LANDING.AERO

LANDING.AERO / RESEARCH & PRACTICAL GUIDANCE

How to Reconcile OOOI Events with ACARS and Flight Schedules

> 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 · 2026-10-01 · oooi events · acars · flight schedules · flight operations data · block time · flight status · data reconciliation · aidx · operations control

Original article: https://www.landing.aero/articles/en/oooi-event-integration-acars-flight-times

In this article

1. Executive Summary
2. Introduction and Background
3. OOOI Events and Time Types
4. ACARS and Flight Status Feed Semantics
5. Build the Shared Flight Event Record
6. Reconciliation Policy and Exception Handling
7. Exception Handling and Audit Ownership
8. Implementation and Handoff to Operational Tools
9. Data Analysis and Evidence
10. Implications and Future Directions
11. Frequently Asked Questions (FAQs)
12. Conclusion

Executive Summary

Reconciling **Out, Off, On, In (OOOI)** events means maintaining one flight leg with several distinct time claims: a published schedule, a current estimate, and a reported actual event. The United States Federal Aviation Administration (FAA) defines the four movements as gate out, wheels off, wheels on and gate in (www.aspm.faa.gov). Aircraft Communications Addressing and Reporting System (ACARS) messages can carry OOOI reports, but a feed's arrival at an operations system does not by itself establish that each displayed time was measured on the aircraft. The FAA's Aviation System Performance Metrics (ASPM) dataset explicitly estimates unavailable OOOI times (www.aspm.faa.gov), while a commercial flight status interface may return a mixture of scheduled, estimated and actual fields (developer.flightstats.com). The practical rule is to **retain the source and status of every timestamp**, then make the choice of display value a documented policy.

A shared record should identify the operating flight leg, station, event type, event time in Coordinated Universal Time (UTC), original source payload, received time, source revision, and quality state. Published schedules and operational estimates remain separate from reported actuals; they should not be overwritten when an event arrives. The International Air Transport Association (IATA) describes Aviation Information Data Exchange (AIDX) as a standard for operational flight-data exchange (www.iata.org), and its schedules material treats UTC and local-time comparison as a distinct concern (www.iata.org). A matching key uses operating carrier, flight number, origin date and leg endpoints, with handling for codeshares, diversions, number reuse and source repeat number when supplied. **Do not match on flight number alone.**

Use a configurable source policy only after comparing each feed's event definition, timestamp convention, coverage, rights and correction process. The FAA's ASPM hierarchy gives Airline Service Quality Performance (ASQP) data priority, followed by Official Airline Guide (OAG) and Traffic Flow Management System (TFMS) OOOI for that analytical product (www.aspm.faa.gov); it is **not** a universal carrier policy. Choose a candidate actual only when flight identity, event semantics and chronology are plausible. Otherwise display a clearly

labeled estimate or a missing state, retain the disputed inputs, and send the record to an owner. A vendor display's actual, estimated, scheduled precedence (developer.flightstats.com) is a useful example of a presentation rule, not proof that every upstream time has identical provenance.

Measured block time is **gate in minus gate out** (www.aspm.faa.gov) and measured airborne time is **wheels on minus wheels off** (www.aspm.faa.gov). Compute either only after the pair refers to the same valid leg and compatible event sources. These intervals do not establish flight-duty compliance. The report provides a source map, five synthetic exception tests, data-quality measures and procurement questions. Public U.S. and European punctuality figures show the scale of time-based analysis, but neither is a benchmark of ACARS reconciliation accuracy (www.transtats.bts.gov) (www.eurocontrol.int).

Introduction and Background

An operations control center (OCC), crew desk, performance analyst and customer-facing service may all ask for the same flight's departure or arrival time while holding different answers. One system has the published departure, another has an updated estimated off-block time, and a third has an ACARS report. A later correction can alter the value again. EUROCONTROL's network tools, for example, retain an estimated off-block field as a separately received value (www.nm.eurocontrol.int), and their history view records the status resulting from an event (www.nm.eurocontrol.int). The integration question is which claim should be displayed, with what label, when those feeds disagree.

