Comments on ETSI TS 104 158-3 V0.0.2

Securing Artificial Intelligence TC (SAI);
AI Incident Reporting
Part 3: AI Common Incident Expression (AICIE) Security Container

Heather Frase, PhD

CEO, Veraitech

March 2026

Please cite as:
Frase, H. (2026). Comments on ETSI TS 104 158-3 V0.0.2, Securing Artificial Intelligence TC (SAI); AI Incident Reporting Part 3: AI Common Incident Expression (AICIE) Security Container. March 2026. Retrieved from https://veraitechus.com/aicie-comments_part3/

Table of Contents

1. Introduction and Scope

This document provides comments on ETSI TS 104 158-3 V0.0.2, the Security Container component of the AI Common Incident Expression (AICIE) specification. It is a companion to my earlier comments on Part 2, the Common Container [Frase 2025].

The focus of these comments is Part 3. However, reviewing Part 3 in isolation revealed structural and coordination issues that span the multi-part specification. Section 2 addresses these cross-cutting issues. Part 1 (Global Framework) has not been comprehensively reviewed for this submission, and comments here are limited to coordination issues observable from Parts 2 and 3.

The AICIE specification addresses a genuine need. The OECD, regulatory bodies, and the research community have called for standardized AI incident reporting to enable trend analysis, cross-incident learning, and regulatory compliance. These comments are offered to strengthen the specification's utility for these purposes, with particular attention to data quality, usability, and analyzability.

My perspective is shaped by work on AI incident taxonomies and reliability frameworks, including contributions to the AI Incident Database, MLCommons AI Risk & Reliability working group, and OECD Network of Experts on AI. My comments emphasize data quality, usability, and analytical utility rather than security-specific technical details. Where I lack expertise to offer specific recommendations, I have flagged issues for the committee's consideration.

2. Cross-Cutting Issues Across All Parts

The AICIE specification is structured as a multi-part deliverable. Part 1 establishes a global framework for discovering and linking AI incident resources. Part 2 provides a "common container" for general incident reporting, derived primarily from the OECD framework. Part 3 provides a "security container" for AI security incidents, drawing from MITRE ATLAS, VERIS, and academic research. This structure reflects a recognition, stated in Part 1, that "a basic bifurcation existed between the needs of AI incident information for 'AI safety' and for 'AI security.'"

While the separation of safety and security concerns has merit, the current specification lacks operational guidance for how the three parts work together. Several structural issues span all parts and will affect the usability and analyzability of incident data.

2.1 Safety/Security Bifurcation Lacks Operational Guidance

Part 1 acknowledges the bifurcation between AI safety and AI security needs but provides no guidance on when a reporter should file Part 2, Part 3, or both.

Some incidents clearly fall into one container. A hiring algorithm that produces discriminatory outcomes due to biased training data, with no attack involved, is a safety incident (Part 2). Model weights stolen from a company's servers, with no downstream harm to users from the AI system itself, is a security incident (Part 3). The separation of containers makes sense for these cases.

But many incidents span both domains. An adversarial attack on an autonomous vehicle (security) may result in physical injury (safety). A prompt injection attack (security) may cause a system to generate harmful content, provide dangerous advice, or expose user data (safety). Deepfake media created through AI (security) may be used to defraud or defame individuals (safety). The specification provides no guidance for these cases.

The specification does not address several practical questions that will arise for reporters.

- **When should both containers be filed?** If an incident has both security and safety dimensions, should the reporter complete both Part 2 and Part 3? The specification is silent.
- **How are related reports linked?** If the same incident generates both a Part 2 and Part 3 filing, there is no shared incident identifier to connect them. Part 2 has no incident ID field at all. Part 3 has AICIESCincidentID, but there is no mechanism to reference this from Part 2.
- **Which container is primary?** For incidents with both dimensions, which report takes precedence if information conflicts?

I recommend establishing clear guidance on when to file each container, requiring a common incident ID when both containers are filed for the same incident, and specifying handoff triggers between Part 2 and Part 3. Part 2 field 4.1.3 (systemRelationship) includes "target of attack" as an enumeration with guidance to "also complete the security container specified in Part 3." This is a useful start, but Part 3 has no reciprocal guidance pointing back to Part 2.

2.2 No Common Incident Metadata Layer

Part 1 is a resource discovery framework. It describes how to find and link to resources such as repositories, specifications, and other directory lists. It does not provide incident-level metadata. This means there is no shared definition layer for fields that appear in both Part 2 and Part 3.

Several fields capture the same information in both containers, but with different names, structures, or formats. These include incident date, geographic location, submitter/reporter information, AI system developer, autonomy level, harm descriptions, and affected parties. Without a common metadata layer, reporters filing in both containers must enter the same information twice, potentially leading to inconsistencies.

I recommend considering one of three approaches. First, a common incident metadata header could be created that both Part 2 and Part 3 reference, having shared fields like incident ID, date, location, reporter, and developer. Second, Part 2 could be designated as the primary container for shared fields, with Part 3 cross-referencing rather than duplicating. Third, if duplication is retained, explicit guidance should clarify which definition governs when formats differ.

