



LANDING.AERO / RESEARCH & PRACTICAL GUIDANCE

EASA Standby and Reserve Rules: Call-Out Test Matrix

LANDING.AERO is a Montreal software studio that builds custom, AI-assisted software for flight operations: operations tools, data integration, workflow automation and dashboards designed around how an airline, charter operator or flight department actually works.

Published by LANDING.AERO · 2026-10-15 · easa standby and reserve rules · easa ftl · standby call-out · airport standby · crew rostering · reserve notification · roster audit trail · flight operations software

Original article: <https://www.landing.aero/articles/en/easa-standby-reserve-call-out-test-matrix>



In this article

1. Executive Summary
 2. Introduction and Background
 3. Key Changes: Roster Transitions, Not Assumed New Law
 4. Event Vocabulary and the Evidence Model
 5. Transition Test Matrix and Event Timelines
 6. Implementation Considerations and Process Changes
 7. Timestamp Interpretation and Export Tests
 8. Vendor Demonstration and API Acceptance
 9. Data Analysis and Evidence
 10. Implications and Future Directions
 11. Frequently Asked Questions (FAQs)
 12. Conclusion
-

Executive Summary

European Union Aviation Safety Agency (EASA) standby and reserve rules require different roster states. In Part-ORO, organisation requirements for air operations, flight time limitations (FTL) are addressed by Subpart FTL. Under ORO.FTL.225, standby start and end times must be defined and notified in advance. Airport standby counts fully as duty for the specified cumulative-duty and rest provisions. Reserve, by contrast, involves assignments notified **at least 10 hours in advance**. A scheduling database therefore needs more than a final duty code to explain a changed assignment. (www.easa.europa.eu) (eur-lex.europa.eu)

For crew-scheduling leads, operations-control managers and information technology (IT) teams, the practical acceptance question is whether an authorized reviewer can reconstruct what was rostered, what was communicated, when contact occurred and when the crew member reported. EASA's frequently asked questions (FAQs) explain that a call during rostered other standby can lead to reporting after its planned end, with the response interval continuing the standby. Duties should in principle start within the operator's defined response time; calls belong within rostered availability. The FAQ also distinguishes changing other standby into airport standby from assigning home standby after reserve ends. These are agency explanations, subject to the controlling provisions and the operator's approved scheme. (www.easa.europa.eu) (www.easa.europa.eu)

The report's proposed test matrix evaluates **event provenance**, meaning the origin, timing and revision history of the evidence behind a result. The World Wide Web Consortium's PROV Ontology (PROV-O) provides a vocabulary for provenance, while CloudEvents provides a common event description. Neither is prescribed here as an aviation requirement. The useful design principle is to retain the original notification and its later corrections, rather than make the current roster the only surviving record. Microsoft's event-sourcing guidance describes that preservation pattern. (www.w3.org) (cloudevents.io) (learn.microsoft.com) (learn.microsoft.com)

Public product descriptions establish useful starting points, not acceptance of these scenarios under a particular scheme. FlightStratix advertises an ORO.FTL engine, Leon documents publication acknowledgements and programmatic report access, and AIMS describes crew notification lists. A vendor demonstration should connect those functions to a replayable case and inspectable export. The decision is

then concrete: use existing configuration when the evidence survives, add an integration when it is accessible elsewhere, or investigate a larger capability gap when historical evidence cannot be recovered. This report proposes that decision process; it provides no live crew-assignment verdict. (flightstratix.com) (wiki.leonsoftware.com) (wiki.leonsoftware.com) (www.aimsairlinessoftware.com)

Introduction and Background

A roster can display a plausible final assignment while leaving an important review question unanswered: **what did the crew member know at the relevant time?** A revised end time, a call near that end, or a later airport report creates several events for review. The operator needs a way to distinguish their meanings before applying its scheme. Pharos Production's current engineering guide already recognizes notified standby inputs and storing results against roster versions, so the useful next step is a transition-specific acceptance exercise. (pharosproduction.com) (pharosproduction.com)

