |
| Type B, Type E | —                                                                                                                                                                          | 0     |

All 17 participation and assurance records were **READY before execution** (S2). These counts refer only to the controlled executed set; the 22-item retrospective audit in S7 is a separate denominator.

### S1.3 Architecture of the executed mixed workflow

The executed workflow combined AI, deterministic tools and engineering judgement as follows (Figure 5.1):

- **AI was used selectively.** It produced the CAD program (Type A). In C[adjudicate] units, AI-produced drafts were subject to acceptance, revision or rejection by the study author. In C[assist] units, AI supplied labelled advisory material, evidence or proposed findings, and the study author produced the judgement.

- **The AI-generated CAD artefact entered only through deterministic Type A assurance:** a geometric gate with a bounded fallback (S3).
- **The deterministic computational chain itself ran automatically through established tools (S4–S5):**
  - scripted meshing at three refinement levels;
  - deck generation with an independent deck checker;
  - a pinned FE solver;
  - deterministic extraction with numerical-integrity checks;
  - deterministic evaluation of the approved criteria.

Engineers did not manually repeat or re-perform this deterministic computation.
- **Separate human-judgement units remained at the predefined adequacy, interpretation and adjudication points (S2, S6), where the framework assigns C[adjudicate] or C[assist], including:**
  - residual CAD adequacy;
  - formalization of the acceptance criteria;
  - load-case and idealization decisions;
  - discretization and analysis adequacy;
  - result interpretation;
  - the response to unresolved criteria.
- **Unresolved evidence was escalated, not converted into PASS.** Four stress clauses and the Q-5 cross-check were dispositioned ADJUDICATE (S5.5).

The executed workflow thus combined an automatically executed deterministic chain with explicit control by the study author at judgement points. This is an initial demonstration of the framework’s structure, not an industrial productivity benchmark. No automation percentage, time saving or optimality is claimed.

#### S1.4 AI generation context (Type A unit)

| Item                 | Value                                                                                                                                                                                                                         |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Provider / interface | Anthropic model-invocation interface                                                                                                                                                                                          |
| Model                | Claude Opus 5.5 (c1aude-opus-5-5); the interface does not echo a runtime model identifier                                                                                                                                     |
| Context              | Fresh context, no conversation history; no access to the reference program, the verifier or downstream results                                                                                                                |
| Prompt               | Frozen generation prompt, sent verbatim (prompt sha256 779754cf . . . 21ad6850f0), with a text-only instruction envelope                                                                                                      |
| Reasoning effort     | The plan specified “High”; the interface did not allow this to be enforced. <b>The run is therefore not evidence of generator performance at a specified reasoning-effort setting.</b> The deterministic gate was unaffected. |
| Output               | One candidate program (sha256 1a69e827 . . . ff72fa8f17), generated in 39.7 s                                                                                                                                                 |

#### S1.5 Frozen study setup and reproducibility

Before generation, a hash manifest of **61 entries** pinned the following, together with the CalculiX 2.23 solver binary (sha256 2e9c75bb . . . 598bb8ee); manifest content hash f55b65c8 . . . 49bb6173e:

- the specifications: geometry, geometric formalization, load specification and material data;
- the acceptance criteria;
- the analysis plans and metric definitions;
- the gate, mesher, FE tools and evaluator;
- the environment locks.

The executed study is defined by these frozen versions and hashes.

**Post-run verification.** The contemporaneous final report records a 61/61 post-run match, but a standalone machine-readable output of that check was not preserved. A later re-hash on 2 October 2026 reproduced the 61/61 result and

the solver-binary hash. This later observation is reported separately and does not replace the original execution-time evidence.

The solver analyses were executed once and not independently repeated. The evidence establishes repeatability of the implemented process, not physical validity of the results or formal tool qualification.

## S2 Participation and assurance allocation (17 workflow units)

**Readiness.** All 17 records were READY before the run: every obligation in the record was satisfied with evidence of an adequate kind. Readiness is a status, not a result. The unit outcomes are reported in S5.6.