2.3 Duplicate Fields With Structural and Format Conflicts

Over a dozen fields appear in both Part 2 and Part 3, or capture related concepts with different approaches. The table in Appendix A details these overlaps. Several have different structures or formats that create reconciliation problems.

- **Date.** Part 2 uses UTC format [year]-[month]-[day]. Part 3 uses the VERIS time/date format with separate day, month, and year components. Reporters and systems processing both containers will need format conversion logic.

- **Location.** Part 2 uses country codes (national level). Part 3 uses postal codes (local level). An incident in Paris would be recorded as "France" in Part 2 and as a postal code like "75001" in Part 3. These capture different granularities, serve different analytical purposes, and neither can be derived from the other.
- **Submitter information.** Part 2 uses a single field with stakeholder group enumerations. Part 3 splits this into separate reporter name and reporting organization fields.
- **Developer.** Part 2 has AICIECCaiSystemDeveloper. Part 3 has AICIESCdeveloper. Both are free text with no structural difference, but the duplication creates data entry burden.
- **Autonomy level.** Both containers include this field with identical intent but no cross-reference.
- **Harm and affected parties.** Part 2 has structured enumerations for harmType and partiesAffected. Part 3 has free-text fields for resultingHarm and integer fields for impactedUserTypes and user counts. These approaches are fundamentally different and cannot be easily reconciled.

For each duplicate field, I recommend clarifying whether the Part 2 or Part 3 definition governs, unifying the definitions where possible, adopting consistent formats, or moving shared fields to a common metadata layer.

2.4 Clarifying Definitions Needed Across All Parts

Most enumerated fields across Parts 2 and 3 provide labels without definitions. In my comments on Part 2 [Frase 2025], I noted that terms like "serious hazard," "hazard," "incident," and "serious incident" in the severity field (4.1.10) lack distinguishing criteria, inviting inconsistent application across reporters.

Part 3 has similar issues. Fields reference attack types, mitigation levels, autonomy levels, and investigation phases without defining what each value means. The document structure groups fields into "Meta Information," "AI System Information," "Incident Specification," and "Incident Implication," but these groupings are not explained, and the boundaries between them are unclear.

I recommend adding brief definitions for all enumeration values across both Part 2 and Part 3, either inline in field descriptions or in a combined glossary annex.

3. Summary of Part 2 Findings

My comments on Part 2 [Frase 2025] identified several issues that inform this review of Part 3. A brief summary follows.

Mixing of units. Several enumerated fields in Part 2 combine incompatible classification dimensions within a single field. AICIECCsystemRelationship (4.1.3) mixed the AI system's causal role with human behavioral factors. AICIECCharmType (4.1.11) mixed harm types with harm targets and normative frameworks. AICIECCharmConsequence (4.1.12) mixed actual consequences with response actions and measurement categories. AICIECCpartiesAffected (4.1.14) mixed demographic, role-based, and entity-type categories. In each case, my proposed revisions aimed to ensure that every enumeration within a field answers the same question, measured in the same “units.”

Vague or undefined categories. Terms like "serious hazard," "hazard," "incident," and "serious incident" in the severity field lack distinguishing criteria. "Public interest" in the harm type field was vague and overlapping with other categories. Without clarifying definitions, reporters will apply these terms inconsistently, limiting the analytical value of aggregated data.

Missing analytically important categories. Part 2 lacked structured fields for discrimination and bias, which is the most commonly documented AI-specific harm in incident databases. Privacy and personal data harms were also not clearly captured. I proposed revised enumerations to address these gaps.

Proposed new field for severity indicators. Part 2 lacked a mechanism to capture binary regulatory triggers such as fatality, serious bodily harm, critical infrastructure disruption, fundamental rights violations, involvement of minors, or widespread harm. I proposed a new AICIECCseriousIncidentIndicators field with boolean indicators aligned to EU AI Act and other regulatory thresholds.

JSON schema issues. Part 2's JSON schema in Annex A contained syntax errors, version mismatches, and typed multi-select fields as single strings. These technical issues would impede implementation.

Several of these issues recur in Part 3 or have parallels in Part 3's structure. The mixing-of-units problem is less prevalent in Part 3, but organizational problems and undefined categories appear throughout.

4. Part 3: High-Level Assessment

Part 3 provides a security container for AI incidents, drawing from multiple source frameworks including MITRE ATLAS, VERIS, the Grosse et al. research, and the OECD common reporting framework. The document is at an earlier stage of maturity than Part 2, with several sections incomplete. The following observations address structural and design issues that affect data quality, reporter burden, and analytical utility.

I recognize that specification design involves tradeoffs. Structured fields improve analyzability but may not translate well across languages or jurisdictions. Conditional logic reduces reporter burden but adds implementation complexity. The observations below are offered to surface considerations that may inform those tradeoffs, not to suggest that simpler alternatives were overlooked.