The jurisdiction is European Union air operations. ORO.FTL.100 describes commercial air transport (CAT) operations with aeroplanes; the applicable certification-specification scope also needs checking for the particular operation. This report addresses the scheduled and charter CAT aeroplane context, without extending its examples to emergency medical service, air taxi, single-pilot or other excluded operations. **An air operator certificate (AOC) alone does not settle the applicable specification.** (www.easa.europa.eu)

The baseline compilation inspected for publication on **15 October 2026** is EASA's **Revision 24, March 2026**. Later entries were checked in the Air Operations index, including a Part-ORO amendment concerning flight simulation training devices. The compilation date should therefore be recorded as a research baseline, not represented as proof that all subsequent legislation is irrelevant. An operator's implementation review must identify its applicable amendments and effective dates. (www.easa.europa.eu) (www.easa.europa.eu)

The analysis distinguishes legal requirements, certification specifications (CS), acceptable means of compliance (AMC), guidance material (GM), agency FAQs and proposed software acceptance criteria. EASA explicitly says its FAQ answers are not legally binding. The operator's approved flight time specification scheme and operations manual remain central to the review. (www.easa.europa.eu) (www.easa.europa.eu)

The integration perspective is relevant because LANDING.AERO describes building applications around an operator's specific needs. That is an adjacent implementation route, not a crew-scheduling product or a source of regulatory approval. The article evaluates the evidence that an existing system or integration would need to expose. (www.landing.aero)

Key Changes: Roster Transitions, Not Assumed New Law

A revised standby end is a notification event

"Changes" in this report means changes to an assignment's state. It does not imply that the scenario interpretations were newly enacted in the publication year. EASA's revised-end FAQ explains that standby may cease following notification during standby and that minimum rest should follow notification of the revised end. (www.easa.europa.eu)

The proposed acceptance test should preserve **original end, revised end, notification event and effective cessation** separately. The reviewer should be able to see whether these coincide or differ, and why. A revision link is preferable to an unexplained overwrite: PROV-O explicitly represents a revised entity's relationship to its original. This is a design recommendation for reconstructability, not a required database technology. (www.w3.org) (learn.microsoft.com)

A near-end call does not make planned end the actual cessation

For testing, give call placement, communication outcome and reporting their own fields. Record the flight duty period (FDP) separately from the reporting event. A single “updated at” value cannot explain which event it represents. The Open Worldwide Application Security Project (OWASP) logging guidance distinguishes event time from logging time, a useful engineering distinction when a mobile record reaches the server later. (cheatsheetseries.owasp.org) (github.com)

Other standby becoming airport standby changes classification

A proposed record should retain the earlier other-standby interval and the airport reporting event, with the classification applying to each segment. A later correction should explain its relationship to the earlier interpretation. Microsoft's event-sourcing description uses compensating events to preserve originals. (learn.microsoft.com) (www.w3.org)

Reserve ending is different from rewriting active reserve

Reserve's definition uses elapsed advance notification, rather than a loose “next-day” label. The original implementing text specifies at least **10 hours** for notification of the assigned duty. (eur-lex.europa.eu)

Test reserve end and any subsequent standby start as distinct events. Preserve the notification of that subsequent standby and any assignment. The absence of a flight should not become permission to relabel reserve retrospectively as recovery rest. EASA addresses that particular retrospective classification separately. (www.easa.europa.eu)

Event Vocabulary and the Evidence Model

The proposed model has two layers: an event history and a current roster view derived from it. Microsoft describes reconstructing current state by replaying events. The proposed demonstration should reconstruct the original case and each corrected version. (learn.microsoft.com) (learn.microsoft.com) (www.w3.org)

Table 1 separates the proposed evidence fields. These are procurement and integration requirements for the test package, not a claim that EASA mandates these exact field names. OWASP's “when, where, who and what” provides a useful general logging framework. (cheatsheetseries.owasp.org)

Evidence group	Fields to preserve	Review question
Roster identity	Crew reference, assignment reference, roster version, operation and scheme references	Which published assignment is being reviewed?