**Table S2.1 — Allocation**

The AI-role column states the AI contribution recorded in each unit’s frozen review or provenance record; “None” means that no AI-produced material entered the unit’s execution.

| #   | Workflow unit                                 | AI role in this execution                                  | Regime        |
|-----|-----------------------------------------------|------------------------------------------------------------|---------------|
| T1  | Geometry specification                        | AI-originated baseline values; adopted by the study author | C[adjudicate] |
| T2a | CAD program generation                        | AI generated the program                                   | A             |
| T2b | Residual CAD adequacy                         | AI-generated geometry under review; AI compiled evidence   | C[adjudicate] |
| T3a | Meshing                                       | None                                                       | D             |
| T3b | Discretization adequacy                       | Advisory: AI compiled evidence and proposed findings       | C[assist]     |
| T4a | Deck generation and checking                  | None                                                       | D             |
| T4b | Load cases and idealization                   | Advisory: AI compiled evidence                             | C[assist]     |
| T5a | FE solution                                   | None                                                       | D             |
| T5b | Analysis adequacy                             | Advisory: AI compiled evidence and proposed findings       | C[assist]     |
| T6a | Extraction and numerical checks               | None                                                       | D             |
| T6b | Result interpretation                         | Advisory: AI compiled evidence and proposed findings       | C[assist]     |
| T7a | Formalization of acceptance criteria          | AI-assisted draft of the criteria                          | C[adjudicate] |
| T7b | Evaluation of approved criteria               | None                                                       | D             |
| T8a | Engineering report assembly (numbers)         | None                                                       | D             |
| T8b | Case-study engineering report narrative       | AI drafted the narrative                                   | C[adjudicate] |
| T9  | Response to unresolved criteria (conditional) | Advisory: AI compiled evidence and proposed findings       | C[assist]     |
| T10 | Study-level review                            | Advisory: AI compiled evidence and proposed findings       | C[assist]     |

**Table S2.2 — Mechanism and fallback**

| #   | Mechanism                                                                    | Fallback (where relevant)                                                                                              | Readiness |
|-----|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-----------|
| T1  | Study author adopts the specification                                        | Author re-specifies geometry                                                                                           | READY     |
| T2a | Deterministic geometric gate (S3)                                            | $\leq 3$ attempts via bounded repair loop $\rightarrow$ engineer-authored CAD through the same gate $\rightarrow$ halt | READY     |
| T2b | Study author accepts or rejects                                              | Reject $\rightarrow$ T2a fallback path                                                                                 | READY     |
| T3a | Scripted mesher: input lineage check, frozen quality rules, conformity check | Halt                                                                                                                   | READY     |
| T3b | Study author judges adequacy                                                 | Further refinement, if decided                                                                                         | READY     |
| T4a | Deterministic deck writer + independent deck checker                         | Halt                                                                                                                   | READY     |
| T4b | Study author decides before analysis                                         | —                                                                                                                      | READY     |
| T5a | Pinned solver within a declared validity domain                              | Halt                                                                                                                   | READY     |
| T5b | Study author judges adequacy                                                 | Report limitation                                                                                                      | READY     |
| T6a | Deterministic parsers; equilibrium and consistency checks                    | Halt                                                                                                                   | READY     |
| T6b | Study author interprets                                                      | —                                                                                                                      | READY     |
| T7a | Study author approves and versions the criteria before any result            | Revise as a new version                                                                                                | READY     |
| T7b | Deterministic evaluator (accepts only approved criteria and material data)   | Halt                                                                                                                   | READY     |
| T8a | Report assembler with numeric-token checker                                  | Halt                                                                                                                   | READY     |
| T8b | Study author accepts or revises                                              | Revise report narrative                                                                                                | READY     |
| T9  | Study author decides the response                                            | —                                                                                                                      | READY     |
| T10 | Study author reviews the study package                                       | —                                                                                                                      | READY     |

**Scope limits of the records:**

1. The study is author-led, not independent.
2. Verifier–generator independence is not claimed: generation and acceptance were procedurally separated only (Section 5.3), and gate soundness rests on the engineering evidence in S3.3.
3. OS-level isolation of the candidate code was not demonstrated; in-process containment monitoring served as defence in depth only.
4. The tip-link has no demonstrated finite-rotation validity domain, so Q-5 was predeclared ADJUDICATE (S5.4).