4.1 Reporter Burden and Field Organization

Part 3 contains over 70 fields compared to approximately 30 in Part 2. While security incidents may require more detailed technical information, the current structure creates a significant reporter burden without clear prioritization.

- **Flat structure.** All fields are presented as a flat list with no conditional logic. A reporter must review all fields regardless of incident type. Many fields are only relevant in specific circumstances, such as attack fields when there was an attack, or data exfiltration fields when data was exfiltrated.

- **Expert knowledge required.** Many fields assume security practitioner expertise that general reporters may lack. Fields like AICIESCmodelSupplyChainModelWeights, AICIESCattackCost, and AICIESCmodelMetrics require insider knowledge that reporters often will not have.
- **Likely empty fields.** Fields requiring information about training data, model weights, system prompts, and attack costs will frequently be empty or marked "unknown" for incidents reported by affected parties, researchers, or the general public rather than system developers.

I recommend considering conditional field organization similar to VERIS, which groups fields by situation. For example, attack-related fields could be grouped with guidance to "complete this section if the incident involved an attack," while data exposure fields could be grouped with "complete this section if data was exfiltrated or exposed." This would reduce cognitive burden while preserving comprehensive coverage for incidents that warrant it.

I also recommend aligning attack-related field content with the MITRE ATLAS taxonomy for interoperability with existing security tooling, while adopting VERIS-style conditional field organization to reduce reporter burden.

4.2 Data Quality Concerns

Part 3 relies heavily on free-form text fields, which reduces the analyzability of incident data across reports.

- **Free-text where structure is possible.** Many TEXT fields capture information for which standardized taxonomies exist. AICIESCattack and AICIESCattackProcedure are free text, but MITRE ATLAS provides structured attack technique identifiers. AICIESCmitigationCategories is free text, but both MITRE and NIST provide mitigation taxonomies.
- **Integer fields without uncertainty guidance.** Fields for user counts (AICIESCnumberOfDirectlyImpactedUsers, AICIESCnumberOfIndirectlyImpactedUsers) are typed as INTEGER, suggesting precise counts. In practice, reporters will often have only estimates or ranges. Consider adding guidance that acknowledges approximate counts are acceptable, or providing range options as alternatives to precise integers.

Appendix B lists the free-form text fields that could benefit from structured enumerations.

4.3 Organizational Problems and Classification Mixing

My comments on Part 2 [Frase 2025] identified a "mixing of units" problem where a single enumerated field combines incompatible classification dimensions. For example, Part 2's harmType field mixed harm types ("physical injury"), harm targets ("critical infrastructure"), and normative frameworks ("human rights") within a single enumeration. This is properly called *classification mixing* or *intra-field mixing*, and it undermines analytical utility because different classification dimensions cannot be meaningfully compared or aggregated.

Part 3 has a different organizational problem that could be confused with classification mixing but is distinct. In Part 3, related fields are scattered across sections and unrelated fields are grouped together. This is *organizational mixing* or *inter-field mixing*. It creates reporter burden and logical confusion but does not have the same analytical consequences as classification mixing.

Part 3's heavy reliance on free-text fields means that classification mixing within enumerated fields is less prevalent. However, many Part 3 fields lack defined enumerations (see Appendix B.3). ***When these enumerations are defined, care should be taken to avoid the classification mixing problems identified in Part 2. Each enumerated field should capture a single classification dimension, with distinct fields for distinct dimensions.***

The following subsections address Part 3's organizational mixing problems.

4.3.1 Attack Information Split Across Sections

Fields describing attacks are distributed across two sections rather than grouped together.

In the "AI System Information" section (4.1.21-4.1.45), the following attack fields appear: AICIESCattack (4.1.36), AICIESCattackProcedure (4.1.37), AICIESCattackCost (4.1.38), AICIESCattackLifecycleStage (4.1.39), and AICIESCattackExfiltratedData (4.1.40).

In the "Incident Specification" section (4.1.46-4.1.63), additional attack fields appear: AICIESCattackEmployed (4.1.54) and AICIESCattackersIntent (4.1.63).

A reporter describing a single attack must navigate between sections to complete the picture. These fields should be grouped together. Grouping them into a dedicated "Attack Details" section, completed only when an attack was involved, would follow the VERIS-style conditional organization recommended in Section 4.1.

4.3.2 Report Mechanics Mixed with Incident Data

Several fields describe the reporting process rather than the incident itself. These include AICIESCtncAgreed (4.1.14) for terms and conditions status, AICIESCtncText (4.1.15) for terms and conditions text, AICIESCschemaVersion (4.1.16) for schema version, AICIESCstatus (4.1.17) for report status, AICIESCincidentSharing (4.1.46) for sharing status, and AICIESCinvestigationPhase (4.1.47) for investigation phase.

These are metadata about the report, not facts about the incident. Mixing report metadata with incident data conflates two different analytical dimensions. Report metadata could be separated into its own layer, distinct from both incident data and the common incident metadata proposed in Section 2.2.

