# The “Last Mile” of PHM: A Verifiable Claim Architecture for Deterministic Airworthiness Compliance Assessment

Simon Trussler, Ralph Poole, Ken Percell, and Frank Zahiri

*Shipcom Federal Solutions, LLC; Houston, TX 77082, USA*
*{strussler, rpoole, kpercell, fzahiri}@shipcom.ai*

## ABSTRACT

To accelerate military airworthiness certification, this paper introduces a deterministic, natural language processing workflow that establishes a verifiable claim architecture (VCA). While PHM strategies detect anomalies and predict subsystem failures, mapping these engineering findings to regulatory mandates remains a documentation bottleneck. The proposed VCA automates evidence discovery by integrating multi-format layout parsing and reproducible Boolean filtering to establish a structured, auditable evidence layer. Large language models are deployed as constrained text-synthesis engines, operating within isolated context windows to populate formal compliance matrices. To eliminate textual hallucination and abstractive drift, a provenance audit layer anchors every generated narrative block to an immutable source document metadata tag. The architecture integrates system safety ontologies to map engineering deviations to standardized risk assessment codes and operational flight restrictions in accordance with MIL-STD-882E. Evaluated by the Air Force Life Cycle Management Center against 7,346 pages of compliance artifacts across multiple aircraft platforms, the framework demonstrated a conservative safety posture, deferring policy exceptions to human engineers. By reducing expert administrative labor hours by an estimated 50% and compressing certification schedules from 18–36 months down to 6–12 months, this domain-agnostic framework streamlines the compliance lifecycle, allowing human experts to focus on high-consequence risk assessment.

## 1. THE “LAST MILE” OF PHM

Prognostics and health management (PHM) has transitioned from traditional time-based maintenance to predictive strategies driven by condition monitoring and fault diagnosis (Fu & Avdelidis, 2023). However, in military and commercial aviation, detecting an anomaly or predicting a subsystem failure addresses only a portion of the operational lifecycle. Aircraft certification is a dual-phase mandate: it requires establishing initial airworthiness to validate the platform's baseline design, and maintaining ongoing airworthiness as the aircraft operates, ages, and undergoes periodic subsystem modifications. Because an aircraft's operational status is governed by these strict safety baselines, predictive insights from PHM must ultimately translate into proof of compliance across both phases. The critical challenge in PHM is therefore certification and compliance: demonstrating that each localized health assessment, subsystem design update, and PHM-driven maintenance action satisfies regulatory mandates (Pavlyuk & Alomar, 2026).

Bridging the gap between low-level PHM diagnostic data and high-level airworthiness criteria creates a substantial documentation burden (Lorenzo & Takahashi, 2024; Onilede, 2026; Pavlyuk & Alomar, 2026; Weiss & Brundage, 2021), particularly when validating engineering changes against initial certification baselines. Critical engineering data, which substantiate asset health and prove ongoing flight safety, are typically embedded across numerous fragmented, multi-format vendor test reports (King, 2021). Consequently, subject matter experts (SMEs) spend excessive time conducting manual text searches and populating compliance matrices to prove a new subsystem design does not invalidate the original safety case. This bottleneck redirects specialized personnel from safety reviews to manual data retrieval (King, 2021), increasing fatigue in a workflow where oversight risks catastrophic failure.

To address this unstructured data problem, the PHM community increasingly adopts natural language processing (NLP) (Löwenmark et al., 2022; Cejas et al., 2023) and large language models (LLMs) to parse technical documentation, ranging from querying unstructured inspection reports to retrieving maintenance evidence from Airworthiness Directives. While these generative language models offer strong contextual summarization capabilities, standard commercial configurations introduce severe risks to safety-critical health management. This risk stems from

hallucination, a phenomenon where a model generates fluent, syntactically correct text that is factually incorrect or ungrounded in source documentation (Alansari & Luqman, 2026; Trilla et al., 2022).

In airworthiness certification, where falsely verifying a degraded component or an untested design modification could authorize a compromised aircraft for flight, probabilistic estimation of factual compliance is unacceptable. Unconstrained generative outputs risk injecting unverified engineering conclusions into formal compliance reports. To deploy generative models safely within a military airworthiness framework, the model must function strictly as a deterministic text-synthesis engine bounded by explicit programmatic constraints.

To address data fragmentation and establish trust in automated PHM workflows, this paper introduces a deterministic, NLP-driven pipeline that optimizes the final stages of the airworthiness compliance lifecycle. We present a verifiable claim architecture (VCA) that automates evidence discovery and maps high-level regulatory mandates directly to fragmented PHM and engineering data.

The architecture programmatically extracts technical evidence while structurally preventing data hallucination. It achieves this by combining structural layout parsing and deterministic Boolean filtering with low-temperature LLMs restricted to isolated context windows. Using KC-46 Pegasus compliance reporting against MIL-HDBK-516C Section 13 (Electromagnetic Environmental Effects or E3) as a primary operational use case, this paper demonstrates how unstructured document corpora are converted into auditable airworthiness artifacts. This framework deploys artificial intelligence (AI) strictly as an operational accelerator, preserving authoritative decision-making for human engineers and ensuring that the final stage of PHM compliance remains verifiable.