### S3 Type A CAD generation and deterministic gate

#### S3.1 Approved geometric formalization

The formalization has **17 clauses**:

| Clause | Family     | Condition (abridged)                                                                                        |
|--------|------------|-------------------------------------------------------------------------------------------------------------|
| E-1    | Execution  | Exit code 0 within a 120 s budget                                                                           |
| E-2    | Execution  | Exactly one STEP file, with the declared name                                                               |
| U-1    | Units      | STEP declares metre as its single length unit                                                               |
| U-2    | Units      | After calibrated metre import, X span = $2.400 \text{ m} \pm 1 \times 10^{-6} \text{ m}$                    |
| V-1    | Validity   | Fresh-process re-import yields a non-empty shape                                                            |
| V-2    | Validity   | BRep validity of the compound and every face                                                                |
| T-1    | Topology   | Exactly 6 faces, all planar                                                                                 |
| T-2    | Topology   | 2 flange, 1 web and 3 stiffener faces, classified by normal; none unclassified                              |
| T-3    | Topology   | Web has 4 circular inner wires; other faces have none                                                       |
| G-1    | Geometry   | Flange planes at $Z = 0.009$ and $0.411 \text{ m}$                                                          |
| G-2    | Geometry   | Flange outlines: $(0, \pm 0.100)$ , $(2.4, \pm 0.060)$ ; straight edges                                     |
| G-3    | Geometry   | Web outline $(0, 0.009) - (2.4, 0.411)$                                                                     |
| G-4    | Geometry   | Hole centres; radius $0.060 \text{ m}$ ; axes $\parallel Y$                                                 |
| G-5    | Geometry   | Stiffeners at $X = 0.6, 1.2, 1.8 \text{ m}$ ; outline $\{0, 0.050\} \times \{0.009, 0.411\}$ ; one-sided +Y |
| A-1    | Area       | Face areas and total area                                                                                   |
| B-1    | Global     | Bounding box X $[0, 2.4]$ , Y $[-0.1, 0.1]$ , Z $[0.009, 0.411] \text{ m}$                                  |
| P-1    | Provenance | Hashes of program, STEP files and formalization recorded                                                    |

#### Tolerances:

- counts and classification: exact;
- coordinates and radii:  $1 \times 10^{-6} \text{ m}$ ;
- parallelism:  $|\cos| \geq 0.999999999$ ;
- areas:  $1 \times 10^{-6}$  relative.

The formalization covers specified geometric properties only. Residual adequacy is a separate C[adjudicate] unit (S6.1). The study author approved the formalization before generation.

#### S3.2 Gate checks in the admitted run

The gate executes the candidate under a fixed time budget with in-process containment monitoring, requires one metre-unit STEP file, re-imports it in a fresh process and measures **23 checks**. All 23 passed:

| Clause        | Checks                                                | Tolerance                     | Result |
|---------------|-------------------------------------------------------|-------------------------------|--------|
| U-1, V-1, V-2 | 3 (unit declaration, non-empty import, BRep validity) | exact                         | pass   |
| T-1, T-2, T-3 | 3 (faces, classification, inner wires)                | exact                         | pass   |
| G-1, G-2, G-3 | 4 (flange planes, two flange outlines, web outline)   | $1\text{e-}6 \text{ m}$       | pass   |
| G-4           | 3 (hole centres, radii, axes)                         | $1\text{e-}6 \text{ m}$ ; dot | pass   |
| G-5           | 4 (stiffener planes, three outlines)                  | $1\text{e-}6 \text{ m}$       | pass   |
| A-1           | 4 (flange, web, stiffener, total areas)               | $1\text{e-}6 \text{ rel}$     | pass   |
| B-1, U-2      | 2 (bounding box, span sentinel)                       | $1\text{e-}6 \text{ m}$       | pass   |

The remaining clauses were also satisfied:

- **E-1:** candidate run time 39.2 s within the 120 s budget;

- **E-2:** one STEP file;
- **P-1:** hashes recorded.

No containment violation was detected. Decision: **ADMIT**.

### S3.3 Verifier evidence

The gate’s soundness is supported by engineering evidence, not by formal proof. A **32-case** test suite ran before the study, and all stated expectations were met:

- **25 cases with stated expectations:**
  - 4 **ADMIT**: reference program, alternative construction route, two determinism runs;
  - 18 **REJECT**: 15 geometric mutations and 3 containment probes;
  - 3 **GENERATION\_FAILED**: exception, time-out, two STEP files;
- **7 record-only cases.**

The suite was complemented by **19 logic-level tests** and a review of the gate code by the study author.