4.3.3 "AI System Information" is Overloaded

The "AI System Information" section (4.1.21-4.1.45) combines five distinct conceptual categories in a single section.

- **Model technical specifications** including model type, version, specification, and metrics
- **Organizational information** including developer identity
- **Intended use and autonomy** including purpose and autonomy level
- **Supply chain information** including data sources, weights provenance, and code dependencies
- **Attack and detection details** including attack type, procedure, cost, and detection method

Attack details and detection details describe what happened during the incident, not what the AI system is. These are fundamentally different kinds of information. Grouping them together under "AI System Information" obscures the distinction between system characteristics and incident events.

I recommend reorganizing this section into distinct groupings. Developer identity and autonomy level, which duplicate Part 2 fields, could be moved to a shared metadata layer or cross-referenced rather than duplicated. Attack and detection details should be separated into their own sections, as they describe incident events rather than system characteristics. The remaining fields (model specifications, intended use, and supply chain information) more appropriately describe the AI system itself.

4.3.4 Timeline Information Scattered

Part 3 captures multiple timestamps across the incident lifecycle. Most timeline fields are well-grouped in the Meta Information section (4.1.6-4.1.10), covering first malicious action, initial compromise, data exfiltration, incident discovery, and containment.

However, detection date (AICIESCdetectionDate, 4.1.42) appears in the AI System Information section, and mitigation date (AICIESCmitigationDate, 4.1.58) appears in the Incident Specification section. Detection logically follows incident discovery and should be grouped with the other timeline fields. All timeline fields should be consolidated into a single section to support incident reconstruction and response-time analysis.

4.3.5 Redundant or Unclear Field Distinctions

Some fields appear to capture overlapping information without clear differentiation.

- **AICIESCattack (4.1.36) vs AICIESCattackEmployed (4.1.54).** Both describe the attack type or vector. The distinction between "attack" and "attack employed" is not explained.
- **AICIESCincidentDiscovery (4.1.9) vs AICIESCdetectionDate (4.1.42).** Both describe when the incident was found. The distinction between "discovery" and "detection" is not explained.
- **AICIESCcausedIncidents (4.1.19) vs AICIESClinksToOtherIncidents (4.1.71).** Both link to other incidents. The distinction between incidents "caused" and incidents "linked" is not explained.
- **AICIESCintendedUse (4.1.27) vs AICIESCactualUseDuringIncident (4.1.52).** These fields capture a useful distinction between intended and actual use, but they appear in different sections (AI System Information and Incident Specification) without cross-reference, making their relationship unclear to reporters.

For each of these pairs, I recommend either clarifying the distinction in the field descriptions, adding cross-references, or consolidating into a single field where appropriate.

Incident linking is particularly problematic. The two incident linking fields not only lack semantic differentiation but also appear in completely different parts of the document. AICIESCcausedIncidents (4.1.19) appears in Meta Information alongside core incident identifiers. AICIESClinksToOtherIncidents (4.1.71) appears at the very end of the field list, separated by over 50 fields. A reporter completing the form sequentially might not realize both fields exist, and the document provides no guidance on when to use one versus the other. This placement makes incident linking, which is essential for identifying attack campaigns and systemic vulnerabilities, unnecessarily difficult. See Section 6.2 for recommendations on a more structured approach.

4.4 Gaps in Security-Relevant Information

Part 3 lacks several fields that Part 2 provides and that would be relevant for security incidents.

- **No severity field.** Part 2 has AICIECCseverity with enumerations for hazard, serious hazard, incident, and serious incident. Security incidents also need severity assessment, but Part 3 has no equivalent field. The harm fields in Part 3 (4.1.65-4.1.69) capture impact but not severity classification.
- **No critical infrastructure enumeration.** Part 2 has AICIECCcriticalInfrastructure (4.1.19) with a detailed enumeration of infrastructure sectors. Critical infrastructure is a prime target for security attacks and a key regulatory trigger. Part 3 should either include this enumeration or cross-reference the Part 2 field.
- **Limited harm and impact structure.** Part 2 provides structured enumerations for harm type and affected parties. Part 3 has only free-text resultingHarm and integer impactedUserTypes. This limits analytical comparability between safety and security incidents.

Rather than duplicating these fields in Part 3, the specification could cross-reference Part 2's structured enumerations or include them in a shared metadata layer. This would ensure analytical consistency between safety and security incidents while reducing redundancy.

5. Part 3: Field-Level Issues

This section highlights field-level observations not already covered in Section 4 (structural problems) or Appendix B (editorial errors, undefined fields, free-form text).

5.1 Meta Information (4.1.0-4.1.20)

Timeline fields (4.1.6-4.1.10) are well-designed. These five fields capture key moments in the incident lifecycle: first malicious action, initial compromise, data exfiltration, incident discovery, and containment. This chronological structure supports incident reconstruction and response time analysis. As noted in Section 4.3.4, detection date (4.1.42) and mitigation date (4.1.58) should be consolidated with these fields.