## 2. RELATED WORK

Automated compliance verification and requirement evaluation have transitioned from heuristic, rule-based matching to neuro-symbolic and verifiable language model architectures. While early engineering document processing relied on regular expressions, manual parsing, and keyword matching to identify standardized criteria, these approaches fail to generalize across dense, multi-page technical reports. Foundation models resolve key semantic extraction bottlenecks, yet they frequently hallucinate and lack native formal verification capabilities (Dhuliawala et al., 2023). Because unverified autoregressive reasoning degrades over long reasoning chains (Zhang, Y. et al., 2024), reliable compliance determination requires models to ground their findings in factual, fine-grained textual evidence (Gao et al., 2023; Zhang, J. et al., 2024) and couple neural generation with external verification mechanisms (Pan et al., 2023; Patlakas et al., 2024). Furthermore, extracting empirical evidence from visually complex engineering artifacts necessitates specialized document conversion pipelines, such as Docling (Livathinos et al., 2025; Nassar et al., 2025), which preserve tabular structure and layout fidelity before passing structured Markdown or JSON to the downstream reasoning engine.

### Verifiable Language Model Architectures

Foundation models frequently hallucinate and lack systematic formal verification capabilities (Dhuliawala et al., 2023). To mitigate these vulnerabilities, researchers integrate neural models with formal proof assistants (Lin et al., 2024). Architectures such as Seed-Prover use iterative refinement and binary feedback from the Lean 4 compiler to improve automated theorem proving (Chen et al., 2025). DeepSeek-Prover-V2 applies reinforcement learning for subgoal decomposition to mitigate reasoning vulnerabilities (Ren et al., 2025), while frameworks like Math-Shepherd introduce step-by-step process rewards to address training signal sparsity without requiring human annotations (Wang, P. et al., 2024). Empirical evaluations indicate that unverified autoregressive reasoning degrades exponentially as the reasoning chain length increases, prompting the development of cumulative reasoning frameworks to sustain accuracy (Zhang, Y. et al., 2024). To isolate neural-to-symbolic translation errors from logic failures, frameworks like Logic-LM empower language models with symbolic solvers for faithful logical reasoning (Pan et al., 2023), and process reward models are utilized to verify reasoning chains step-by-step (Lightman et al., 2024).

### Automated Compliance Checking

Translating complex regulations into verifiable systems has superseded purely probabilistic text generation. To ensure models ground their outputs in factual evidence, frameworks and benchmarks have been developed to force large language models to generate exact, fine-grained citations for their claims during long-context question answering (Gao et al., 2023; Zhang, J. et al., 2024). In domain-specific compliance, semantic web-based automated checking pipelines integrate external evaluations, such as Finite Element Analysis, to verify physical and structural constraints directly (Patlakas et al., 2024). Industry standards increasingly mandate this type of structural independence, deploying external engines decoupled from internal model guardrails.

### Technical Document Extraction

Verifying airworthiness compliance requires extracting information from visually rich documents. Vision-language models have unified extraction pipelines, explicitly bypassing traditional multi-stage optical character recognition to recover text and formatting (Blecher et al., 2024; Poznanski et al., 2025). Models like Qwen2.5-VL compress visual text representations while maintaining layout fidelity and decoding complex mathematical formulas in a single pass (Bai et al., 2025). For standard enterprise

workloads, modular pipelines like Docling, SmolDocling, and MinerU process digital PDFs using specialized, lightweight computer vision models such as RT-DETR and TableFormer (Livathinos et al., 2025; Nassar et al., 2025; Wang, B. et al., 2024). This pipeline targets logical row and column recovery, converting documents into structured JSON and Markdown formats optimized for language model ingestion.

## 3. THE VERIFIABLE CLAIM ARCHITECTURE

To mitigate data fragmentation and engineering overhead inherent in legacy airworthiness compliance processes, this system implements a verifiable claim architecture (VCA). Under the VCA framework, every compliance determination anchors directly to programmatically extracted technical data, linking empirical evidence to verification criteria. Engineering conclusions cannot enter standardized reports without traceable data lineage to specific, verifiable source text (Zhang et al., 2024). This constraint transforms compliance reporting from ad hoc document compilation into a structured, verifiable ledger of airworthiness evidence.

The VCA manages complex requirement cascades by propagating discrete equipment results through the requirements hierarchy into multi-criterion compliance reports. For example, localized subsystem test findings, such as line voltage or frequency amplitude extracted from a laboratory datasheet, propagate to satisfy sub-tier military standards, which aggregate into top-level MIL-HDBK-516C airworthiness criteria. To preserve audit transparency for reviewing authorities, each extracted finding maps directly to a persistent document anchor that identifies the exact section or table in the source report. If an engineer questions a regulatory conclusion, the system isolates the specific location of the underlying data.

Although the VCA accelerates data discovery and evidence routing, final decision authority remains strictly with United States Air Force engineers. The system operates within a human-in-the-loop co-writing workflow where automated processes perform data discovery, while human engineers evaluate, edit, and formally approve every compliance claim. The software cannot bypass verification constraints and lacks autonomous authority to finalize an airworthiness determination. Automating evidence discovery ensures that the SME's effort centers on engineering judgment and safety verification.

## 4. THE DATA PIPELINE

Automating evidence discovery within defined structural constraints improves primary airworthiness evaluation metrics by replacing manual document review with a repeatable workflow. The deterministic data pipeline (Figure 1) enforces structural uniformity across unstructured engineering artifacts, ensuring reproducible document evaluations.