**Table S3.3 — Geometric mutations of the engineer-adopted reference program (all REJECT)**

| Mutation                                 | Clauses failed                |
|------------------------------------------|-------------------------------|
| Hole shifted +1 mm in X                  | G-4                           |
| Hole radius 0.059 m                      | A-1, G-4                      |
| Hole deleted                             | A-1, G-4, T-3                 |
| Stiffener at X = 1.21 m                  | G-5                           |
| Stiffener deleted                        | A-1, G-5, T-1, T-2            |
| Stiffener on –Y side                     | G-5                           |
| Tip flange width 0.130 m                 | A-1, G-2                      |
| Kinked flange, same endpoints            | A-1, G-2                      |
| Web extended to Z = 0.420 m              | A-1, B-1, G-3                 |
| Duplicated stiffener face                | A-1, G-5, T-1, T-2            |
| STEP truncated after export              | U-2, V-1                      |
| Correct bounding box printed, hole moved | G-4                           |
| Geometry scaled $\times 1000$            | A-1, B-1, G-1...G-5, U-2      |
| Thickened solid web                      | A-1, G-1...G-5, T-1...T-3     |
| Millimetre export of metre geometry      | A-1, B-1, G-1...G-5, U-1, U-2 |

The containment probes (network access, writing outside the run directory, spawning a child process) were all rejected.

### S3.4 Bounded repair loop and fallback (predeclared)

1. At most three sequential AI attempts.
2. Attempts 2–3 receive only the frozen prompt plus the machine-readable failed clauses.
3. After three rejections, an engineer-authored CAD program must pass the same gate.
4. Failing that, the unit halts and the halt is recorded.

### S3.5 Outcome

The first candidate was **admitted at the first attempt, with 23/23 checks passed**, and was **used unmodified**. The repair loop and the fallback were not used.

*Run-order note.* An earlier gate invocation failed before the candidate executed, because of an operator-side path error. It is retained as an invalid run; it did not alter the candidate and consumed no generation attempt.

## S4 Deterministic downstream workflow

After admission, no further AI-generated engineering artefact entered the computational chain. Each step below is Type D, and its recorded tool output is the evidence.

### S4.1 Meshing and refinement

The mesher was a scripted Gmsh 4.15.2 driver. It meshed only the gate-admitted geometry, verified by a lineage check, using homogeneous quadratic triangular shell elements (CalculiX S6 elements) at three refinement levels.

| Level  | Elements | Quality rules<br>R1–R10 | Geometry<br>conformity | min SJ | max aspect |
|--------|----------|-------------------------|------------------------|--------|------------|
| Coarse | 3,218    | all pass                | pass                   | 0.775  | 2.22       |
| Medium | 12,152   | all pass                | pass                   | 0.881  | 1.66       |
| Fine   | 47,046   | all pass                | pass                   | 0.942  | 1.68       |

min SJ: minimum scaled Jacobian; R1–R10: the ten frozen project mesh-quality rules.

Geometry conformity was checked in the CAD kernel, independently of the mesher; the maximum distance was  $8.9 \times 10^{-16}$  m. Repeated meshing on the same host gave identical canonical hashes at every level.

### S4.2 Solver input decks

A deterministic deck writer produced **21 decks**:

- 12 linear static: 3 levels  $\times$  4 load cases;
- 6 linear buckling: 3 levels  $\times$  2 senses;
- 2 doubled-load buckling scaling checks (fine mesh);
- 1 NLGEOM run (fine mesh, positive ultimate load).

An independent deterministic checker verified every deck (33–35 checks each). Among other things, it reconstructs the rigid tip load-introduction constraints and confirms that loads and supports are complete. **Result: 21/21 decks passed, and every deck regenerated byte-identically.** The solver ran only after all decks had passed.

### S4.3 Solver

CalculiX 2.23 was run as a hash-pinned binary. **Result: 21/21 jobs completed, with no errors or warnings recorded.** The tool path is supported by the solver’s official test suite run on this binary and by project fixtures for bending, buckling and the tip link.

### S4.4 Extraction and numerical integrity

Results were extracted deterministically from the solver’s printed output. Equilibrium is checked in a precision-aware way: each printed value is treated as an interval set by its printed precision. The tolerances are  $1 \times 10^{-6}$  for force and  $1 \times 10^{-4}$  for moment, relative to the applied load.

| Jobs                                                   | Force equilibrium                                  | Moment equilibrium                                                | Output consistency       |
|--------------------------------------------------------|----------------------------------------------------|-------------------------------------------------------------------|--------------------------|
| 12 linear static jobs (3 levels $\times$ 4 load cases) | 12/12 accepted (error $\leq 5.7 \times 10^{-12}$ ) | 12/12 accepted (error $\leq 6.3 \times 10^{-7}$ )                 | all conditions satisfied |
| NLGEOM (fine, positive ultimate)                       | accepted ( $2.5 \times 10^{-13}$ )                 | <b>not met:</b> $2.01 \times 10^{-4}$ ; diagnostic only, see S5.4 | all conditions satisfied |

The fine NLGEOM output files (over 1 GB) were too large for the planned whole-file parsers. The unchanged parsers were applied to byte-exact excerpts of the final increment. On a fixture, the excerpt parse was shown to equal the full-file parse.