Evidence group	Fields to preserve	Review question
Notified window	Standby or reserve type, location, notified start and end, publication reference	What availability period was communicated?
Communication	Call or message event, channel, outcome, delivery and acknowledgement evidence where available	Which event supports notification or contact?
Reporting	Designated reporting point, instructed report time, actual report time, source	When did reporting actually occur?
State transition	Earlier state, later state, effective time, reason, author	Which interval has which classification?
System chronology	Occurrence time, receipt time, recorded time, correction time	Was the event late, or merely imported late?
Decision provenance	Rule configuration, scheme version, input references, result and reviewer disposition	Can the result be reproduced and explained?

The table deliberately separates operational meaning from storage chronology. OWASP recommends interaction identifiers for related events. In this test package, a correlation reference should join a roster publication, notification attempt, acknowledged change and resulting assignment without pretending that those are the same event. An export should preserve both the individual records and their relationship. (cheatsheetseries.owasp.org)

Origin and responsibility should travel with corrections. PROV-O attribution links an entity to an agent. The proposed export can use simpler terminology, such as source system, editor and reviewer, provided the meaning is documented. A shared account or automated import should be identifiable as such rather than silently appearing to be a crew acknowledgement. (www.w3.org) (cheatsheetseries.owasp.org)

The following timelines are **hypothetical examples**. Their purpose is to test what the product preserves. They do not certify an assignment or replace scheme-specific calculations. An event timeline is useful precisely because a final-state view may conceal the sequence; replay is the relevant capability to demonstrate. (learn.microsoft.com)

Hypothetical Example: revised standby end

Published window -> standby begins -> revised-end notification -> effective cessation -> rest record -> later correction, if any

Hypothetical Example: near-end call

Published window -> call within availability -> original planned end -> actual report -> assigned duty -> review

Hypothetical Example: other standby to airport duty

For this hypothetical conversion, assume less than six hours of other standby.

Other standby -> assignment notification -> airport report (subsequent FDP counting origin in this conversion) -> airport duty segment -> later flight assignment -> review

EASA's conversion FAQ counts the maximum FDP from reporting for airport standby in this case. Ordinary airport standby under CS FTL.1.225(a)(2)(i) uses the start of the assigned FDP as its FDP counting origin. (EASA conversion FAQ) (CS FTL.1.225)

Hypothetical Example: reserve to later standby

Rostered reserve -> reserve end -> separately notified standby -> contact or release -> review

Transition Test Matrix and Event Timelines

Table 2 turns those paths into demonstrations. Its “expected evidence” column specifies the proposed test contract. The legal references identify what the reviewer must consult, not a universal pass/fail ruling generated by the article.

Scenario	ORO / CS / FAQ reference	Source event and state before/after	Expected evidence	Human exception owner	Vendor demonstration
Revised end, no FDP	ORO.FTL.225(a), GM1 CS FTL.1.225(a), revised-end FAQ. (www.easa.europa.eu)	Notified standby becomes ceased standby	Original window, revision notice, effective end, subsequent rest	Crew-scheduling lead with scheme reviewer	Show both versions and the evidence used to start rest review
Call before end, report after end	CS FTL.1.225(b), late-call FAQ. (www.easa.europa.eu)	Other standby continues through response to reporting	Call time, notified end, response requirement, actual report	Operations control and scheme reviewer	Replay without clipping the interval at planned end
Other standby to airport	ORO.FTL.225(d), conversion FAQ. (www.easa.europa.eu)	Other standby becomes airport duty in this scenario	Earlier interval, call, airport report, later FDP reference	Authorized FTL reviewer	Show segment classifications and the FDP origin used
Active reserve relabelled	CS FTL.1.230, reserve-transition FAQ. (www.easa.europa.eu)	Reserve remains active when a home-standby change is proposed	Original reserve window, proposed effective time, notification history	Crew-scheduling lead	Route the proposed overlap to review; retain original state
Reserve ends, later standby begins	Same reserve-transition explanation; ORO.FTL.225 notification requirement. (www.easa.europa.eu)	Ended reserve becomes a separate standby assignment	Reserve cessation and standby notification as independent events	Crew scheduling	Show that later standby was not a retrospective rewrite
Reserve ends without assignment	ORO.FTL.230, CS FTL.1.230; reserve/rest FAQ. (www.easa.europa.eu)	Reserve ends with no assigned duty	Rostered reserve, no-assignment outcome, subsequent activity	Authorized FTL reviewer	Distinguish reserve accounting from subsequent rest classification