In its Part 234 on-time reporting directive, the United States Bureau of Transportation Statistics (BTS) says reporting carriers using automated systems rely on the Aircraft Communication Addressing and Reporting System (ACARS), Docking Guidance System (DGS) or Airborne Flight Information System (AFIS) (www.bts.gov). The source can affect the event trigger: BTS Directive 40 gives an example of one DGS recording gate departure when the aircraft moves more than 1 meter from the parking mark within 15 seconds (www.bts.gov). That is why an operator must confirm its own aircraft, airport and vendor definitions before treating two similarly named fields as interchangeable. The decision also depends on the recipient. A crew application may need the current operational view; a performance dataset needs a reproducible final value; a passenger display may need a concise status without the full audit trail.

This report defines a reconciliation design for ground systems. It does not prescribe dispatch actions or calculate regulatory flight and duty time. The carrier or charter operator must apply its approved procedures and applicable contracts when deciding how a derived time can be used. **LANDING.AERO** describes custom flight-operations integration and dashboards in its own materials (www.landing.aero) (www.landing.aero). That makes a custom integration one possible implementation route when existing systems cannot preserve the required provenance. The event definitions and policy choices below apply regardless of who builds the software.

OOOI Events and Time Types

Four movements, several time claims

Out is the departure from the gate or parking position; **Off** is wheels off the runway; **On** is wheels on at landing; **In** is arrival at the gate or parking position. The FAA groups these as actual aircraft movements (www.aspm.faa.gov). The names sound simple, yet a data feed can present a planned, estimated, observed or derived value for any one of them. A published schedule sets a baseline. A flight-plan estimate describes an expected operational milestone. A reported actual asserts that the movement occurred. A derived value is computed from other observations or a model. These types must remain visible in the data model even if an interface combines them into one prominent clock time.

The FAA's flight-level dictionary renders scheduled gate departure both as GMT seconds since January 1, 1980 and as local clock time (www.aspm.faa.gov) (www.aspm.faa.gov). Thus a string such as `23:55` is insufficient for reconciliation. It needs a date, station, time-zone interpretation and a type. BTS's U.S. on-time-reporting format uses local scheduled and actual clock times (www.bts.gov), while IANA warns that an alphabetic time-zone abbreviation is not a unique identifier (www.iana.org). Store an unambiguous UTC instant and the original supplied value, then render the local time using the station and the applicable historical rules.

ACARS and Flight Status Feed Semantics

ACARS is a transport and message family

An ACARS message can convey an OOOI event, but the exact trigger and format must be verified for the operator's fleet and feed contract. SITA describes processing that parses and distributes ACARS messages according to predefined rules (www.sita.aero); it also says the messages need integration into an organization's infrastructure to be useful (www.sita.aero). A message timestamp may mean event occurrence, aircraft creation, ground processing or receipt. These are different clocks. An integration must preserve the raw message and map its fields to an explicitly named event time and receipt time. If the mapping is uncertain, mark the result as unresolved rather than silently promoting it to an actual.

An external flight-status feed can be valuable, but it introduces another semantic layer. Cirium's documented flight display uses actual, estimated and scheduled precedence (developer.flightstats.com), while its flight-status documentation cautions that a flight need not have every gate or runway time (developer.cirium.com). The presence of a vendor's `actual` field should be interpreted according to that interface's documentation and the operator's data agreement, not assumed to have arrived directly from the aircraft. A feed update may also arrive after the flight, so **event time and received time must be separate fields**.

Table 1 maps candidate sources to the claims they can make. It is an integration checklist, not a universal ranking.