### S4.5 Forward-carrying controls

Three controls carry the admission decision through the chain:

- the mesher accepts only the gate-admitted geometry;
- the solver runs only after every deck passes the independent checker;
- the evaluator accepts only the approved criteria and material data.

These controls are independent of the geometry's source by design, although the single run did not test this.

## S5 Engineering criteria, convergence and dispositions

### S5.1 Frozen criteria and material data

The criteria were approved by the study author and frozen before any result existed. They are project-defined comparison criteria; they are **not** a CS-25 compliance finding.

| Clause      | Load level                                 | Criterion                                                                                                   |
|-------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Q-1+ / Q-1– | Limit, positive (1.0) / negative (–0.4)    | Converged $\sigma_{proj,g} \leq F_{ty,g}$ for each group g                                                  |
| Q-2+ / Q-2– | Ultimate, positive (1.5) / negative (–0.6) | Converged $\sigma_{proj,g} \leq F_{ty,g}$ (elastic-through-ultimate project criterion)                      |
| Q-3+ / Q-3– | Buckling, positive / negative (fine mesh)  | Converged smallest positive buckling factor $\lambda \geq 1.5$                                              |
| Q-4         | All levels                                 | Tip deflection and $\delta/L$ reported; no threshold                                                        |
| Q-5         | $1.5 \times$ positive limit load (fine)    | Linear vs NLGEOM relative difference $\leq 0.05$ ; escalation if $\delta/L > 0.01$ or $ \theta  > 0.05$ rad |

**Stress measure.** Von Mises stress is computed from unaveraged integration-point stresses extrapolated to the element corners on both surfaces. The group maximum (rule S1 of the frozen engineering plans) is the largest such value in a component group.  $\sigma_{proj}$  (rule S3 of the same plans) is taken outside a predeclared root exclusion zone ( $x_{ex} = 0.20$  m) and is bounded below by the nominal beam-theory reference.

**Allowables.** MIL-HDBK-5J (2003), Table 3.7.6.0(b1), 7075-T651 plate, A-basis. For the 18 mm flange (0.709 in; 0.500–1.000 in thickness range), the minimum of the L and LT  $F_{ty}$  values is 68 ksi  $\approx$  **469 MPa**. For the 8 mm web and stiffeners (0.315 in; 0.250–0.499 in range), the minimum is 67 ksi  $\approx$  **462 MPa**.  $F_{tu}$  is context only.

### S5.2 Convergence rule

Convergence of each quantity  $q$  is judged by its relative medium-to-fine change

$$c(q) = \frac{|q_{fine} - q_{medium}|}{\max(|q_{fine}|, q_{floor})},$$

where  $q_{floor}$  is a frozen, quantity-specific floor that prevents division by near-zero values:  $10^{-6}$  m for displacement,  $10^6$  Pa for stress and  $10^{-3}$  for buckling factors. In every reported case  $|q_{fine}|$  exceeds its floor (for stresses by a factor of at least about 30; for displacements and buckling factors by several orders of magnitude), so the floor did not affect any reported value.

Thresholds were frozen before the run:

- tip displacement:  $c \leq 0.01$ ;
- buckling factors:  $c \leq 0.01$ ;
- stress metrics:  $c \leq 0.05$ .

A quantity that fails is **NOT\_CONVERGED**, and the threshold is never relaxed.

*Worked example (flange  $\sigma_{proj}$ , positive ultimate):*  $c = |246.54 - 182.60| / 246.54 = 0.259 > 0.05$ , so **NOT\_CONVERGED**.

### S5.3 Results

The relative changes agree across load cases to within rounding (tip displacement 9.1–9.2e-5; stresses identical to the precision shown), consistent with the linear analysis.