5.2 AI System Information (4.1.21-4.1.45)

Model identification fields are reasonable. AICIESCmodelType (4.1.22), AICIESCmodelVersion (4.1.23), and AICIESCmodelSpecification (4.1.24) provide basic model identification, though 4.1.22 lacks a description (see Appendix B.1) and 4.1.24 has a typo in the field name.

5.3 Incident Specification (4.1.46-4.1.63)

AICIESCdeploymentContext (4.1.50) provides useful examples but no formal enumeration. The description mentions "cloud, on-premises, edge, user device" but these are examples, not a defined list. Converting to a formal enumeration would improve consistency.

Mitigation fields (4.1.56-4.1.62) are comprehensive but inconsistent. This group includes security measures, mitigation level, mitigation date, mitigation categories, lifecycle phase, additional details, and fail-safe measures. AICIESCmitigationLevel (4.1.57) and AICIESCmitigationLifecyclePhase (4.1.60) lack defined values (see Appendix B.3). AICIESCmitigationCategories (4.1.59) is free text but could reference MITRE ATLAS mitigations or NIST controls (see Appendix B.4).

5.4 Incident Implication (4.1.64-4.1.70)

Impact count fields (4.1.67-4.1.68) need uncertainty guidance. AICIESCnumberOfDirectlyImpactedUsers and AICIESCnumberOfIndirectlyImpactedUsers are typed as INTEGER, suggesting precise counts. In practice, reporters will often have only estimates or ranges. Consider adding guidance that acknowledges approximate counts are acceptable, or providing range options (e.g., 1-10, 11-100, 100+) as alternatives to precise integers.

AICIESCsecurityImpact (4.1.69) references CIA triad but lacks structure. The description mentions "confidentiality, integrity, availability" but provides no enumeration. VERIS provides a structured CIA impact model that could serve as a reference, with separate assessments for each dimension including data type affected, extent of impact, and duration. A structured approach would be more analytically useful than free text.

AICIESCattackGeneralization (4.1.70) captures an important question. This field asks whether the incident "potentially affects other AI systems," which is valuable for assessing broader risk. However, the description provides no guidance on how to assess generalizability.

6. Candidates for a Common Metadata Layer

Section 2.2 proposed a common metadata layer to address duplicate fields between Part 2 and Part 3. This section expands that recommendation to include both existing fields that should be consolidated and new fields that are relevant to both safety and security incidents but are not currently captured in either container.

These recommendations are offered as options for the committee's consideration, recognizing that some may require substantial revision. The case for undertaking such a revision rests on three benefits. First, a common metadata layer would enable cross-domain analysis of AI incidents, allowing researchers and regulators to identify patterns that span safety and security boundaries. Second, reducing duplicate fields would lower reporter burden and eliminate inconsistencies when the same incident is reported in both containers. Third, addressing structural issues now, while the specification is still maturing, is far less costly than retrofitting changes after implementations are deployed. The AICIE specification is positioned to become foundational infrastructure for AI incident reporting; investments in its architecture at this stage will yield returns across all future use.

6.1 Existing Fields to Consolidate

Appendix A identifies over a dozen fields that appear in both Part 2 and Part 3 with different structures or formats. Six of these are strong candidates for a shared metadata layer: incident ID, incident date, location, reporter/submitter information, developer, and autonomy level. These capture foundational incident information that should be consistent across both containers.

In addition, two fields currently in Part 2 only should be considered for the metadata layer, given their relevance to security incidents.

Severity. Part 2 has AICIECCseverity with enumerations for hazard, serious hazard, incident, and serious incident. Security incidents also require severity classification for prioritization and regulatory compliance. A shared severity field would enable consistent severity analysis across safety and security incidents.

Critical infrastructure. Part 2 has AICIECCcriticalInfrastructure with a detailed enumeration of infrastructure sectors. Critical infrastructure is a prime target for security attacks and a key regulatory trigger under both the EU AI Act and NIS2 Directive. A shared critical infrastructure field would support cross-domain analysis of infrastructure-related incidents.

6.2 New Fields for Both Containers

The following fields are not currently captured in either Part 2 or Part 3, but are relevant to both safety and security incident analysis. These are candidates for inclusion in a shared metadata layer.

Monitoring status and effectiveness. Part 3 captures security measures (4.1.56), detection method (4.1.43), and fail-safe measures (4.1.62), but does not explicitly ask whether monitoring was active at the time of the incident or whether existing monitoring detected the incident. Part 2 does not address monitoring at all. For both safety and security incidents, distinguishing between "monitoring detected the incident," "monitoring existed but did not detect the incident," and "no relevant monitoring was in place" would support root cause analysis and identify monitoring gaps. The type of monitoring may differ between safety (AI performance monitoring, output quality monitoring, guardrail monitoring) and security (intrusion detection, log monitoring, anomaly detection), but the question of monitoring effectiveness is relevant to both.

Prior known risk or foreseeability. Neither container captures whether the vulnerability or failure mode was previously known, whether it had been flagged in risk assessments, or whether similar incidents had occurred before. In cybersecurity, this is a well-established practice through CVE databases and vulnerability tracking. For AI safety, it corresponds to known failure modes and prior risk assessments. This information is relevant for assessing organizational responsibility, distinguishing novel failures from repeated failures, and identifying systemic issues.

