LANDING.AERO

LANDING.AERO / RESEARCH & PRACTICAL GUIDANCE

Flight Status Data Feeds for OCC: Schedules and OOOI

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 · flight status data feeds · flight status api · oooi · operations control center · flight tracking · flight schedule integration · flightaware · cirium · flightradar24

Original article: https://www.landing.aero/articles/en/flight-status-feeds-occ-tracking-schedules-oooi

In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes
4. Event Provenance and Reconciliation Matrix
5. Implementation Considerations and Process Changes
6. Delivery and Supplier Evaluation
7. Packaged Products and Custom Adapters
8. Data Analysis and Evidence
9. Implications and Future Directions
10. Conclusion

Executive Summary

Flight status data feeds should be evaluated as inputs to an airline's operational picture, with separate treatment of schedules, movement events and aircraft positions. **FlightAware AeroAPI** is query based, while **Firehose** provides a live JavaScript Object Notation (JSON) stream over Transmission Control Protocol (TCP) with Secure Sockets Layer/Transport Layer Security (SSL/TLS). (www.flightaware.com) **Cirium** similarly distinguishes pull-based application programming interfaces (APIs) from its push-based Sky Stream. (www.cirium.com) These delivery choices do not, by themselves, establish which source should prevail when records conflict.

The **out, off, on and in (OOOI)** distinction is fundamental. The US Federal Aviation Administration (FAA) defines these as gate out, wheels off, wheels on and gate in in its Aviation System Performance Metrics (ASPM) documentation. ASPM also estimates missing movement times. An integration therefore needs the event's meaning and provenance, rather than a single undifferentiated departure or arrival timestamp. (www.aspm.faa.gov) (www.aspm.faa.gov) Cirium's published Flight Status Feed cadence of about **six seconds** describes dataset updates on average, not a tested event-to-customer latency guarantee. (developer.cirium.com) Its documented throttle is **3,000 requests per minute per account**. (developer.cirium.com)

Rights can determine suitability before architecture does. **Flightradar24's** API terms prohibit flight-critical or safety-critical decisions and supplementing or backfilling another provider's near-time or real-time flight data. (www.flightradar24.com) (www.flightradar24.com) Its API overview also states a **30-day** storage limit from first receipt. (fr24api.flightradar24.com) These documented conditions require explicit review against the intended workflow; public API access alone does not establish permission for operational reconciliation or long-term audit storage.

The recommended decision process is to define the event contract, confirm permitted use, test representative operator records, and then choose a query API, change feed, streaming connection, existing dispatch product or custom adapter. The report's central asset is an **event provenance and reconciliation matrix**, followed by a hypothetical conflicting-event timeline and transparent Coordinated Universal Time (UTC) lag calculation.

The proposed reconciliation rules are engineering recommendations for operator evaluation, separate from approved operating procedures. No comparative coverage test or operational latency benchmark was performed.

Introduction and Background

An operations control center (OCC) needs a coherent answer to several different questions: what was scheduled, what is now expected, what movement has been observed, and when each assertion reached the system. Buying additional tracking data does not settle those distinctions. OAG treats flight schedules and flight status as separate products, while OpenSky explicitly says its tracking data are not mapped to commercial flight schedules. These are useful boundaries when assessing flight tracking versus flight schedule data. (www.oag.com) (opensky-network.org)

For this report, the **operational picture** means a proposed application view linking the operator's schedule and movement records to external observations, while retaining disagreements and provenance. It is a data-design concept, not a transfer of operational authority to a feed. Schedule exchange has its own established framework: the International Air Transport Association (IATA) describes the Standard Schedules Information Manual (SSIM) as covering airline schedule exchange, airport coordination information and minimum connecting time data. (www.iata.org)

The audience is airline and charter OCC managers, flight department operations teams, and the information technology (IT) staff supporting them. Their decision is usually narrower than choosing a universal flight-tracking vendor. The practical question is whether an existing product already supplies the necessary workflow, whether selected queries can enrich it, or whether continuous event ingestion and reconciliation are justified.

This is a **2026 documentation-based assessment**, with an October 4 publication anchor and source access on October 15 recorded in the accompanying metadata. Undated live documentation is evidence of published behavior at retrieval, not proof of historical contract terms. No vendor accounts, customer payloads or end-to-end feed trials were available for comparative testing. The distinction between documented capability, negotiated entitlement and measured behavior remains explicit throughout.