Source	Possible event claim	Keep alongside the value	Typical display treatment
Published schedule	Planned gate out and gate in; a local departure may be the airline's published time (developer.flightstats.com)	Schedule version, origin date, local zone and publisher	Baseline marked **scheduled**, even after an actual arrives
Flight plan or operational estimate	Expected off-block, takeoff or arrival milestone; EUROCONTROL tracks received estimates separately (www.nm.eurocontrol.int)	Producer, issue time, intended milestone and revision	**Estimated**, with freshness visible
ACARS OOOI report	Candidate aircraft-reported Out, Off, On or In; the parser's rules matter (www.sita.aero)	Raw payload, aircraft and flight identity, event and receipt clocks	**Reported actual** after mapping and validation
Airport or third-party status feed	Gate or runway event or an estimate; completeness varies by flight (developer.cirium.com)	Original field name, provider, rights and update identifier	Actual or estimated only as defined by the feed
Derived or inferred record	Modelled event when observation is absent	Inputs, algorithm version, confidence and reason	**Estimated or inferred**, never an unqualified actual

The distinction matters because a consumer might otherwise compute a duration from a schedule at one end and a measured event at the other. Table 1 also makes the vendor handoff testable: the operator can ask the feed owner for the event definition, source clock and correction rules for every column it receives.

Build the Shared Flight Event Record

Identity before timestamp selection

The reconciliation unit is a **flight leg**, not a flight number printed on a screen. Keep operating carrier, flight number and suffix, scheduled origin date, departure and arrival stations, source repeat number when supplied, and an internal immutable leg identifier. EUROCONTROL's schedule upload distinguishes commercial and air traffic control operator designators (www.nm.eurocontrol.int). IATA says airline designators appear in schedules and telecommunications (www.iata.org). IATA's AIDX guide defines a commercial flight-leg identifier from several fields (www.iata.org). These examples illustrate why the matching layer needs a mapping table, not a single string comparison.

The **origin date** should be defined by the source contract; AIDX defines it as the UTC scheduled date of departure of a flight. A flight departing near midnight may have a local operation date that differs from the UTC date, and a multi-leg operation may retain a common marketing number. For a multi-leg flight, IATA's AIDX guide assigns every leg the scheduled departure date of the first leg as its origin date (www.iata.org), while the FAA's flight export stores time instants in GMT seconds (www.aspm.faa.gov). Preserve the source's date convention, normalize it once, and retain the mapping decision. Location codes also need an effective date; BTS notes that airport codes can change or be reused (www.transtats.bts.gov). Do not rewrite historical legs merely because a current code lookup differs.

Minimum event schema

Each candidate event should be append-only at ingestion. The materialized flight view can change, but the inputs remain available for review. A compact schema contains the following fields:

- **Identity:** internal leg ID, operating identifier, flight number, scheduled origin date, leg endpoints, source repeat number when supplied, and station for the event.
- **Event:** Out, Off, On or In; original field name; source's event definition and candidate UTC instant.
- **Time context:** original text or epoch value, declared time zone or offset, precision, event clock, received clock and processing clock.
- **Provenance:** source system, message ID, raw payload reference, transformation version and provider update identifier. W3C's provenance model supports exchanging provenance across systems (www.w3.org).
- **Revision:** superseded candidate, source revision, correction reason, effective time and operator who approved a manual override.
- **Quality:** scheduled, estimated, reported, verified, inferred, disputed or missing; validation result; freshness and confidence, if the source supplies them.

This schema avoids conflating a correction with a duplicate. Cirium documents update records with a time when a change was applied and a source description (developer.flightstats.com) (developer.flightstats.com). An event store can use those fields to reconstruct what each consumer knew at a particular moment. AWS cautions that event IDs can change across API calls (docs.aws.amazon.com), so a transport-generated ID alone is a weak cross-system identity. The internal leg and candidate IDs should be stable across retries and replays.