Scenario	ORO / CS / FAQ reference	Source event and state before/after	Expected evidence	Human exception owner	Vendor demonstration
Unused reserve recoded as recovery rest	GM1 ORO.FTL.230(a), recovery-rest FAQ. (www.easa.europa.eu)	Historical reserve is proposed for recoding	Original publication, proposed replacement and reason	Scheme owner	Retain history and expose the recoding for review
Delayed or corrected evidence	Proposed integration acceptance case, not a separate EASA rule	Received record changes historical knowledge	Occurrence, receipt and correction times; earlier result	IT/data owner with operational reviewer	Replay the earlier view and corrected view independently

A useful outcome is not necessarily a green legality indicator. For an ambiguous communication record, **evidence incomplete** is a meaningful software result. The proposed workflow should identify what is missing and who can resolve it. OWASP's distinction between event and logging time supports this treatment of a delayed import; the eventual arrival of a record should not rewrite what the system knew earlier. (cheatsheetseries.owasp.org)

The event timeline should also distinguish a corrected fact from a changed interpretation. PROV-O's revision relationship provides a general way to link versions, while attribution identifies the responsible agent. A reviewer should be able to determine whether airport reporting was corrected, a scheme parameter changed, or someone applied a different classification to unchanged facts. (www.w3.org) (www.w3.org)

Implementation Considerations and Process Changes

Preserve history without prescribing a platform replacement

ORO.FTL.245 requires specified individual records and reports to be maintained for **24 months**. That requirement should not be represented as an explicit mandate for every proposed event field in this report. The operator should map mandatory records and its additional review evidence to a documented retention policy. (www.easa.europa.eu)

An integration may suffice when existing roster, notification or crew-application events need a combined view. Microsoft says traditional data management suffices for most systems. Begin with an evidence inventory and replay demonstration before selecting an architecture. (learn.microsoft.com)

Use a proposed export contract with defined relationships. CloudEvents supplies a common event envelope, but the operator still needs the crew-specific payload and semantics. PROV-O offers provenance concepts without obliging an implementation to use its full ontology. The interface should explain its fields in terms that scheduling and IT staff can both review. (cloudevents.io) (www.w3.org)

Acceptance requirements are:

- **Identity:** stable source-system and event references, plus the affected assignment. (github.com)
- **Chronology:** occurrence, receipt, publication and correction meanings documented separately. (cheatsheetseries.owasp.org)

- **Classification:** interval type and location retained before and after a change. (www.w3.org)
- **Notification:** communication outcome preserved without equating dispatch, delivery and acknowledgement.
- **Configuration:** scheme and calculation configuration identified in the result. (pharosproduction.com)
- **Correction:** an explicit replacement or correction relationship, with attribution. (www.w3.org) (www.w3.org)
- **Export:** machine-readable history that an operator can inspect outside the user interface. (www.w3.org)
- **Review:** named ownership for missing evidence and disputed classifications. (www.w3.org)

For identity, the official CloudEvents specification requires a unique combination of source and identifier and permits consumers to recognize repeated combinations as duplicates. A roster integration should document its equivalent convention. Reprocessing the same source event should not create a second operational call or a second crew acknowledgement. (github.com) (github.com)

Test transport behavior as well as calculation behavior

Delivery guarantees depend on the actual interface. Amazon Simple Queue Service (SQS) standard queues, for example, document possible duplicate delivery and occasional out-of-order arrival. That does not establish how any named roster product works. It does justify asking an integration supplier to demonstrate its own duplicate and ordering behavior explicitly. (docs.aws.amazon.com) (docs.aws.amazon.com)

The replay exercise should include the following proposed tests:

- **Duplicate import:** replay an identical event and inspect the resulting history. (github.com)
- **Late arrival:** receive the reporting event before an older notification record. (docs.aws.amazon.com)
- **Correction:** amend a source time while preserving the previously recorded value. (learn.microsoft.com)
- **Partial outage:** reconnect a crew device and distinguish event time from receipt time. (cheatsheetseries.owasp.org)
- **Re-export:** compare successive exports for stable identifiers and documented changes. (github.com)
- **Configuration change:** evaluate the same evidence under identified configuration versions. (learn.microsoft.com)
- **Access control:** demonstrate who can correct evidence and who can approve its interpretation. (www.w3.org)
- **Recovery:** rebuild the review view from the retained source evidence. (learn.microsoft.com)

These are synthetic acceptance inputs, not claims about observed vendor failures. Microsoft's guidance recommends event version identifiers. In this context, the test should verify how an older exported event is interpreted after a schema change, and whether the archived case remains understandable without a current production interface. (learn.microsoft.com)

Timestamp Interpretation and Export Tests

Request for Comments (RFC) 3339 permits numeric offsets, and its discussion of string ordering assumes consistent representations, including fractional-second precision. Sort normalized instants rather than assuming arbitrary timestamp strings produce a reliable chronology. Preserve the supplied value alongside the

normalized value when it is important to the review. (www.rfc-editor.org) (www.rfc-editor.org)

Database type selection needs explicit testing. PostgreSQL documents that a timestamp without time zone can ignore a supplied zone indication, while full zone names can carry daylight-saving rules. A migration test should therefore cover local-clock transitions, missing offsets and import conventions. It should demonstrate the chosen product's actual behavior, not assume that all database types interpret the same input identically. (www.postgresql.org) (www.postgresql.org)

Occurrence time should mean the event's claimed real-world time. CloudEvents defines a time attribute for occurrence, but makes it optional and permits a consistent substitute when occurrence time is unknown. Consequently, accepting a generic event envelope is insufficient: the operator should specify what each exported time actually means. (github.com) (cheatsheetseries.owasp.org)

UTC and local context also need separate treatment. RFC 3339 defines numeric offsets relative to Coordinated Universal Time (UTC). PostgreSQL's timestamp-with-time-zone storage does not retain the originally supplied zone. Preserve the original local representation and applicable zone separately when they are needed to explain a roster boundary; do not assume a UTC column can recover them. (www.rfc-editor.org) (www.postgresql.org)

The proposed round-trip test should start with an exported record, pass it through the intended integration, and compare the reconstructed operational instant and retained local context. Include a missing offset, a repeated local-clock value and a later correction. Mark ambiguous inputs for review rather than inventing a zone. Test stable ordering independently of how the user interface formats the timestamps. The documented offset and storage semantics explain why both checks matter. (www.rfc-editor.org) (www.postgresql.org)

Vendor Demonstration and API Acceptance

Public feature descriptions are starting evidence. Pharos's updated guide discusses storing verdicts against roster versions. FlightStratix advertises checks when assignments are entered. Those statements support a concrete demonstration request, but do not establish how every historical transition is stored or exposed under the operator's configuration. (pharosproduction.com) (flightstratix.com)

Leon documents automatic publication notifications, acknowledgement tracking and report data through its GraphQL application programming interface (API). AIMS describes notification lists and an option to mark crew as notified. The acceptance exercise should ask what those statuses mean, who or what sets them, and which associated times appear in the export. (wiki.leonsoftware.com) (wiki.leonsoftware.com) (www.aimsairlinesoftware.com)

A proposed vendor demonstration script is:

- **Load the case:** identify the hypothetical crew, applicable operation and scheme configuration.
- **Publish the window:** export the original assignment and publication evidence.
- **Create the transition:** revise the end, place the call, or change the subsequent assignment.
- **Inspect communication:** show message attempts, outcomes and acknowledgements actually retained.
- **Record reporting:** enter instructed and actual reporting independently.
- **Inspect the result:** identify input intervals and configuration behind the displayed outcome.