Key Changes

The changes below describe a proposed integration approach, rather than an announced revision to an aviation standard.

Separate schedules, estimates, observations and positions

A schedule is not merely an early version of an actual movement. OAG's Schedules API describes current and future schedule information, including the structure needed to represent planned services. Cirium's Flight Status documentation separately describes scheduled, estimated and actual departure and arrival times. A canonical record should therefore preserve these categories instead of overwriting a schedule with whichever value arrives next. (knowledge.oag.com) (developer.cirium.com)

A position report belongs in another category. OpenSky's state-vector documentation distinguishes the last position-update timestamp from the time of the last valid transponder message. Consequently, recent contact need not mean a recent position. An OCC display should make that distinction visible rather than calculate freshness from an unrelated message timestamp. (openskynetwork.github.io) (openskynetwork.github.io)

Choose delivery by the change model

A **query API** answers a request for selected records. A **streaming connection** continuously delivers subscribed messages. FlightAware's public comparison provides concrete examples through AeroAPI and Firehose. Neither description supplies a universal ranking: the suitable model depends on the selected fleet, required events, recovery arrangements and contractual access. (www.flightaware.com)

A product called a feed need not use a permanently open push connection. Cirium describes its Flight Status Feed as a stream of changes rather than a lookup mechanism, retrieved through feed resources. That distinction matters when designing cursors and recovery: fetching successive changes differs from repeatedly requesting the latest flight snapshot. (developer.cirium.com)

Table 1 compares documented delivery patterns, together with the next question each raises.

Interface	Documented delivery pattern	Question before adoption
FlightAware AeroAPI	Query-based flight tracking and recent historical data. (www.flightaware.com)	Which records, pages and result sets will the intended polling pattern consume?
FlightAware Firehose	Live JSON stream over TCP with SSL/TLS. (www.flightaware.com)	Which layers, gate events and recovery access are provisioned?
Cirium APIs and Sky Stream	Pull APIs and a push-based stream; Sky Stream requires Advanced Message Queuing Protocol (AMQP) compatibility. (www.cirium.com) (www.cirium.com)	Does the customer need snapshots, successive changes or a subscribed stream?
Flightradar24 API	Endpoint access and rate limits depend on the plan; its terms restrict critical decisions and cross-provider supplementation. (fr24api.flightradar24.com) (www.flightradar24.com) (www.flightradar24.com)	Is the intended noncritical use permitted, including combination and retention?

The table compares interface patterns, not equivalent operational entitlements. A flight status API for airline operations must be assessed against the actual use case; a technically suitable transport does not establish permitted use, complete fields or approved procedures.

Replace arrival-order updates with explicit reconciliation

Receiving a message later does not establish that its contents are newer. Microsoft documents that Azure Event Grid does not guarantee delivery order, while Amazon Simple Queue Service (SQS) documents circumstances in which a message copy can be delivered again. These are infrastructure examples, not claims about any named aviation feed's delivery guarantees. They explain why a custom adapter should make duplicate and ordering assumptions explicit. (learn.microsoft.com) (docs.aws.amazon.com)

Keep FAA definitions within their documented context

FAA ASPM defines OOOI as actual movement times, yet documents estimation when those observations are missing. Its source preference is ASQP OOOI, then OAG OOOI, then Traffic Flow Management System (TFMS) OOOI. The Airline Service Quality Performance (ASQP) source priority describes ASPM construction; it is not a universal hierarchy for an operator's live systems. (www.aspm.faa.gov) (www.aspm.faa.gov) (www.aspm.faa.gov)

Event Provenance and Reconciliation Matrix

The proposed contract should distinguish **scheduled, estimated, predicted and actual** assertions for each movement milestone. FlightAware documents standardized OOOI fields in Firehose, while Cirium documents scheduled, estimated and actual status times. Availability still needs verification against the customer's provisioned messages. (www.flightaware.com) (developer.cirium.com)

Provenance is more specific than a vendor name. FAA ASPM can estimate gate out by subtracting median taxi-out time from wheels off, and gate in by adding median taxi-in time to wheels on. Those are documented ASPM estimation methods, not measured gate observations and not a proposed operational fallback for an OCC. (aspmhelp.faa.gov) (aspmhelp.faa.gov)

Table 2 is a proposed reference asset for evaluating real-time flight status data feeds. Each lag entry is a **question to agree with the provider and operator**, not an asserted service level. “Operator system” means the system designated during discovery; the table does not presume it is authoritative for every event.