Reconciliation Policy and Exception Handling

Candidate selection as a documented rule

Start by validating **identity**, **event meaning**, **time conversion**, **sequence** and **source authority**. An operator may choose its own trusted direct ACARS feed for a verified Out event, accept an airport status report for a missing In event, and use an estimate only for a clearly marked live display. That order is a design example, not a general aviation rule. The FAA's ASPM chooses ASQP OOOI, then OAG OOOI, then TFMS for its analytical dataset (www.aspm.faa.gov) and estimates when input is absent (www.aspm.faa.gov). Copying that hierarchy into a carrier's live OCC would ignore different contracts, latency and event definitions.

Apply the candidate checks in a fixed order so each rejection has a reason:

- **Leg match:** resolve the operating identifier, scheduled origin date, leg endpoints and source repeat number when supplied.
- **Station match:** confirm the event belongs to the correct endpoint.
- **Event mapping:** confirm the field means Out, Off, On or In.
- **Clock basis:** convert the original value to a UTC instant.
- **Chronology:** compare it with other accepted milestones and exceptions.
- **Authority:** apply the approved source policy for this consumer.

A **display value** should be derived from the candidate set, not written over the candidate set. An actual can replace an estimate on the live board without deleting the prior estimate. Cirium's flight-information display response documents actual, then estimated, then scheduled priority (developer.flightstats.com). That is an example of a user-interface convention. The system still needs to show a provenance badge and the time last checked. For a disputed actual, a policy may temporarily retain the last uncontested estimate and raise a review task. It should not force a mathematically tidy event sequence by editing a raw reported time.

Exception Handling and Audit Ownership

The following cases are **synthetic tests** for an integration. All times are invented UTC values on a single illustrative leg unless a row states otherwise. Each row is labeled to prevent the test input from being read as observed operational data.

Table 2 shows the minimum decision, badge and owner for five exception paths.

Synthetic input	Chosen display state	Provenance badge	Escalation owner
Duplicate (Hypothetical Example): two identical Out payloads with the same source ID, repeat number 1 and 10:04 event time; a separate Out payload carries repeat number 2	Deduplicate the repeat-1 delivery, retain both receipts and hold repeat 2 as a distinct attempt	Reported actual, duplicate delivery; separate repeat attempt	Integration operations
Late correction (Hypothetical Example): In 14:22 arrives after a provisional 14:26 estimate, then source revises In to 14:24	Show 14:24 after validated revision; preserve both earlier claims	Reported actual, revised	OCC data steward
Missing event (Hypothetical Example): Off 10:18 exists, Out is absent	Show Off as reported; Out remains missing or explicitly estimated under policy	Reported Off; missing or inferred Out	OCC controller
Mismatched identity (Hypothetical Example): On 12:49 carries a different operating identifier from the scheduled leg	Hold On outside the leg pending match review	Disputed identity	Flight-data analyst
Overnight rollover (Hypothetical Example): local Out 23:55 and local In 01:10 on the next calendar date	Convert both at their stations to UTC before ordering or duration calculation	Reported actual, date normalized	Integration operations

Table 2 shows why **received order** cannot serve as event order. IBM documents that a message pipeline using batching and retries can duplicate or reorder messages (cloud.ibm.com). The rule engine should make deduplication idempotent, preserve both arrival and occurrence timestamps, and allow a later source correction to revise the materialized view. A missing event must remain visible as missing even if a derived estimate helps a live screen. A mismatch should not be resolved solely by proximity in time: the wrong flight can be nearby at the same airport.

Chronology and review

The expected progression is Out, Off, On, In, but a rule should allow known operational exceptions and corrections. In U.S. Part 234 reporting, BTS instructs carriers handling a gate return to report the last gate departure before wheels off, and to preserve the first departure in another field (www.bts.gov). That reporting convention is jurisdiction and program specific; the broader design lesson is to retain all movements and define which one supplies a particular report. Do not erase a first Out simply because a later Out becomes the selected block-time start for one use case.