### Document Parsing and Preprocessing

The data ingestion pipeline converts heterogeneous PDF and DOCX artifacts into standardized Markdown using the Docling library (Livathinos et al., 2025). For PDF files, the ingestion engine executes PdfPipelineOptions configured with structural table extraction (`do_table_structure=True`), cell alignment (`TableFormerMode.ACCURATE`), embedded image extraction, and optical character recognition to parse non-selectable textual layers. For DOCX files, processing uses WordFormatOption alongside SimplePipeline.

A preprocessing routine intercepts non-standard font symbols, normalizing them to standard Unicode characters to resolve font corruption, including checkmarks and checkboxes, prior to Markdown generation. Automated regular expression filters then sanitize the parsed text by stripping embedded image references, administrative headers, release dates, local system file paths, signature blocks, and structural artifacts.

### Multi-Tiered Keyword Matching Strategy

To isolate candidate text chunks within technical reports, the pipeline employs a deterministic, multi-tiered keyword matching mechanism:

- **Standards-Specific Lexicon:** A precompiled dictionary captures terminology tied to specific military standards, e.g., "10 kHz to 10 MHz," "Line Impedance Stabilization Network," and "LISN" for CE102 evaluations.
- **Empirical Outcome Indicators:** A secondary lexicon flags empirical performance descriptors, e.g., "compliant," "exceeded," "margin," "deviation," and "tolerance."
- **Exact Requirement Identifiers:** Alphanumeric requirement designations (such as "CE102" or "13.1.1") are treated as greedy matching tokens, enforced via whole-word regular expression boundaries to prevent substring collisions.
- **Dynamic Keyword Expansion:** The system supports dynamic ingestion of supplementary domain keywords through an external configuration file, appending entries directly to the results lexicon pool.

When a target keyword or standard ID occurs within a table, the algorithm parses bidirectionally to extract the complete table structure and its adjacent text context, preserving column headers and quantitative cell values

### Rule-Based Filtering

To eliminate false positives and isolate verifiable test data, the programmatic layer replaces keyword search with reproducible selection rules that evaluate textual context. To ensure deterministic evidence retrieval, the architecture applies strict Boolean filters that discard candidate text

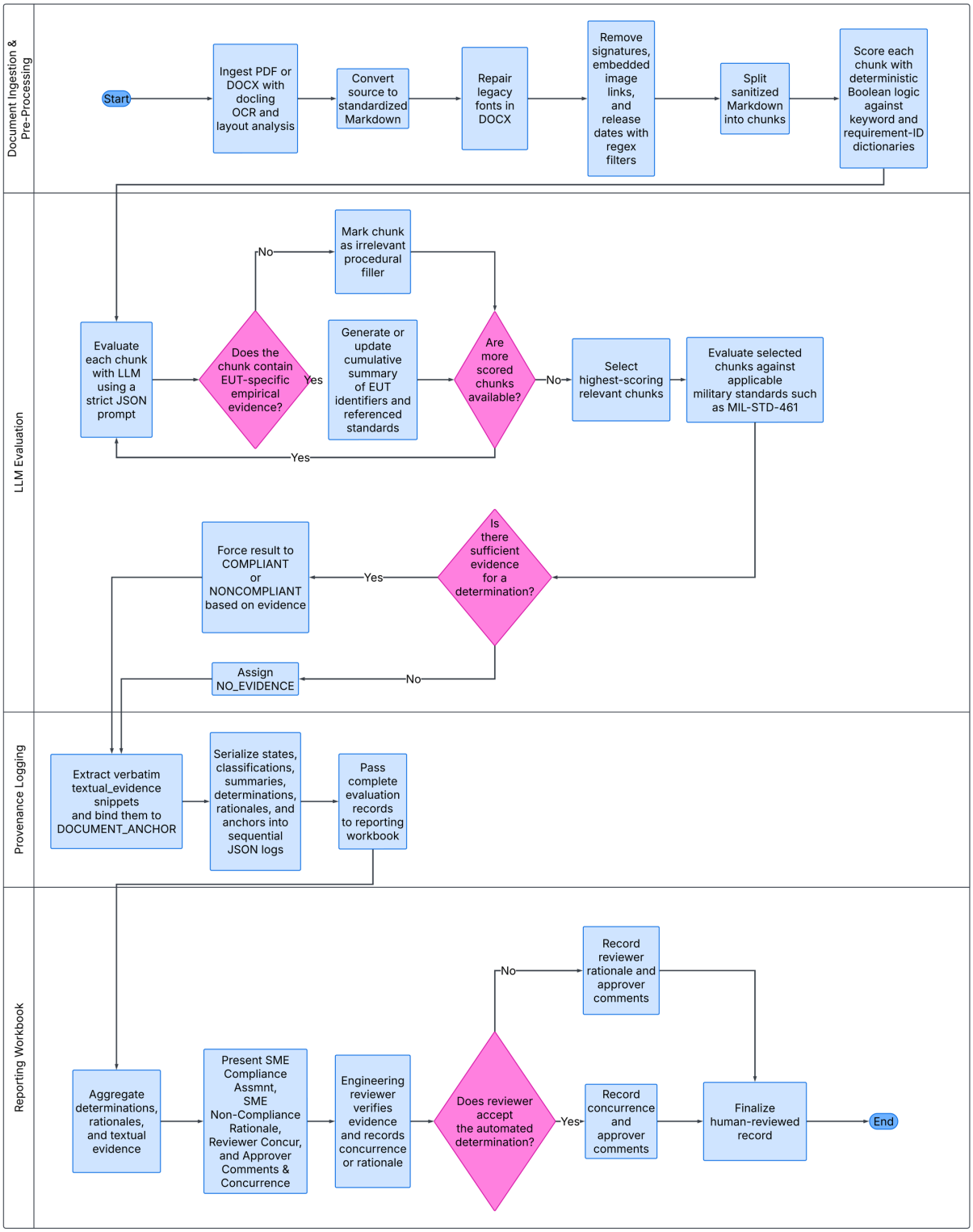


Figure 1. Block diagram of the airworthiness compliance system.