Event or record	Source system to inspect	Field and time semantics	Delivery mode to verify	Allowable lag to agree	Proposed fallback	Audit field to retain
Scheduled OOOI	Operator schedule; licensed schedule feed	Separate scheduled out, off, on and in where supplied; preserve schedule revision. (knowledge.oag.com)	Snapshot or schedule change	Schedule publication versus receipt	Last accepted schedule, visibly dated	Schedule version and operating date
Estimated or predicted OOOI	Operator movement system; status feed	Expected milestone, separate from actual; identify which milestone is estimated. (developer.cirium.com)	Query, change feed or stream	Estimate age and receipt delay separately	Retain previous estimate with stale flag	Provider version, issue time and confidence if supplied
Actual out	Operator-approved movement source; entitled gate feed	Gate departure, not takeoff; document gate-return treatment. BTS reporting uses the last gate departure before wheels off in that case. (www.bts.gov)	Movement message or status update	Event-to-receipt and correction lag	Mark unavailable or disputed	Source, actual/derived flag and raw value

Event or record	Source system to inspect	Field and time semantics	Delivery mode to verify	Allowable lag to agree	Proposed fallback	Audit field to retain
Actual off	Operator movement source; external movement feed	Wheels off, distinct from gate out. (www.aspm.faa.gov)	Event or revised snapshot	Observation-to-receipt lag	Preserve competing observations	Event time, receipt time and revision
Actual on	Operator movement source; external movement feed	Wheels on, distinct from gate arrival. (www.aspm.faa.gov)	Event or revised snapshot	Observation-to-receipt lag	Keep landing status separate from gate status	Arrival airport and source event key
Actual in	Operator movement source; entitled gate feed	Gate arrival; confirm the purchased layer includes gate times. (www.flightaware.com)	Gate message or status update	Event-to-receipt lag	Display missing gate event explicitly	Gate, source and correction history
Position	Licensed surveillance interface	Position time differs from latest contact time. (openskynetwork.github.io) (openskynetwork.github.io)	Query or subscribed positions	Position age, transport delay and coverage gaps	Stale-position indication	Observation time, coordinate reference and source
Cancellation or diversion	Operator record; licensed status interface	Preserve cancellation scope; retain intended and actual destination separately. FR24 documents an actual destination for diversions. (fr24api.flightradar24.com)	Status change or revised record	Decision publication versus receipt	Conflict queue for operator resolution	Reason if supplied, destination and superseded state
Alerts and replay	Provider delivery system; adapter journal	Preserve event identity and source; confirm whether replay returns historical changes or current state. (github.com) (www.flightaware.com)	Webhook, feed cursor or reconnect	Recovery delay and missing interval	Approved query or manual fallback	Provider cursor, receipt key and replay flag

The matrix makes missing information visible. It does not turn an inferred event into an observation. For example, Flightradar24 defines first seen as when an aircraft was first detected for a flight leg in UTC. That field should not be silently mapped to gate out or takeoff. UTC normalization also does not change the original event's meaning. (fr24api.flightradar24.com)

Implementation Considerations and Process Changes

Define identity before joining records

A provider flight identifier, flight number, callsign, registration and internal leg key should be treated as separate concepts until samples establish their relationships. FlightAware explicitly recommends treating its flight identifier as opaque. Flightradar24 describes its identifier as assigned to each flight leg and describes its commercial flight number as interpreted from the callsign. Parsing one provider's ID to manufacture another system's identity is therefore an unnecessary assumption. (www.flightaware.com) (fr24api.flightradar24.com) (fr24api.flightradar24.com)

For flight schedule data integration, retain marketing and operating relationships rather than collapsing codeshares into independent aircraft movements. OAG's schedule schema distinguishes marketing-flight references and an operating-flight identifier. Its service suffix is assigned for operational purposes, so omitting that field from matching should be a conscious, tested decision. (knowledge.oag.com) (knowledge.oag.com)

A published IATA dated-operating-flight model defines its flight identifier date using the UTC date of scheduled departure from the station of origin. This older, proposed model snapshot is useful evidence that a flight's identifier date has specific semantics, not proof that every vendor uses the same convention. Document each provider's rule and retain the operator's original operating date. (airtechzone.iata.org)

Specify the adapter's canonical envelope

CloudEvents supplies a common event-description specification. Its source and ID combination identifies distinct events, and its time attribute describes occurrence time with documented provision for a consistently determined substitute when occurrence time is unknown. These concepts can inform an internal envelope without asserting that the selected flight feed implements CloudEvents. (cloudevents.io) (github.com) (github.com)