- **Correct one fact:** show the previous value and the reason for correction.
- **Export history:** produce records that reconstruct both the original and corrected case.
- **Replay independently:** rebuild the review view using the exported package.
- **Assign exceptions:** show the operational and technical owner of unresolved evidence.

A distinction between **aggregate reporting** and **event evidence** is important. Leon's Roster Robustness documentation says changes made within **five minutes** of the preceding update do not trigger that report's change count. That is a documented counting behavior, not proof that underlying audit records are missing. Ask for the underlying evidence rather than treating a report counter as a complete history.
(wiki.leonsoftware.com)

Likewise, an API claim does not establish a complete event schema. Leon's documented programmatic report access should prompt a field-level inspection. Ask the supplier to identify which endpoint, permission, date filter and historical-version mechanism actually returns the evidence needed by the matrix. A screen capture is useful context, but a repeatable export is the proposed acceptance artifact. (wiki.leonsoftware.com)

The decision can be recorded as **existing capability**, **configuration work**, **integration work** or **unresolved evidence requirement**. These are proposed project categories, not a market ranking. Record the evidence supporting the selected project category and the operational reviewer responsible for accepting it.
(learn.microsoft.com)

Data Analysis and Evidence

The quantitative core is deliberately limited to sourced rule parameters and reproducible evidence tests. It does not estimate fatigue from a roster counter. *Table 3* summarizes selected CS provisions that the authorized reviewer may need when interpreting the scenario package. All entries come from the inspected CS text; their applicability and interaction still require the approved scheme.

Parameter	Specified treatment or boundary	Evidence needed for review
Airport standby	FDP counts from its start; reduction for standby exceeding 4 hours ; combined 16-hour limit for the referenced basic/extended FDP arrangements. (www.easa.europa.eu)	Airport report, FDP start, arrangement used
Other standby	Maximum 16 hours ; 25% counts for ORO.FTL.210. (www.easa.europa.eu)	Type, actual interval, accounting purpose
Other-standby FDP reduction	6-hour threshold, extended to 8 hours for referenced in-flight-rest or split-duty cases. (www.easa.europa.eu)	Reporting and applicable extension
Night provision	For standby starting between 23:00 and 07:00 , specified reduction counting depends on contact. (www.easa.europa.eu)	Start, contact, relevant local-time context
Awake-time procedure	Design combination to avoid more than 18 hours awake . (www.easa.europa.eu)	Procedure and operational review, not an assumed sleep measurement
Reserve	No duty counting for ORO.FTL.210/235; roster 8 hours without contact on each reserve day to protect sleep opportunity. (www.easa.europa.eu)	Reserve interval and protected window

The table separates clock boundaries from evidence requirements. It does not make the referenced airport combined-duration rule universal across every FDP arrangement, or turn standby plus FDP into observed wakefulness. The CS source is used here as a concise parameter inventory rather than a ready-to-dispatch calculator.

Hypothetical Example: elapsed interval. Suppose the notified standby start is **08:00**, a revised-end notification is recorded at **11:30**, and the effective end is also **11:30**. The elapsed interval is **3 hours 30 minutes**, by direct subtraction. That arithmetic is illustrative, not a legal verdict. The evidence test is whether the export preserves the original end, both notification and effective-end meanings, and any later correction. OWASP's event-versus-log distinction explains why the server's receipt time should not silently replace the operational event time. (cheatsheetseries.owasp.org)

A quantitative acceptance report should publish denominators rather than unsupported performance claims. Proposed measures include:

- **Reconstructable cases:** cases whose required evidence is available, divided by cases executed.
- **Missing fields:** absent required evidence items, grouped by scenario and source system. (cheatsheetseries.owasp.org)
- **Replay agreement:** cases reproducing the approved expected interpretation, divided by cases reviewed. (learn.microsoft.com)
- **Duplicate stability:** repeated imports that leave the interpreted event count unchanged. (github.com)
- **Correction traceability:** corrected cases retaining the original fact and responsible editor. (learn.microsoft.com) (www.w3.org)