segments when explicit exclusion criteria are met. For example:

1. **Exclusion of Alternative Criteria:** The pipeline dynamically filters text segments detailing unrelated test methods. For example, the system discards evidence evaluating the MIL-STD-461 CE102 requirement when querying CS114 or validating CE106.

1. **Length Threshold Validation:** The pipeline discards extracted evidence if the character count falls below a fixed threshold, filtering non-substantive fragments such as document front matter or isolated page footers.
2. **Enforcing Document Authority Signposts:** For generalized engineering concepts that may occur in non-compliance contexts, such as "electrical grounding," the system verifies the presence of explicit military handbook identifiers or specific compliance tags (e.g., MIL-STD-461) within the immediate evidence context before approving the segment.

Chunks of text that trigger any of the Boolean rejection filters are no longer processed.

Because the software relies exclusively on string matching and fixed logic loops, the extraction script is deterministic. Executing the pipeline on 1,000 pages of test data produces identical noise filtering and table isolation across repeated runs. This reproducible selection logic provides compliance auditors with a fully auditable dataset that yields verifiable evidence across trials.

**Deterministic Filtering and Chunk Prioritization**

Before inference, the pipeline executes rule-based filtering to optimize context window utilization and eliminate non-evidential text. The ingestion filter excludes files whose naming conventions denote procedural or assembly guidelines, such as "Installation and Removal Procedures," "Acceptance Test Procedure." Sections containing 150 characters or fewer are discarded as deficient in verifiable evidence. If a chunk and its associated metadata explicitly reference alternative military standard criteria without citing the target requirement, the system bypasses evaluating the chunk for the current criterion. Candidate chunks meeting these validation thresholds are sorted using a ranking tuple: [is_relevant (boolean), greedy_keywords count, standards_keywords count, results_keywords count]. This ordering ensures that evidence-dense passages are prioritized during downstream analysis.

**Structured Output Enforcement**

All LLM reasoning phases enforce strict JSON schemas to guarantee programmatic parsing, eliminating unformatted conversational text across four functional stages:

- **Relevance Classification:** Returns an {"is_relevant": boolean} payload to distinguish empirical outcome data from procedural background text.
- **Cumulative Context Tracking:** Generates a structured object containing equipment under test (EUT) identifiers, mentioned standards and tests, chunk content nature, signposting for future evidence, and a brief narrative overview to propagate context across sequential chunks.
- **Compliance Determination:** Emits an outcome (COMPLIANT, NONCOMPLIANT, or NO_EVIDENCE), an analytical rationale, and a list of textual evidence and visual evidence (e.g., plots of test data) objects pairing verbatim source excerpts (text) with exact structural locations (e.g., section heading, table title).
- **Plan/Result Detection:** Executes a secondary validation pass to filter determinations derived from prospective testing intent rather than completed test records.

Because LLM text responses can contain syntax anomalies, the pipeline routes the generation stream through an automated repair layer. This validation layer parses and corrects malformed strings, validating output against the target schema to ensure key-value alignment and anchor completeness before writing records to the compliance database.

## 5. CONSTRAINING THE LARGE LANGUAGE MODEL

Preliminary investigations indicate that unconstrained LLM text generation produces non-deterministic classification errors: abstractive extrapolation causes false positives, and context fragmentation causes false negatives.

Without programmatic constraints, permissive sampling temperatures and unbounded token selection increase decoding variance. In standard configurations, the model relies on parametric associations to resolve ambiguous phrasing, treating prospective statements (e.g., "the subsystem will meet grounding specifications") as empirical evidence. This extrapolation asserts regulatory compliance without supporting quantitative test records, producing false positives.

Conversely, processing unstructured, full-length documents causes context fragmentation. When artifacts span dozens of pages across non-contiguous technical orders, tables, and appendices, attention mechanisms suffer from dilution over long context windows. As a result, the unconstrained model overlooks critical non-compliance indicators, such as localized frequency exceedances or anomalous transient spikes in secondary tables, and misclassifies non-compliant equipment as acceptable.

Evaluating identical artifact passages across successive inference cycles also yields contradictory compliance determinations. This stochastic instability prevents auditors from independently verifying results, which makes unconstrained generation unsuitable for safety-critical airworthiness assessments.

Although the deterministic pipeline establishes a verified, traceable evidence layer, generating formal compliance narratives requires text synthesis. LLMs summarize context and structure text effectively; however, unconstrained configurations introduce risks of abstractive drift and

ungrounded generation (Alansari & Luqman, 2026; Trilla et al., 2022). To safely deploy an LLM within an airworthiness certification workflow, the architecture programmatically constrains the model to structured text synthesis, preventing generation beyond the supplied evidence.