An exception queue needs the input payload, source clocks, selected state, reason code and accountable owner. It should show the effect on every downstream view before a manual override is accepted. Keep an audit trail of the prior selected candidate and the operator's rationale. A network flight-history interface that records event and resulting status (www.nm.eurocontrol.int) is a useful pattern for this traceability. Retention periods and redistribution rights are contract and jurisdiction questions, so the implementation team must confirm them from its own agreements rather than infer them from an API example.

The queue should make each review answerable without a separate log search:

- **Evidence:** raw payload and provider event ID.
- **Clock trail:** supplied event time, receipt time and converted UTC.
- **Identity trail:** candidate leg, competing legs and match rationale.
- **Policy trail:** selection version and failed validation rule.
- **Ownership:** named team and due state for the review.
- **Outcome:** approved correction, rejected candidate or unresolved status.

Implementation and Handoff to Operational Tools

Pipeline and ownership

Separate **ingestion**, **normalization**, **matching**, **selection** and **distribution**. Ingestion stores original messages and provider acknowledgments. Normalization converts clocks and field names while preserving source values. Matching resolves a candidate to one leg or an exception. Selection applies a versioned policy. Distribution sends the current state and provenance badge to OCC, crew, performance and customer-facing systems. The receiver should be able to request the selected state plus its supporting candidate IDs and policy version, so two screens can explain a discrepancy instead of merely showing different clocks.

Each recipient should get a contract shaped for its decision:

- **OCC board:** current selected time, state, source and freshness.
- **Crew desk:** validated event and correction history for its own rules.
- **Performance store:** reproducible snapshot and later revision links.
- **Customer display:** concise status with estimate versus actual labeling.
- **Data steward:** rejected candidates and open identity exceptions.
- **Integration support:** delivery lag, retry history and failed mappings.

Adapters require source-specific tests. Cirium describes a feed as a stream of changes rather than a lookup service (developer.flightstats.com). A connector must know whether it receives snapshots, deltas, or both; how it catches up after downtime; and whether a revision replaces a field or appends a new event. FlightAware asks API buyers to check usage and storage restrictions by tier (www.flightaware.com). That review belongs in procurement and the data contract, because an event archive may exceed a feed's permitted use even when the API technically returns the field.

Consumers should subscribe to the **meaning** of a change, not just a new timestamp. An OCC alert may care that an actual In was revised. Crew scheduling may need a validated actual with the source and correction time, but any duty-time or pay calculation belongs to its own approved rule set. Performance analytics may freeze a reporting snapshot, then accept later corrections in a new version. A passenger display may show a simplified current state but should not turn an inferred arrival into an unqualified actual. EUROCONTROL's airport collaborative decision-making specification connects milestone updates to downstream estimates and notifications (www.eurocontrol.int). That is evidence for treating handoff as an event lifecycle rather than a one-time file import.

Build-versus-buy questions

An operator with an existing dispatch or crew system should test its event semantics before commissioning a replacement. The comparison is between an integration layer, configurable components and a platform's native capability. **LANDING.AERO** positions its work as tailored data integration and operational dashboards (www.landing.aero) (www.landing.aero); that makes it a custom-build option for this particular integration problem, not an off-the-shelf dispatch or crew-scheduling product.

Table 3 compares implementation routes. The rows describe delivery models rather than equivalent products or prices.

Route	What the operator should verify	Fit for OOOI reconciliation
Existing dispatch or crew system configuration	Native event fields, source labels, correction history and export rights	Useful when the installed system already preserves candidate-level provenance
Integration middleware or a flight-status service	Event definitions, feed permissions, replay, and the boundary between a status feed and the carrier's own record	Useful when several systems need a shared transport or external status inputs
LANDING.AERO custom integration	Scope of a tailored data integration and dashboard build (www.landing.aero) (www.landing.aero); operator ownership of policy and source contracts	A build-to-order route when existing systems cannot supply the needed event ledger

The table makes an operational distinction: a status feed supplies input, whereas the operator still needs a policy and accountable owner for the shared record. Evaluate any builder or vendor against the same acceptance cases:

- **Evidence:** Can users inspect the original message, field definition, source clock and correction history?
- **Identity:** Can the product separate operating and commercial identifiers, reused flight numbers and overnight legs?