These are proposed project measurements. No vendor benchmark or acceptance percentage is asserted. Microsoft's replay pattern and CloudEvents' duplicate-identification convention explain what the tests exercise, but the operator must approve the expected interpretations and acceptance criteria. (learn.microsoft.com) (github.com)

Original aviation research also distinguishes duty data from sleep observations. A controlled-rest study used actiwatches and sleep logs; a separate roster survey collected reported duty, standby and rest information. Those are different evidence types. Neither justifies inventing an individual crew member's sleep from an unassigned period on the roster. (www.frontiersin.org) (commons.erau.edu)

Implications and Future Directions

The useful procurement question is **which evidence is recoverable**, not merely whether a supplier advertises an FTL engine. FlightStratix's documented engine and AIMS's notification features describe relevant functions, while Pharos explicitly discusses version-linked results. A fair evaluation should acknowledge those published capabilities and then test the operator's actual scenarios without assuming undocumented support or absence. (flightstratix.com) (www.aimsairlinesoftware.com) (pharosproduction.com)

A small integration may be appropriate when an existing product holds the events but the review workflow cannot join them. LANDING.AERO's stated scope includes integrating disparate data sources; that represents a custom implementation category, not a substitute for a crew-roster product or competent-authority approval.

Any project should begin with the source inventory, export sample and approved review responsibilities. (www.landing.aero)

Future interface improvements should emphasize explicit meanings rather than additional generic timestamps. CloudEvents standardizes a common description; PROV-O supports interchange of provenance. Those concepts can help operators specify portable evidence without claiming that adoption of either standard makes a system compliant. Suppliers can demonstrate the result through a historical case rather than architecture slides. (cloudevents.io) (www.w3.org)

Operator ownership should include both technical interpretation and crew experience. Original qualitative aviation research identifies value in pilots' experience within fatigue risk management. For this project, that supports including operational users when reviewing whether a notification history is intelligible, while keeping regulatory interpretation and scheme ownership with the authorized functions. It does not establish an effectiveness benchmark for the proposed integration. (journals.plos.org)

The practical next deliverable is an agreed evidence package and a signed acceptance record. PROV-O's attribution concept is useful here: the package should identify who supplied a source fact, who corrected it and who accepted an interpretation. Future corrections should supplement that record rather than make the earlier review disappear. (www.w3.org) (learn.microsoft.com)

Frequently Asked Questions (FAQs)

Is reserve simply home standby with a different label?

No. Reserve's definition provides advance notification of an assigned duty of at least **10 hours**. The distinction matters even if both periods occur away from the airport. The export should identify the classification actually rostered and the notification evidence, rather than infer the category from location alone. (eur-lex.europa.eu)

Does a visible notification status prove every relevant timestamp exists?

No such conclusion follows from a feature label. AIMS documents a notified status option, and Leon documents acknowledgement tracking. The proposed acceptance test asks for the underlying meaning, author and timestamps. It should record what the product demonstrates instead of assuming an API exposes every detail visible in the interface. (www.aimsairlinesoftware.com) (wiki.leonsoftware.com)

What should a timed transition and its correction export show?

Hypothetical revised-end acceptance test: all times below are UTC on the same day. Publish a standby window of **08:00–14:00**. Initially record revised-end notification and effective cessation at **11:30**, received by the system at **11:32**. At **12:00**, add a correction stating that both notification and cessation occurred at **11:25**. These are synthetic test inputs, not an operational or legal verdict.

The proposed export should contain:

Exported view	Expected interval	History retained
Original published window	08:00–14:00, planned only	Original publication reference

Exported view	Expected interval	History retained
Before correction	08:00–11:30, elapsed 3 hours 30 minutes	Notification and cessation at 11:30; receipt at 11:32
After correction	08:00–11:25, elapsed 3 hours 25 minutes	Correction recorded at 12:00, linked to the original record

Require the correction to retain the previous values, reason, source reference and responsible editor. Replay the before-correction and after-correction views separately; keep the planned window identifiable in both. (learn.microsoft.com)

Conclusion

A useful standby and reserve evaluation begins with the question the reviewer must answer, then works backwards to the evidence. The matrix focuses on revised ends, late calls, changes to airport duty and reserve transitions because they expose the difference between a final roster and a reconstructable history. The proposed deliverable is an explainable case package, not an additional generic counter.