### 5.1. Strictly Isolated Context Windows

The primary defense against model hallucination is an isolated context environment. Rather than allowing the LLM to rely on parametric memory from training weights or query external repositories, the framework isolates the runtime environment. The context window contains only two components: the target requirement parameters (such as criteria from MIL-HDBK-516C) and the localized evidence chunk extracted by the deterministic filtering layer. This restriction blocks external retrieval and training priors, ensuring that synthesized compliance rationales derive solely from the supplied text. The model cannot infer missing parameters or synthesize an affirmative compliance narrative if the isolated context lacks explicit proof of test completion.

### 5.2. Model, System Prompts and Instruction Design

The system executes the 27-Billion parameter Gemma 3 variant (gemma3:27b-it-qat, Q4_0). Standard text generation applies higher temperatures and permissive sampling thresholds to select lower-probability tokens, introducing lexical variation. To enforce deterministic, uniform outputs and minimize variance across consecutive runs, the runtime configuration constrains decoding across a context window (num_ctx) of 16,384 tokens. Specifically, the runtime fixes the temperature to 0.2 to contract the next-token probability distribution over top-ranked candidates, while setting Top-P to 0.8 and Top-K to 40 to restrict token selection to a narrow probability mass and eliminate the low-probability tail.

#### System Persona and Output Constraints

The pipeline instructs the LLM to act as a quality control analyst and airworthiness certification assistant, evaluating documents against military standards including MIL-HDBK-516, MIL-STD-461, and MIL-STD-464. The architecture restricts all synthesis and analytical outputs to predefined JSON schemas, which eliminates conversational filler, introductory prose, and unformatted rationale.

#### Relevance Filtering and Context Tracking

To process long inputs, the pipeline combines binary classification with cumulative context tracking:

- **Evidence vs. Procedure Classification**: A binary classification step (`is_relevant: true/false`) filters individual text chunks, separating empirical EUT test outcomes from procedural descriptions, test plans, or definitions.
- **Cumulative Contextual Summarization**: A dedicated prompt extracts non-evidential metadata from each chunk, including EUT identifiers, referenced standards, and signposts pointing to upcoming evidence.
- **Context-Informed Chunk Evaluation**: The pipeline evaluates each incoming chunk alongside the running summary of prior chunks. This cumulative summary enables the model to resolve ambiguous references, such as attributing tabular test data to the correct EUT, without exceeding the context window.

#### Compliance Evaluation and Quality Control

The compliance determination stage applies targeted assessment and validation protocols:

- **Criterion-Bound Evaluation**: The model evaluates chunks against isolated requirements (such as CE102 limits) while disregarding adjacent, non-applicable data.
- **Verbatim Evidence Extraction**: The system outputs an explicit determination (COMPLIANT, NONCOMPLIANT, or NO_EVIDENCE), a synthesized rationale, and verbatim textual anchors from the source document (such as table numbers or section headings) to ground each decision.
- **Rationale Quality Control**: A secondary evaluation prompt audits the generated compliance rationales, classifying each rationale as grounded in historical test evidence or limited to prospective test intent.

### 5.3. Provenance Audit Layer and Eliminating Abstractive Drift

The cornerstone of the VCA's auditability is the provenance audit layer, which binds every generated narrative block to an immutable metadata tag tracking its pedigree back to the source engineering file. When the constrained LLM populates the “textual_evidence” and “visual_evidence” fields, each item in these arrays must include its technical asset (either a verbatim text string or a specific figure/table reference) and a mandatory DOCUMENT_ANCHOR mapping to that source data chunk. The architecture cross-references these extractions to populate compliance reports with reference tags linked to the validating source report and anchor.

This multimodal metadata linkage restricts output structures to keep narrative rationales bounded by the source text, preventing unverified commentary and abstractive drift. The model cannot generate compliance conclusions or extrapolations that lack a traceable link to the verified evidence layer, preserving an auditable data path for human verification and inspection.

## 6. SYSTEM SAFETY AND RISK ASSESSMENT INTEGRATION

The framework automates hazard tracking alignment to convert raw engineering deviations into standardized military risk metrics. The pipeline extracts safety descriptions directly from non-compliant technical documentation and maps them to hazard data populated from the system rule base. This mapping links observed electromagnetic anomalies or structural deviations directly to their corresponding system-level safety impacts.

When deviations are identified in source documentation, the architecture replaces subjective human grading with systematic risk mapping, assigning risk assessment codes (RACs) in accordance with MIL-STD-882E. If a data chunk reveals a localized failure, such as an emission exceedance, the system correlates the finding with severity and probability matrices established within a formal domain ontology.

To capture program risk across the lifecycle, the report schema maintains dedicated fields to track risk over time:

- **Initial RAC Baseline:** The pipeline documents the unmitigated hazard level observed during the initial failure.
- **Mitigated RAC Values:** The schema preserves a chronological record of risk reduction after engineering or operational controls are applied.

To link risk identification with safe flight operations, the generated output documents required operational restrictions and risk reduction measures. If a hazard cannot be mitigated through engineering controls, the pipeline extracts required operational boundaries, such as transmitter standoff distances or frequency-dependent flight limitations.

The framework assembles these safety matrices into the compliance report. Organizing hazard descriptions, initial RACs, mitigations, final RACs, and operational restrictions into structured data formats ensures traceability. This structural alignment prevents the loss of safety indicators within unstructured text, providing reviewers with a consistent baseline for hazard accountability.