**Table S5.3a — Tip displacement (mm)**

| Load case         | Coarse   | Medium   | Fine     | c      | Status    |
|-------------------|----------|----------|----------|--------|-----------|
| Limit + (1.0)     | 20.8968  | 20.9006  | 20.9025  | 9.1e-5 | CONVERGED |
| Ultimate + (1.5)  | 31.3452  | 31.3510  | 31.3538  | 9.1e-5 | CONVERGED |
| Limit – (–0.4)    | –8.3587  | –8.3603  | –8.3610  | 9.1e-5 | CONVERGED |
| Ultimate – (–0.6) | –12.5381 | –12.5404 | –12.5415 | 9.2e-5 | CONVERGED |

**Table S5.3b — Smallest positive buckling factor**

| Sense    | Coarse   | Medium   | Fine     | c      | Status    |
|----------|----------|----------|----------|--------|-----------|
| Positive | 4.852521 | 4.839756 | 4.834881 | 1.0e-3 | CONVERGED |
| Negative | 12.1364  | 12.09921 | 12.08619 | 1.1e-3 | CONVERGED |

**Table S5.3c — Flange stress (MPa)**

The raw maximum lies at the clamped root corner and is reported separately as a non-converged local peak; it is not removed.

| Load case  | Quantity        | Coarse | Medium | Fine  | c     | Status              |
|------------|-----------------|--------|--------|-------|-------|---------------------|
| Limit +    | raw maximum     | 170.1  | 199.5  | 226.8 | 0.120 | reported separately |
| Limit +    | $\sigma_{proj}$ | —      | 121.7  | 164.4 | 0.259 | NOT_CONVERGED       |
| Ultimate + | raw maximum     | 255.1  | 299.3  | 340.2 | 0.120 | reported separately |
| Ultimate + | $\sigma_{proj}$ | —      | 182.6  | 246.5 | 0.259 | NOT_CONVERGED       |
| Limit –    | raw maximum     | 68.0   | 79.8   | 90.7  | 0.120 | reported separately |
| Limit –    | $\sigma_{proj}$ | —      | 48.7   | 65.7  | 0.259 | NOT_CONVERGED       |
| Ultimate – | raw maximum     | 102.0  | 119.7  | 136.1 | 0.120 | reported separately |
| Ultimate – | $\sigma_{proj}$ | —      | 73.0   | 98.6  | 0.259 | NOT_CONVERGED       |

**Table S5.3d — Web and stiffener group-maximum stress (rule S1; MPa)**

| Load case  | Web (coarse / medium / fine) | Web c | Stiffener (coarse / medium / fine) | Stiffener c |
|------------|------------------------------|-------|------------------------------------|-------------|
| Limit +    | 119.1 / 118.4 / 122.6        | 0.035 | 51.8 / 65.6 / 78.5                 | 0.164       |
| Ultimate + | 178.7 / 177.5 / 183.9        | 0.035 | 77.6 / 98.4 / 117.8                | 0.164       |
| Limit –    | 47.7 / 47.3 / 49.1           | 0.035 | 20.7 / 26.2 / 31.4                 | 0.164       |
| Ultimate – | 71.5 / 71.0 / 73.6           | 0.035 | 31.1 / 39.4 / 47.1                 | 0.164       |

Web: CONVERGED. Stiffener: NOT\_CONVERGED; its peak lies at a stiffener–flange junction, and junctions are included in the metric by design.

**Nominal root references:** flange 114.2 / 171.3 / 45.7 / 68.5 MPa, and web 34.1 / 51.2 / 13.6 / 20.5 MPa (Limit + / Ultimate + / Limit – / Ultimate –).

## S5.4 Clause verdicts

**Table S5.4a — Stress clauses**

The ratio  $\sigma_{proj} / F_{ty}$  is shown for information only. A non-converged metric cannot yield PASS.

| Clause | Flange | Web   | Stiffener | Evaluator verdict |
|--------|--------|-------|-----------|-------------------|
| Q-1+   | 0.350  | 0.265 | 0.170     | NOT_CONVERGED     |
| Q-1−   | 0.140  | 0.106 | 0.068     | NOT_CONVERGED     |
| Q-2+   | 0.526  | 0.398 | 0.255     | NOT_CONVERGED     |
| Q-2−   | 0.210  | 0.159 | 0.102     | NOT_CONVERGED     |