Keep meanings separate: planned, notified, actual and corrected times; communication outcomes; interval classifications; source facts; and reviewer interpretation. General provenance and logging guidance support those separations, while the operator defines the concrete payload and acceptance process. Neither a common event format nor a database pattern replaces the approved scheme. (www.w3.org) (cheatsheetseries.owasp.org)

Require a demonstration: create the scenario, inspect the result, export the evidence, correct a fact and replay the case. Judge published vendor functions against what is actually demonstrated in the configured environment. Where the necessary history exists, configuration or integration may be enough. Where evidence is unavailable, record the unresolved requirement and its owner before accepting the workflow. (learn.microsoft.com)

The operator remains responsible for the operational interpretation. Scheduling, operations control, safety and IT should leave the acceptance exercise with a shared record of the evidence preserved, the assumptions applied and the exceptions requiring authorized review. That makes the software decision concrete while keeping regulatory and operational authority in the appropriate hands.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-air-operations?erules-id=ERULES-1963177438-11941>
2. <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32014R0083>
3. <https://www.easa.europa.eu/en/faq/47642>
4. <https://www.easa.europa.eu/en/faq/47644>
5. <https://www.w3.org/TR/prov-o/>
6. <https://cloudevents.io/>

7. <https://learn.microsoft.com/en-us/azure/architecture/patterns/event-sourcing>
8. <https://flightstratix.com/crew-management>
9. <https://wiki.leonsoftware.com/wiki/how-leon-modules-collaborate-for-efficient-flight-operations>
10. <https://www.aimsairlinessoftware.com/crew-management>
11. <https://pharosproduction.com/insights/engineering/crew-rostering-software/>
12. <https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-air-operations>
13. <https://www.easa.europa.eu/en/regulations/air-operations>
14. <https://www.easa.europa.eu/en/faq/47556>
15. <https://www.landing.aero/>
16. <https://www.easa.europa.eu/en/faq/47607>
17. https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html
18. <https://github.com/cloudevents/spec/blob/main/cloudevents/spec.md>
19. <https://www.easa.europa.eu/en/faq/47608>
20. <https://www.easa.europa.eu/en/faq/47643>
21. <https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-air-operations?erules-id=ERULES-1963177438-12213>
22. <https://www.easa.europa.eu/en/faq/47645>
23. <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/standard-queues.html>
24. <https://www.rfc-editor.org/rfc/rfc3339>
25. <https://www.postgresql.org/docs/current/datatype-datetime.html>
26. <https://wiki.leonsoftware.com/wiki/report-roster-robustness>
27. <https://www.frontiersin.org/journals/environmental-health/articles/10.3389/fenvh.2024.1368628/full>
28. <https://commons.erau.edu/ijaaa/vol8/iss3/5/>
29. <https://www.landing.aero/about/>
30. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0195530>

THE PUBLISHER

About LANDING.AERO

LANDING.AERO builds custom software for aviation operations teams at airlines, charter operators and flight departments. It is based in Montreal.

What it builds

Operations tools that reduce manual steps in day-to-day flight operations; integration of existing systems and data sources into one operational picture; workflow automation; real-time operational dashboards; bridges between legacy systems and modern applications; and mobile apps for staff in the field. Each application is designed around the operator's own workflows rather than sold as an off-the-shelf platform.

How it works

Work starts with discovery of the operator's workflows, continues with iterative development alongside the operator's team, and includes integration with the tools already in use and ongoing support.

Its own tools

LANDING.AERO also publishes small tools for aviation professionals, including UTC Alarm, which sets alarms in Zulu time, and NOTAM Notifier, announced for NOTAM change alerts on chosen airports.

Visit

See LANDING.AERO and its projects.

Articles published here explain general practice and cite their primary sources, naming the jurisdiction of any rule they describe. They are not operational, regulatory or legal advice, do not replace an operator's approved procedures or manuals, and requirements change.

Website: <https://www.landing.aero>

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.