## 7. SECURE DEPLOYMENT AND QUANTIFIABLE OUTCOMES

Deploying an automated text-synthesis environment within the defense sector requires adherence to institutional data-handling protocols and cloud infrastructure mandates. Operational security and information assurance must remain intact alongside document-processing utility. To meet these constraints, the system architecture integrates automated airworthiness tooling and deterministic data filtering within sandboxed, context-constrained environments. This design satisfies Department of Defense (DoD) cloud security standards while serving as a model for regulated engineering workflows.

### 7.1. Secure Deployment Lifecycle

The deployment pipeline is partitioned into progressive maturation phases to isolate sensitive technical data and prevent information leakage during model training and initialization:

- **Isolated Local Prototyping:** Core development, initial algorithmic vetting, and early-stage model tuning occur within local, air-gapped prototyping environments. This design ensures that raw technical orders and proprietary vendor data sheets remain contained within secure local hardware boundaries before scaling.
- **Transition to IL5 Platform One:** Following local validation, the architecture transitions to the accredited Impact Level 5 (IL5) Platform One cloud environment. This deployment allows the pipeline to ingest enterprise data while complying with DoD cloud security requirements and access management controls.

### 7.2. Quantifiable Ingestion and Processing Outcomes

Transitioning the airworthiness compliance lifecycle into a deterministic, NLP-driven workflow reduces SME review time while maintaining technical rigor. Programmatic evidence discovery and structural constraints improve evaluation metrics across several operational areas:

- **Accelerated Compliance Timelines:** Automating evidence extraction and document formatting reduces processing latency, compressing programmatic schedules and accelerating certification milestones.
- **Mitigation of Textual Fabrication:** Restricting the LLM environment to the isolated evidence layer prevents hallucination, eliminating ungrounded data, unverified assumptions, and unsupported compliance assertions.
- **Deterministic Reproducibility:** Replacing manual keyword searches with precompiled programmatic filters and Boolean selection loops ensures identical data extraction across successive document evaluations..
- **Audit Traceability:** The provenance audit layer anchors each generated narrative block to an immutable source file identifier and persistent Markdown anchor, enabling direct context verification for compliance reviewers.

### 7.3. Extensible Compliance and Scalable Focus

The system establishes a repeatable baseline that converts unstructured data into auditable, standardized formats. Because the core pipeline relies on string matching and document layout preservation rather than domain-specific heuristics, the architecture is domain-agnostic. The pipeline transfers to other compliance frameworks or military handbooks by updating the precompiled NLP data arrays and the corresponding criteria schema definition files.

This scalability extends to adjacent engineering standards, supporting tasks such as populating MIL-STD-882E hazard tracking matrices and generating flight-test limitation reports. By delegating data extraction, table reconstruction, and verification tasks to automated workflows, the system reduces administrative overhead while preserving human engineering oversight. This structure allows engineers to focus on safety-critical evaluations and compliance verification.

## 8. IMPACT

To evaluate architectural generalizability, the Air Force Life Cycle Management Center (AFLCMC) Mobility Directorate, Airworthiness, independently processed blind test cases. Although the pipeline was designed and optimized on the KC-46 platform, AFLCMC conducted this external validation using compliance artifacts from five distinct aircraft platforms. The authors did not have direct access to AFLCMC's evaluation dataset, which comprised 20 projects, 126 applicable criteria, 150 artifacts, and 7,346 pages of evidence. Consequently, standard quantitative performance metrics like precision, recall, and F1-score cannot be computed, and the reported findings reflect the aggregate outcomes provided directly by AFLCMC. This cross-platform evaluation demonstrates that the deterministic extraction and verification layers remain domain-agnostic and function without platform-specific modifications or retraining.

Comparing automated outputs to manual SME assessments demonstrates the system's reliability and designed conservatism. Across the 126 criteria, the system matched the initial SME evaluations in 63 instances. The remaining 63 discrepancies consisted of 53 false negatives and 10 false positives. A contextual review reveals that 51 of the 53 false negatives resulted from SME domain overrides rather than algorithmic classification failures. In these instances, the pipeline correctly identified non-compliance against documented MIL-STD thresholds. SMEs subsequently marked the requirements as “met” by applying external operational knowledge unavailable to the system (e.g., noting that the aircraft contains no radios operating at the non-compliant frequency). The pipeline executed the correct technical evaluation within its defined data boundary, while domain experts resolved application-specific exceptions using contextual facts absent from the source documentation.

The system generated 10 false positives by classifying items as compliant when SMEs determined them to be non-compliant. Five instances stemmed from prospective phrasing in test reports, where statements that a component “will demonstrate compliance” triggered an affirmative classification. The remaining five false positives exhibited no discernible systematic pattern or shared failure mode. While these discrepancies require human verification, restricting the language model to isolated technical evidence prevents ungrounded generation and focuses engineering oversight on safety-critical evaluations.

The system provides traceability by presenting the underlying evidence (including verbatim text, figures, and tabular data) used for each compliance determination. Linking each output to a source anchor enables SMEs to audit the system's rationale efficiently. This structured review workflow allows false positives to be identified and resolved, preserving human oversight throughout final certification.