**Table S5.4b — Other clauses**

| Clause | Values                                                                                   | Evaluator verdict |
|--------|------------------------------------------------------------------------------------------|-------------------|
| Q-3+   | $\lambda = 4.834881 \geq 1.5$ ; run complete; converged                                  | PASS              |
| Q-3−   | $\lambda = 12.08619 \geq 1.5$ ; run complete; converged                                  | PASS              |
| Q-4    | $\delta/L = 0.0087, 0.0131, -0.0035, -0.0052$ (Limit +, Ultimate +, Limit −, Ultimate −) | REPORTED          |
| Q-5    | $r(\delta_{tip}) 2.4e-4$ ; $r(\sigma)$ flange 0.0196, web 0.0075, stiffener $3.4e-4$     | SUPPORTED         |

### Buckling diagnostics (not a verdict condition).

- The doubled-load scaling check raised no anomaly ( $e = 3.2 \times 10^{-7}$  and  $4.9 \times 10^{-7}$ ).
- Each fine-mesh buckling run returned 20 positive factors.

**Q-5.** Before any analysis, Q-5 was declared to require adjudication: the finite-rotation validity of the linearized tip-load relation had not been established. The results support that caution:

- $\delta/L$  at ultimate load = 0.0131, above the 0.01 escalation threshold;
- the NLGEOM run showed a moment-equilibrium deficit ( $2.01 \times 10^{-4}$ ) consistent with the known limitation of the linearized link (S6.3).

## S5.5 Final dispositions

| Clause | Evaluator verdict | Final disposition        |
|--------|-------------------|--------------------------|
| Q-1±   | NOT_CONVERGED     | ADJUDICATE               |
| Q-2±   | NOT_CONVERGED     | ADJUDICATE               |
| Q-3±   | PASS              | PASS                     |
| Q-4    | REPORTED          | Not applicable           |
| Q-5    | SUPPORTED         | ADJUDICATE (predeclared) |

Favourable stress ratios (at most 0.526) did **not** yield PASS, because the required convergence evidence was absent. Those clauses received an ADJUDICATE disposition, and no threshold or criterion was changed.

### S5.6 Unit outcomes: READY is not PASS

Unit outcomes are study-level outcome labels, distinct from clause verdicts, deployment readiness and the regime guarantees. Over the 17 units, all READY before the run:

| Outcome                            | Count | Units                                                     |
|------------------------------------|-------|-----------------------------------------------------------|
| PASS                               | 12    | T1, T2a, T2b, T3a, T4a, T5a, T6a, T6b, T7b, T8a, T8b, T10 |
| ADJUDICATE                         | 3     | T3b, T5b, T9                                              |
| Not applicable (pre-run decisions) | 2     | T4b, T7a                                                  |
| FAIL                               | 0     | —                                                         |

The evaluation unit (T7b) recorded PASS because it executed the approved criteria as specified, even though several of the dispositions it produced were ADJUDICATE.

## S6 Human judgement and additional diagnostics

### S6.1 Engineering judgements made by the study author

| Judgement                             | Regime        | Decision                                                                                                                                                                                                                                                                                   |
|---------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Residual CAD adequacy                 | C[adjudicate] | The study author judged the admitted geometry adequate for the study and accepted it with limitations. Defensive code branches unnecessary in the pinned environment were recorded rather than edited, because the predeclared protocol allowed modification only through the repair loop. |
| Formalization of acceptance criteria  | C[adjudicate] | The study author approved the criteria (S5.1) before any result.                                                                                                                                                                                                                           |
| Discretization adequacy               | C[assist]     | The study author accepted the three-level study with limitations. Tip displacement, buckling factors and the web stress converged; flange $\sigma_{proj}$ and the stiffener did not. No further refinement was made within this study.                                                     |
| Analysis adequacy and Q-5             | C[assist]     | The study author kept Q-5 at ADJUDICATE per the predeclared rule and judged the NLGEOM moment deficit consistent with $\theta^2/2$ (S6.3). It was not used as release-supporting evidence.                                                                                                 |
| Result interpretation                 | C[assist]     | The study author identified the first buckling mode in both senses as a global lateral-torsional mode of the laterally unrestrained cantilever, so the factors depend on the idealization of tip and lateral restraint.                                                                    |
| Response to unresolved stress clauses | C[assist]     | Recorded as a limitation, with no design change. Model refinement was deferred to a new versioned iteration, and no criterion was changed.                                                                                                                                                 |
| Study-level review                    | C[assist]     | The demonstration was accepted with limitations as a research demonstration, not a certification.                                                                                                                                                                                          |

## S6.2 Post-hoc diagnostics (not verdict inputs)

These were computed **after** the verdicts were recorded, at positive ultimate load. They were not passed to the evaluator, changed no verdict, and led to no change in any criterion, formalization, metric, tolerance or exclusion zone.