- **Policy:** Can selection rules differ by fleet, station, provider and downstream use without changing raw records?
- **Recovery:** Can it replay missed messages and deduplicate retries without inventing an event?
- **Rights:** Does the agreement permit the intended storage, redistribution and analytics use?
- **Operations:** Who owns exception review, policy changes, monitoring and rollback?

An ADS-B history source alone is not a schedule or gate-event system. OpenSky says its API lacks commercial schedules (openskynetwork.github.io). A tracking product might help corroborate airborne phases, but gate Out and In still require a source whose semantics support those events. This is a testable boundary for a procurement demonstration.

Data Analysis and Evidence

The quantitative question for a reconciliation project is **how often the selected operational record is complete, timely and defensible**. No public primary source in this research supplied a cross-operator benchmark for ACARS OOOI delivery latency, event completeness or false match rate. Those measures must come from a pilot using the operator's own messages and schedules. The FAA and BTS publish definitions and large reporting datasets, but those are not substitutes for an operator's feed-quality measurement. ICAO says basic performance indicators can combine ACARS OOOI with airline schedule times (ganpportal.icao.int), which reinforces the need to preserve both sides of the comparison.

Public punctuality numbers show why the distinction matters at scale, while measuring a different thing. BTS's **2025 annual** marketing-carrier table lists **7,736,770 flight operations** and **76.42%** on-time arrivals in its reported population (www.transtats.bts.gov) (www.transtats.bts.gov). BTS defines an on-time gate arrival or departure using a threshold of **less than 15 minutes** after schedule (www.bts.gov). EUROCONTROL reported **72.4%** European arrival punctuality within 15 minutes for **2024** (www.eurocontrol.int) and an average **17.5-minute** departure delay per flight (www.eurocontrol.int). These populations, geographies, years and methods differ. None of those percentages estimates the probability that an ACARS event is correct, and they should not be used to score a feed provider.

Use a small pilot scorecard with explicit denominators. These formulas are design proposals, not published industry benchmarks:

- **Candidate coverage:** legs with each named OOOI candidate divided by eligible legs.
- **Verified actual coverage:** eligible legs with an identity-matched, valid actual divided by eligible legs.
- **Arrival latency:** receipt UTC minus event UTC, reported by source and event percentile.
- **Revision rate:** selected actuals later changed by their source divided by selected actuals.
- **Identity exception rate:** candidates held for ambiguous or mismatched legs divided by received candidates.
- **Mixed-provenance duration rate:** displayed intervals with incompatible endpoints divided by all displayed intervals.

The scorecard is useful only if the measurement design is stable. The pilot should publish a data dictionary, record missing values separately from zero latency, and compare the same eligible population across versions. Reports also have different release cadences: BTS requires each reporting air carrier to file Form 234 for each calendar month (www.bts.gov), and Eurostat says EUROCONTROL-sourced monthly commercial-flight counts are published **10 days** after the reference month (ec.europa.eu). A live OCC feed, a monthly regulatory report and an analyst's historical extract are therefore different products even when they share flight identifiers.

For each leg, compute **measured block time = In UTC minus Out UTC** only if both actual candidates are accepted for that leg; the FAA defines actual block time by those endpoints (www.aspm.faa.gov). Compute **measured airborne time = On UTC minus Off UTC** under the same condition; the FAA uses those endpoints for airborne time (www.aspm.faa.gov). If one endpoint is inferred, name the result `estimated block` or `estimated airborne`, record its method and keep it out of a measured-only metric. A calculated interval that looks plausible is not evidence that its input events were observed.