Beyond improving evaluation fidelity and traceability, integrating the automated data pipeline reduces programmatic timelines. Exhaustive manual evidence discovery and aggregation have historically extended military airworthiness certification schedules to 18–36 months. By automating data extraction and document formatting, the framework is projected to reduce these timelines to 6–12 months. This acceleration is driven by a 90% reduction in engineering compliance prep time, specifically the hours spent manually searching technical orders and populating initial compliance matrices. This acceleration is supported by an estimated 50% reduction in expert labor hours dedicated to administrative document retrieval, improving fleet readiness while maintaining focused engineering risk assessment.

## 9. TECHNICAL AND OPERATIONAL LIMITATIONS

While the VCA significantly accelerates the compliance lifecycle and mitigates the risks of unconstrained generative models, its deployment within safety-critical PHM workflows is bounded by several technical and operational limitations. Acknowledging these constraints is essential for maintaining appropriate human-in-the-loop oversight and guiding future architectural refinements.

### The Epistemic Boundary of Document-Grounded AI

The primary limitation of the VCA stems directly from its core security feature: the strictly isolated context window. Because the architecture structurally prohibits external data retrieval and parametric memorization to prevent hallucination, the system operates under a strict “closed-world” assumption. It evaluates compliance based exclusively on the provided engineering artifact. As observed in the blind test evaluations (Section 7), this isolation generates operational false negatives when subsystem non-compliance is resolved by external, platform-level factors. For example, if a laboratory report indicates a frequency exceedance, the system correctly classifies the requirement as non-compliant. However, SMEs routinely override this by applying tacit operational knowledge, such as the fact that the specific aircraft variant does not operate avionics on the affected frequency. The VCA cannot bridge this gap autonomously, as it lacks access to contextual fleet configurations outside the ingested document.

### Linguistic Nuance and Prospective Phrasing

Although the deterministic filtering layer and low-temperature sampling effectively restrict the language model to verifiable evidence, the LLM inherently remains susceptible to subtle semantic ambiguities in engineering documentation. Early testing revealed a notable failure mode involving the misclassification of prospective or declarative intent as empirical proof. Phrases such as "the subsystem *will demonstrate* compliance" or "the component *is designed to meet* MIL-STD-461" occasionally triggered false positive compliance assertions from the model.

In response, explicit semantic guardrails and instruction constraints have been integrated into the deployed system to help the model reliably distinguish between theoretical design intent and actual quantitative test completion. While these guardrails effectively capture and filter known patterns of prospective phrasing, similar semantic edge cases may still be encountered in unseen documents that utilize novel linguistic structures or non-standard vendor terminology. Consequently, the provenance audit layer remains a critical operational fallback, allowing SMEs to rapidly identify and correct any residual false positives by tracing the generated claim directly back to its source document anchor.

### Limitations in Multimodal and Graphical Data Extraction

The current data pipeline excels at layout-aware parsing, successfully reconstructing multi-line tables, normalized symbols, and spatial text formats into structured Markdown. However, aerospace PHM compliance heavily relies on complex graphical data, such as raw spectrum analyzer outputs, finite element analysis heatmaps, and continuous waveform plots. The VCA currently treats these visual elements as opaque document anchors; it can extract the adjacent caption and route the image to the SME, but it does not currently mathematically or semantically interpret the data embedded directly within a graphical plot. Consequently, compliance criteria that rely solely on un-annotated charts require manual human interpretation, limiting the system's end-to-end automation capability for highly visual engineering domains.

### Operational Dependency on Human Oversight

Finally, the system is fundamentally engineered as an operational accelerator, not an autonomous decision-making agent. By design, the VCA lacks the autonomous authority to finalize an airworthiness determination. The deterministic constraints that make the system safe for military certification also enforce a rigid dependency on human reviewers. While the architecture reduces the administrative burden of data discovery by an estimated 50%, the reviewing engineers must still manually audit the cited evidence, resolve false positives, and formally sign off on the compliance matrices.

## 10. CONCLUSIONS

The final phase of Prognostics and Health Management certification has historically been limited by manual compliance review. This paper presents the Verifiable Claim Architecture (VCA), a deterministic, NLP-driven workflow connecting fragmented engineering documentation with regulatory mandates. By implementing layout-aware parsing, Boolean filtering, and constrained LLMs, the pipeline extracts technical evidence into a structured JSON schema while eliminating textual hallucination. Every generated narrative block is anchored through a provenance audit layer to immutable metadata tags linked to specific source document anchors, providing audit transparency for reviewing authorities.

The integration of system safety and risk assessment capabilities maps engineering deviations to RACs and operational flight restrictions in accordance with MIL-STD-882E. Blind test evaluations conducted by the Air Force Life Cycle Management Center (AFLCMC) Mobility Directorate across 7,346 pages of evidence from multiple aircraft platforms confirm the architecture's generalizability and conservative safety posture. Because the framework relies on string matching and document layout preservation rather than domain-specific heuristics, it is domain-agnostic and scales to adjacent compliance frameworks. By reducing traditional certification timelines from 18–36 months to 6–12 months, anchored by a 90% reduction in engineering compliance prep time, and decreasing expert labor hours by an estimated 50%, this architecture enables safety engineers to prioritize safety-critical verification and risk assessment.

## REFERENCES

Alansari, A., & Luqman, H. (2026). Large language models hallucination: A comprehensive survey. *Computer Science Review*, *61*, 100970.

Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., & Qwen Team. (2025). Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Blecher, L., Cucurull Preixens, G., Scialom, T., & Stojnic, R. (2024, May). Nougat: Neural optical understanding for academic documents. In *International Conference on Learning Representations* (Vol. 2024, pp. 37646-37663).