Recurrence and incident linking. Neither container adequately captures whether an incident is a first occurrence or a recurrence of a previously reported issue, or how incidents relate to each other. Part 3 has two fields for linking incidents (AICIESCcausedIncidents 4.1.19 and AICIESClinksToOtherIncidents 4.1.71), but as noted in Section 4.3.5, the distinction between them is unclear, and they appear in completely different sections of the document.

A more structured approach would consolidate incident linking into a single mechanism with explicit relationship types:

- **Causal relationships.** This incident caused another incident, or this incident was caused by another incident. Essential for understanding cascading failures and attack chains.
- **Shared characteristics.** Incidents sharing the same vulnerability, the same AI system, or the same attack method. Essential for identifying systemic issues and tracking attack campaigns.
- **Temporal patterns.** This incident is a recurrence of a previously reported issue or part of an ongoing series. Essential for distinguishing isolated events from persistent problems.

This structure would support several analytical use cases that incident databases are meant to enable: tracking the spread of novel attack techniques, identifying AI systems with repeated failures, recognizing coordinated attack campaigns, and distinguishing truly novel incidents from variations on known problems. For both safety and security analysis, understanding incident

relationships is foundational for moving from individual incident response to systemic risk management.

Time from deployment to incident. Neither container captures when the AI system was first deployed. The interval between deployment and incident is analytically useful for understanding system maturity and failure patterns. In other safety domains, equivalent metrics are standard: time since manufacture and flight hours in aviation, time since installation in medical devices, and mileage in automotive safety. For AI systems, deployment-to-incident time helps distinguish day-one issues (design flaws, inadequate testing) from emergent issues (model drift, distribution shift, adversarial discovery over time). This is relevant to both safety and security incidents.

Direct and indirect impact distinction. Part 3 distinguishes between directly and indirectly impacted users (4.1.67-4.1.68), which is a useful distinction not currently in Part 2. Consider including this in a shared metadata layer so both safety and security incidents can capture the scope of impact consistently.

6.3 Report Metadata as a Separate Layer

Section 4.3.2 identified several Part 3 fields that describe the reporting process rather than the incident itself: terms and conditions status (4.1.14-4.1.15), schema version (4.1.16), report status (4.1.17), sharing status (4.1.46), and investigation phase (4.1.47).

These are report metadata, not incident metadata. They describe the state and handling of the report rather than facts about the incident. Mixing report metadata with incident data conflates two different analytical dimensions.

Consider separating report metadata into its own layer, distinct from both incident data and the common incident metadata proposed above. This would clarify the structure and support scenarios where the same incident has multiple reports (initial report, updates, different reporters) with different report metadata but shared incident metadata.

Report lifecycle and updates. Neither Part 2 nor Part 3 provides a clear mechanism for updating, supplementing, or correcting a previously filed report when new information comes to light. Other reporting frameworks address this explicitly. Suspicious Activity Reports (SARs), for example, distinguish between initial reports, supplemental reports (adding new information), and corrective reports (fixing errors in prior filings), with each subsequent report referencing the original. A similar approach for AI incident reports would support ongoing investigations and ensure that incident records reflect the most current understanding. This could include report type (initial, supplemental, corrective), reference to the prior report being updated, and a summary of what changed.

7. Additional Observations

The following observations do not propose specific changes but may inform future revisions.

Source attribution is useful. Part 3 includes bracketed source indicators ([M] for MITRE, [G] for Grosse, [O] for OECD, [V] for VERIS, [B] for Barbera) in its field descriptions. This transparency about field provenance is helpful for understanding design rationale and for tracing fields to their source frameworks. Consider retaining this in future versions and adding it to Part 2.

Opportunity for tighter MITRE ATLAS alignment. Part 3 draws from multiple sources, which contributes to some inconsistencies. MITRE ATLAS is the most established AI security taxonomy and is actively maintained. Closer alignment with ATLAS terminology, technique identifiers, and structure would improve interoperability with existing security tooling and reduce maintenance burden as ATLAS evolves.

Positive design choices. Part 3 makes several good design choices worth preserving. The timeline fields in Meta Information (4.1.6-4.1.10) provide a useful structure for incident chronology. The inclusion of fail-safe measures (4.1.62) addresses a gap in many incident reporting frameworks.

Reporter confidence level. Given that many Part 3 fields require insider knowledge (see Appendix B.5), reporters will often be uncertain about their responses. A confidence indicator could improve data quality by distinguishing verified facts from inferences. This is a data quality consideration rather than a standard field in incident reporting frameworks.

Appendix A: Part 2 / Part 3 Field Overlap

The following table identifies fields that appear in both Part 2 and Part 3, noting structural differences and format conflicts.