Implications and Future Directions

The first operational benefit is a common explanation for why two tools show different times. One may display the schedule; another may show a fresh estimate; a third may have a validated actual. A shared event service can publish all three claims and a selected view with a short provenance badge. IATA's AIDX standard exists to exchange operational flight information (www.iata.org), but interface conformance alone cannot determine the operator's local selection rule. The policy must be versioned, reviewed and tested with representative exceptions.

The next improvement is to make **correction behavior** a first-class requirement. A source may revise an actual after an initial status notification, and downstream systems may have already acted on the earlier value. Cirium documents an update time and source description in its status response (developer.flightstats.com) (developer.flightstats.com). Those fields support a replayable history and an explicit correction notice. Store the state each recipient received, so an analyst can reproduce a report as it stood on a prior date while still incorporating later corrections in the current view.

Finally, procurement should distinguish **data access** from **data authority**. A provider may expose scheduled and actual block fields (www.flightaware.com), yet the contract can limit retention or onward distribution (www.flightaware.com). The operator should document event semantics, permission scope, update guarantees and fallback ownership before building dashboards around a feed. Time-zone rules and identifier tables also change; IANA maintains historical local-time rules (www.iana.org), and BTS warns that airport codes may be reused (www.transtats.bts.gov). The design must support controlled reference-data updates without rewriting the provenance of past flights.

Frequently Asked Questions (FAQs)

What are OOOI events in aviation?

They are the four actual movement milestones defined above: **Out** from a gate or parking position, **Off** the runway, **On** the runway, and **In** at a gate or parking position. A schedule or estimated time for one milestone is a different type of claim and should carry its own label.

Are OOOI times the same as scheduled flight times?

No. Scheduled times are a planned baseline, while a reported OOOI event asserts that a movement occurred. The FAA gives scheduled gate departure a distinct field (www.aspm.faa.gov). A current estimated time is a third category. The quality flag must reveal when an absent observation has been filled with an estimate.

How should ACARS OOOI data be integrated?

Capture the original message, determine its aircraft and leg identity, map each field to a defined event, convert its event instant to UTC, and record receipt time and source revision. Then validate and publish the selected state with provenance. SITA describes ACARS parsing and distribution by defined rules (www.sita.aero). The operator should verify those rules against its own aircraft and ground provider, because a generic field name does not prove a universal trigger.

How are actual and estimated flight times reconciled?

Keep both values, choose the live display according to a versioned policy, and label the chosen value. A verified actual can displace an estimate on a live board while the prior estimate remains in the audit record. A documented vendor display uses actual, then estimated, then scheduled precedence (developer.flightstats.com). That is one interface convention, not a mandate for every operational system.

What is the main OOOI data-quality test?

Confirm that a timestamp belongs to the correct leg and means the claimed event. Then check UTC conversion, chronology, duplicates, late revisions and whether the event was observed or inferred. A provider itself cautions that a flight may lack some gate or runway fields (developer.cirium.com). Missing data should be measured and labeled, not hidden by copying a scheduled value into an actual field.

Can a block-time value establish crew duty compliance?

No. The arithmetic interval defined above does not by itself determine a duty or pay outcome. Any such determination requires the operator's applicable rules, procedures and validated input definitions. A duration computed from a mixed schedule and actual pair is especially unsuitable for that purpose. The reconciliation service should pass the underlying events and quality states to the authorized crew system.

Conclusion

The reliable answer to ACARS-versus-schedule disagreement is an event ledger with a clear selection rule. Preserve planned, estimated, reported and inferred values as separate claims. Match each candidate to the operating leg before comparing times; retain the original source, UTC event instant, receipt instant, revision and quality state. Convert to local time for presentation while keeping the UTC instant as the arithmetic reference.