Cejas, O. A., Azeem, M. I., Abualhaija, S., & Briand, L. C. (2023). Nlp-based automated compliance checking of data processing agreements against gdpr. *IEEE Transactions on Software Engineering*, *49*(9), 4282-4303.

Chen, L., Gu, J., Huang, L., Huang, W., Jiang, Zhicheng, Jie, A., Jin, X., Jin, X., Li, C., Ma, K., Ren, C., Shen, J., Shi, W., Sun, T., Sun, H., Wang, J., Wang, S., Wang, Zhihong, Wei, C., Wei, S., Wu, Y., Wu, Y., Xia, Y., Xin, H., Yang, F., Ying, H., Yuan, H., Yuan, Z., Zhan, T.,

Zhang, C., Zhang, Y., Zhang, G., Zhao, T., Zhao, J., Zhou, Y., & Zhu, T. H. (2025). Seed-prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint arXiv:2507.23726*.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. (2024). Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024* (pp. 3563-3578).

Fu, S., & Avdelidis, N. P. (2023). Prognostic and health management of critical aircraft systems and components: An overview. *Sensors*, *23*(19), Article 8124.

Gao, T., Yen, H., Yu, J., & Chen, D. (2023, December). Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 6465-6488).

King, J. C. (2021). *Using model based system engineering to identify safety critical functions in airworthiness certifications* (Publication No. AFITENVMS21M241) [Master's thesis, Air Force Institute of Technology]. Defense Technical Information Center.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., & Cobbe, K. (2024). Let's verify step by step. In *International Conference on Learning Representations* (Vol. 2024, pp. 39578-39601).

Livathinos, N., Auer, C., Lysak, M., Nassar, A., Dolfi, M., Vagenas, P., Berrospi, C., Omenetti, M., Dinkla, K., Kim, Y., Gupta, S., Teixeira de Lima, R., Weber, V., Morin, C., Meijer, I., Kuropiatnyk, V., & Staar, P. W. J. (2025). *Docling: An efficient open-source toolkit for AI-driven document conversion* (arXiv:2501.17887)

Lorenzo, W. P., & Takahashi, T. T. (2024). Can We Fly it? Yes We Can: A Comparative Study of Military Airworthiness and Flight Operations. In *AIAA SCITECH 2024 Forum* (p. 2213).

Löwenmark, K., Taal, C., Schnabel, S., Liwicki, M., & Sandin, F. (2022). Technical Language Supervision for Intelligent Fault Diagnosis in Process Industry. *International Journal of Prognostics and Health Management*, *13*(2).

Nassar, A., Marafioti, A., Omenetti, M., Lysak, M., Livathinos, N., Auer, C., Morin, L., Teixeira de Lima, R., Kim, Y., & Staar, P. (2025). SmolDocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 21972-21983).

Onilede, M. O. (2026). Designing an audit-safe analytics architecture for aircraft maintenance decision support in regulated aviation environments. *Aircraft Engineering and Aerospace Technology*, 1-11.

Pan, L., Albalak, A., Wang, X., & Wang, W. (2023, December). Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 3806-3824).

Patlakas, P., Christovasilis, I., Riparbelli, L., Cheung, F. K., & Vakaj, E. (2024). Semantic web-based automated compliance checking with integration of Finite Element analysis. *Advanced Engineering Informatics*, *61*, 102448.

Pavlyuk, D., & Alomar, I. (2026). Artificial Intelligence Technologies for Aircraft Maintenance: A Systematic Literature Review. *International Journal of Prognostics and Health Management*, *17*(1).

Poznanski, J., Rangapur, A., Borchardt, J., Dunkelberger, J., Huff, R., Lin, D., Wilhelm, C., Lo, K., & Soldaini, L. (2025). olmocr: Unlocking trillions of tokens in pdfs with vision language models. *arXiv preprint arXiv:2502.18443*.

Ren, Z. Z., Shao, Z., Song, J., Xin, H., Wang, H., Zhao, W., Zhang, L., Fu, Z., Zhu, Q., & Yang, D. (2025). Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*.

Trilla, A., Mijatovic, N., & Vilasis-Cardona, X. (2022). Towards learning causal representations of technical word embeddings for smart troubleshooting. *International Journal of Prognostics and Health Management*, *13*(2).

Wang, B., Xu, C., Zhao, X., Ouyang, L., Wu, F., Zhao, Z., ... & He, C. (2024). Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., & Sui, Z. (2024). Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 9426-9439).

Weiss, B. A., & Brundage, M. P. (2021). Measurement and evaluation for prognostics and health management (phm) for manufacturing operations–summary of an interactive workshop highlighting phm trends. *International journal of prognostics and health management*, *12*(1), 10-36001.

Zhang, J., Bai, Y., Lv, X., Gu, W., Liu, D., Zou, M., ... & Li, J. (2025, July). Longcite: Enabling llms to generate fine-grained citations in long-context qa. In *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 5098-5122).

Zhang, Q., Liu, J., & Chen, X. (2024). A literature review of the digital thread: Definition, key technologies, and applications. *Systems*, *12*(3), 70.

Zhang, Y., Yang, J., Yuan, Y., & Yao, A. C. C. (2023). Cumulative reasoning with large language models. *arXiv preprint arXiv:2308.043*