Concept	Part 2 Field	Part 3 Field	Differences
Version	AICIECCversion (4.1.0)	AICIESCversion (4.1.0)	Same structure. No conflict.
Incident Date	AICIECCdateFirstOccurred (4.1.5)	AICIESCincidentDate (4.1.4)	Different formats. Part 2 uses UTC [year]-[month]-[day]. Part 3 uses VERIS time/date.
Location	AICIECCincidentCountry (4.1.6)	AICIESCgeographicLocation (4.1.5)	Different granularity. Part 2 uses country codes. Part 3 uses postal codes. Not reconcilable.
Reporter/ Submitter	AICIECCsubmitterInformation (4.1.4)	AICIESCincidentReporter (4.1.2) + AICIESCreportingOrganisation (4.1.3)	Different structure. Part 2 uses single field with stakeholder enumerations. Part 3 splits into name and organization.
Developer	AICIECCaiSystemDeveloper (4.1.9)	AICIESCdeveloper (4.1.25)	Both free text. Duplication without added value.
Autonomy Level	AICIECCautonomyLevel (4.1.26)	AICIESCautonomyLevel (4.1.28)	Both text fields describing autonomy. No cross-reference between parts.
Intended Use	(covered in description)	AICIESCintendedUse (4.1.27)	Part 3 has explicit field. Part 2 captures in product description.
Harm	AICIECCharmType (4.1.11) + AICIECCharmConsequence (4.1.12)	AICIESCresultingHarm (4.1.65)	Fundamentally different. Part 2 uses structured enumerations. Part 3 uses free text.
Affected Parties	AICIECCpartiesAffected (4.1.14)	AICIESCimpactedUserTypes (4.1.66)	Different structure. Part 2 uses enumerated party types. Part 3 types this as INTEGER, which appears to be a specification error since "types" cannot be meaningfully represented as an integer.
Number Affected	(none)	AICIESCnumberOfDirectlyImpactedUsers (4.1.67) + AICIESCnumberOfIndirectlyImpactedUsers (4.1.68)	Part 3 only. Part 2 lacks count fields.
Additional Info	AICIECCadditionalInfo (4.1.29)	AICIESCcomments (4.1.72)	Both free text for supplementary information.
Actions Taken	AICIECCactionTaken (4.1.27)	AICIESCmitigationCategories (4.1.59) + related fields	Part 3 has more detailed mitigation structure with 5+ fields.
Steps to Reproduce	AICIECCstepsReproduced (4.1.28)	AICIESCattackProcedure (4.1.37)	Similar purpose but different framing. Part 2 is general reproduction. Part 3 is attack-specific.
Incident ID	(none)	AICIESCincidentID (4.1.1)	Part 3 only. Part 2 lacks incident identifier. Critical gap for linking reports.

Appendix B: Editorial, Technical, and Low-Priority Issues

B.1 Editorial Errors

- **Field name typo (4.1.24).** AICIESmodelSpecification is missing the "C" that appears in all other field names. Should be AICIESCmodelSpecification.
- **Description typo (4.1.15).** "SRING" should be "STRING."
- **Description typo (4.1.54).** "vectory" should be "vector."
- **Copy-paste error (4.1.13).** The description for AICIESCthirdPartyNotification is copied from 4.1.5 (AICIESCgeographicLocation). It reads "This optional INTEGER – constrained as a postal Postal Code value - describes the geographic location of the incident" when it should describe third party notification.
- **Copy-paste error (4.1.51).** The description for AICIESCbreadthOfDevelopment reads "thestage of learning where the attack occurred," which is copied from 4.1.39 (AICIESCattackLifecycleStage). It should describe breadth of development.
- **Copy-paste error (4.1.69).** The description for AICIESCsecurityImpact has extra text appended: "...the confidentiality, integrity, availability of the AI system.the lifecycle phase of mitigation." The final phrase should be removed.
- **Missing description (4.1.22).** AICIESCmodelType has no description at all.
- **Incomplete description (4.1.66).** AICIESCimpactedUserTypes description reads "the types of impacted" with no object. Should specify "types of impacted users" or similar.
- **Grammatical error (4.1.62).** AICIESCfailsafeMeasures description reads "describes there were any fail-safe measures" instead of "describes if there were any fail-safe measures."
- **Grammatical error (4.1.72).** AICIESCcomments description reads "This optional TEXT value additional comments" with the verb "describes" missing.

B.2 Incomplete Content

- **Annex A (JSON schema).** Contains only "[TBD]" with no schema definition.
- **Annex C (Mapping to Grosse Report).** Section header exists but content is empty.
- **Annex D (Mapping to OECD).** Section header exists but content is empty.

B.3 Fields Lacking Definitions

The following fields reference categories, levels, or phases without defining what each value means. Reporters will interpret these inconsistently without clarifying definitions.

When defining these enumerations, care should be taken to avoid classification mixing (see Section 4.3). Each enumerated field should capture a single classification dimension.