An operator should define its own hierarchy for each event and recipient after validating source semantics and rights. The FAA's analytical hierarchy illustrates why such a policy exists, but it is specific to ASPM. A missing event should remain identifiable as missing or inferred; a disputed event should enter a review queue. Late messages and corrections should update the selected view without erasing the earlier evidence.

The practical acceptance test is whether OCC, crew, performance and customer-facing tools can explain the value they display. They should be able to show which candidate won, why it won, when it was received and whether an estimate was used. If they can, block and airborne intervals become reproducible and exceptions become actionable. If they cannot, a single attractive timestamp on a dashboard may mask several incompatible time claims and lead to inconsistent downstream records.

Source links

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

1. https://www.aspm.faa.gov/aspmhelp/index/OOOI_Data.html
2. https://developer.flightstats.com/api-docs/fids/v1/fidsresponse
3. https://www.iata.org/en/publications/info-data-exchange/
4. https://www.iata.org/en/publications/manuals/standard-schedules-information/
5. https://www.aspm.faa.gov/aspmhelp/index/ASPM__Individual_Flights__Definitions_of_Variables.html
6. https://www.transtats.bts.gov/Marketing_Monthly.aspx?5ry_Nv42146=DDD&5ry_Pn44vr4=DDD&5ry_lrn4=FDFI&N44_Qry=E&heY_fryrp6Z106u=H&heY_fryrp6lrn4=FDFI
7. https://www.eurocontrol.int/publication/all-causes-delays-air-transport-europe-annual-2024
8. https://www.nm.eurocontrol.int/HELP/Details.html
9. https://www.nm.eurocontrol.int/HELP/Flight_History.html
10. https://www.bts.gov/explore-topics-and-geography/modes/aviation/number-40-technical-directive-reporting-time
11. https://www.landing.aero/
12. https://www.landing.aero/about/
13. https://www.aspm.faa.gov/aspmhelp/index/ASPM_Data_Download__Flight_Level_Data.html
14. https://www.bts.gov/topics/airlines-and-airports/%E2%80%A2-number-27-technical-directive-time-reporting-effective-jan-1-2018
15. https://www.iana.org/time-zones/tz-link
16. https://www.sita.aero/solutions/sita-for-aircraft/operational-intelligence/sita-aircraft-datasuite/
17. https://www.sita.aero/solutions/sita-for-aircraft/operational-intelligence/aircom-serverplatform/

18. https://developer.cirium.com/apis/data/Cirium-flight-status
19. https://developer.flightstats.com/api-docs/scheduledflights/v1/scheduledFlightResponse
20. https://www.nm.eurocontrol.int/HELP/Uploading_schedules.html
21. https://www.iata.org/en/services/codes/
22. https://www.iata.org/contentassets/cfe998bcf9214859afda9c8bf4ff75c3/aidx-xml-imp-guide-v22.1.pdf
23. https://www.transtats.bts.gov/TableInfo.asp?QO_fu146_anzr=b0-gvzr&gnoyr_VQ=FGJ
24. https://www.w3.org/TR/2013/REC-prov-o-20130430/
25. https://developer.flightstats.com/api-docs/flightstatus/v2/flightstatusresponse
26. https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-ge-best-practices.html
27. https://cloud.ibm.com/docs/EventStreams?topic=EventStreams-producing_messages
28. https://developer.flightstats.com/api-docs/flight-status-feed/v2/flight-status-feed-api
29. https://www.flightaware.com/commercial/aeroapi/
30. https://www.eurocontrol.int/publication/eurocontrol-specification-airport-collaborative-decision-making-cdm
31. https://openskynetwork.github.io/opensky-api/index.html
32. https://ganpportal.icao.int/PerformanceManagement/DataAnalytics/AdaptingGanpKpi
33. https://www.bts.gov/topics/airline-time-tables
34. https://ec.europa.eu/eurostat/web/transport/information-data/air-transport
35. https://www.flightaware.com/commercial/data

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.