| Diagnostic                                                   | Medium    | Fine      | c      |
|--------------------------------------------------------------|-----------|-----------|--------|
| Interior flange, $0.20 \text{ m} \leq x \leq 2.20 \text{ m}$ | 179.4 MPa | 179.7 MPa | 0.2 %  |
| Stiffener, excluding elements within 0.05 m of the flanges   | 41.9 MPa  | 48.2 MPa  | 13.0 % |

The flange result suggests that the flange non-convergence is confined to the end regions: the root clamp and the tip load-introduction corner. The stiffener remains unresolved. These diagnostics help explain the outcome; they do not establish compliance.

## S6.3 Finite-rotation behaviour of the tip link

At positive ultimate load the tip rotation is  $\theta \approx 0.0202$  rad. The NLGEOM moment deficit ( $2.01 \times 10^{-4}$ ) matches  $\theta^2/2 = 2.05 \times 10^{-4}$ . This is the deficit of a linearized rigid-link moment under finite rotation, consistent with the predeclared limitation that the link has no demonstrated finite-rotation validity domain.

## S6.4 Formalization limitation observed in execution

The predeclared stress rule represented the root-clamp region but not the tip load-introduction region revealed by the run. This was recorded as a limitation. Any refinement would be a new version of the formalization, fixed before a future rerun, not adopted retrospectively.

# S7 Retrospective framework audit of the original case-study record

Before the controlled execution, the framework was applied retrospectively to the **original** wing-spar case-study record (the record of an earlier execution of the same analysis, from April 2026, made before the controlled protocol was defined), using only the implementation and evidence that actually existed. The retrospective audit identified **22 auditable items** in the original case-study record.

**This 22-item retrospective audit is not interchangeable with the 17-workflow-unit controlled execution (S1–S6).** The decomposition, the evidence base and the execution differ, so the two sets must not be pooled or compared as rates.

**Table S7.1 — Retrospective regime allocation (22 audited items)**

| Activity                      | Audited items and retrospective regime                                                                           |
|-------------------------------|------------------------------------------------------------------------------------------------------------------|
| Geometry definition           | T1a numeric specification: C[adjudicate]; T1b concept and sizing: C[adjudicate]                                  |
| CAD generation                | T2a CAD program generation: C[adjudicate]; T2b residual CAD adequacy: C[adjudicate]                              |
| Meshing                       | T3a mesh generation: C[adjudicate]; T3b discretization adequacy: C[assist]                                       |
| Loads and boundary conditions | T4a card values: C[adjudicate]; T4b load cases and idealization: C[assist]                                       |
| FE solution                   | T5a solver execution: D; T5b analysis adequacy: C[assist]                                                        |
| Result extraction             | T6a eigenvalue/load extraction: C[adjudicate]; T6b peak extraction: C[adjudicate]; T6c interpretation: C[assist] |
| Compliance                    | T7a formalization: C[adjudicate]; T7b evaluation: D                                                              |
| Engineering report            | T8a numerical consistency: C[adjudicate]; T8b report narrative: C[adjudicate]                                    |
| Design modification           | T9a option estimates: C[adjudicate]; T9b decision: C[assist]; T9c criterion change: C[adjudicate]                |
| Final approval                | T10a numeric restatement: C[adjudicate]; T10b final authorization: C[assist]                                     |

**Tally:** C[adjudicate] **14**, C[assist] **6**, D **2**, A **0**, B **0**, E **0**. **Readiness:** **0/22**.

**What the audit shows.** Checks that the original case-study record had presented as formal or statistical verification were, on the surviving evidence, unrecorded human adjudication. The framework classified them accordingly rather than forcing Type A or Type B. Several audited items have a Type A or D **target regime**, for example:

- an implemented geometric verifier for CAD generation;
- a validated parser for result extraction;
- a deterministic numerical cross-check for report numbers.

These targets motivated the controlled execution.

**What NOT READY means here.** NOT READY means that the retrospective evidence package does not satisfy every obligation of the assigned record: identified authorization, AI provenance, approved formalizations or tool validation evidence (where an implemented verifier was missing, this determined the classification and a Type A target regime, not readiness). It does **not** mean that the engineering work was unsafe or wrong, or that a deterministic tool was unreliable. For example, the original solver run completed with machine-precision equilibrium, yet its record is NOT READY because validation evidence and a declared validity domain were not cited.