Field	Section	Issue
AICIESCstatus	4.1.17	"Status of the report" with no defined values
AICIESCassuranceCategory	4.1.18	"Assurance category" with no defined values
AICIESCautonomyLevel	4.1.28	"Level of autonomy" with no scale or categories
AICIESCattackLifecycleStage	4.1.39	"Stage of learning where the attack occurred" with no defined stages
AICIESCinvestigationPhase	4.1.47	"Investigation phase" with no defined phases
AICIESClifecycleStage	4.1.49	"Stage of the AI lifecycle" with no defined stages
AICIESCdeploymentContext	4.1.50	Gives examples (cloud, on-premises, edge, user device) but not a formal enumeration
AICIESCbreadthOfDevelopment	4.1.51	"Breadth of development" with no definition
AICIESCmitigationLevel	4.1.57	"Level of mitigation" with no defined levels
AICIESCmitigationLifecyclePhase	4.1.60	"Lifecycle phase of mitigation" with no defined phases
AICIESCsecurityImpact	4.1.69	References "confidentiality, integrity, availability" but no structured enumeration

B.4 Free-Form Text Fields

The following TEXT fields could benefit from structured enumerations. Where standardized taxonomies exist (noted in parentheses), the specification could reference them rather than relying on free-form text.

Important: When converting these free-text fields to structured enumerations, care should be taken to avoid classification mixing (see Section 4.3). Each enumerated field should capture a single classification dimension. If multiple dimensions are relevant, use separate fields rather than combining them in a single enumeration.

Attack and Detection Fields

- AICIESCattack (4.1.36) — could reference MITRE ATLAS technique identifiers
- AICIESCattackProcedure (4.1.37) — could reference MITRE ATLAS procedures
- AICIESCattackEmployed (4.1.54) — could reference MITRE ATLAS or VERIS threat actions
- AICIESCdetectionMethod (4.1.43) — could reference VERIS discovery methods
- AICIESCdetectionDataSource (4.1.45) — could reference MITRE data sources

Mitigation Fields

- AICIESCmitigationCategories (4.1.59) — could reference MITRE ATLAS mitigations or NIST controls
- AICIESCsecurityMeasures (4.1.56) — could reference NIST SP 800-53 controls

Impact and Harm Fields

- AICIESCresultingHarm (4.1.65) — could cross-reference Part 2 harmType enumeration

System Information Fields

- AICIESClifecycleStage (4.1.49) — could reference OECD AI lifecycle stages
- AICIESCdataTypes (4.1.30) — could enumerate common data types (text, image, audio, video, structured, multimodal)

Appropriately Free-Form

The following TEXT fields are appropriately free-form and do not require structured enumerations:

- AICIESCcomments (4.1.72)
- AICIESCmitigationAdditionalDetails (4.1.61)
- AICIESClinksToOtherIncidents (4.1.71)
- AICIESCregulatoryViolation (4.1.64) — jurisdictional variability makes a fixed enumeration impractical

B.5 Fields with Reporter Knowledge Constraints

The following fields require insider knowledge that most reporters will not have. These fields may be valuable for first-party reporting by developers or operators but will frequently be empty or marked "unknown" for third-party reports.

Requires Developer/Operator Knowledge

- AICIESCmodelSupplyChainData (4.1.29) — "if publicly available data was used, and if yes, from where"
- AICIESCmodelSupplyChainModelWeights (4.1.32) — "if the model weights were derived from an existing model"
- AICIESCmodelSupplyChainCode (4.1.33) — "what public code was used to generate and run the model"
- AICIESCmodelMetrics (4.1.26) — requires access to model evaluation results
- AICIESCsystemPrompt (4.1.35) — typically proprietary and not disclosed
- AICIESChardware (4.1.34) — "what hardware was used, and if it could have affected the incident"
- AICIESCdataset (4.1.31) — "level of dataset used" requires insider knowledge

Requires Attacker Knowledge or Speculation

- AICIESCattackCost (4.1.38) — cost of the attack from the attacker's perspective
- AICIESCattackersIntent (4.1.63) — often unknown or speculative
- AICIESCknowledge (4.1.55) — "knowledge required for the attack" requires inference

Situational Fields

- AICIESCmodelExclusiveVulnerability (4.1.41) — "if the attack targeted the model explicitly and the model only"
- AICIESCphysicalDomain (4.1.48) — boolean for physical environment incidents; increasingly relevant as AI systems are deployed in cyber-physical contexts
- AICIESCattackGeneralization (4.1.70) — "if the incident potentially affects other AI systems"; valuable for systemic risk assessment but requires expert judgment

Observation: Part 3 appears designed for comprehensive first-party reporting by security teams with full access to system internals. Many fields will be inapplicable or unanswerable for third-party reporters, researchers, or affected individuals. Consider identifying a minimum viable subset of fields for different reporter types or marking fields as "first-party only" to set appropriate expectations.

References

Frase, H. (2025). Comments on ETSI TS 104 158-2 V0.0.4, Securing Artificial Intelligence TC (SAI); AI Incident Reporting Part 2: AI Common Incident Expression (AICIE) Common Container. November 2025. Retrieved from https://veraitechus.com/aicie-comments/