The following fields are **proposed design requirements**, to validate with actual payloads:

- **Identity:** internal leg key, provider flight ID, operating flight number, callsign, aircraft identity and mapping version.
- **Event meaning:** milestone, scheduled/estimated/predicted/actual category, observation method and missing-value reason.
- **Time:** original timestamp, original offset or epoch, normalized UTC instant and supplied precision.
- **Provenance:** producing system, provider, source event key and any upstream attribution actually delivered.
- **Delivery:** provider publication time if available, adapter receipt time, cursor or sequence and replay indicator.
- **Decision:** selected assertion, alternatives, reconciliation rule version and any operator-entered reason.

For OOOI data integration for airlines, the transformation should be testable without reading a dashboard. A reviewer should be able to trace a displayed gate arrival back to its raw assertion and explain why that value was selected. **LANDING.AERO's** published integration scope describes connecting existing systems and data

sources into a unified operational picture; that is relevant as an integration perspective, not as a source of tracking data. (www.landing.aero)

Establish UTC, precision and update rules

RFC 3339 describes timestamps as unambiguous representations of instants and defines numeric offsets as local time minus UTC. Preserve the original representation alongside the normalized value. Cirium's Flight Tracks documentation, for example, says local time is used by default when its UTC option is false. An adapter should select and verify time behavior explicitly. (www.rfc-editor.org) (www.rfc-editor.org) (developer.cirium.com)

Recommended acceptance cases include:

- **Midnight crossover:** match the intended dated leg without deriving identity from receipt date.
- **Repeated local time:** preserve the offset through a daylight-saving transition.
- **Duplicate delivery:** repeat ingestion without creating another operational movement.
- **Late correction:** retain a corrected observation without accepting every later-arriving stale value.
- **Aircraft reassignment:** update the aircraft relationship without creating a duplicate scheduled leg.
- **Diversion and continuation:** preserve destination history and test whether the provider creates another leg.

These cases are proposed customer-environment tests, not demonstrated results. Flightradar24's sandbox returns predefined static responses matching production schemas. Such a sandbox can support structural checks, but static responses alone cannot establish production coverage, update ordering or observed latency. (fr24api.flightradar24.com)

Delivery and Supplier Evaluation

Evaluate recovery independently from transport

A connection being open is not proof that a consumer has processed the latest available event. FlightAware keepalives expose server system time and a point-in-time replay (pitr) timestamp. Its connection documentation recommends resuming from the pitr timestamp of the last received message. A proposed adapter should persist its recovery state consistently with the records it has durably accepted, then test reconnect behavior. (www.flightaware.com) (www.flightaware.com)

Cirium's Flight Status Feed documents resumption using the last successfully processed payload's unique URL. That establishes a documented mechanism to investigate, not an unlimited contractual replay window. A buyer should ask how long cursors remain usable, whether corrections are included, and what happens when the consumer falls behind the retained interval. (developer.cirium.com)

Internal retention also differs from external entitlement. Apache Kafka documents configurable event retention and ordering within a topic partition. A retained internal log can support reprocessing of lawfully stored messages; it cannot grant rights to retain provider data or establish event-time ordering across sources. (kafka.apache.org) (kafka.apache.org)

Ask procurement questions that expose the data contract

The following questions form a proposed evaluation checklist. Answers should be recorded against the applicable contract, documentation version and representative sample:

- **Access scope:** which aircraft, regions, event types and source layers are included?
- **Gate completeness:** are out and in supplied for the actual fleet, or only off and on?
- **Identifiers:** how do IDs behave across rescheduling, cancellations, diversions and continuation legs?
- **Update ordering:** is there a sequence, revision or cursor, and what is its scope?
- **Corrections:** how are withdrawn estimates, corrected actuals and null values represented?
- **Replay:** what recovery window, retrieval method and outage behavior are contractually available?
- **Retention:** may raw payloads, normalized events, derivatives and backups be stored for the required periods?
- **Combination rights:** may this feed be reconciled with operator records and other licensed providers?
- **Distribution:** may staff, contractors, customer applications and downstream systems receive the data?
- **Restrictions:** how are blocked aircraft and authorized operator access handled?
- **Failure behavior:** what signals throttling, authentication failure, subscription expiry and missing coverage?
- **Support evidence:** what field dictionary, change notice and sample-validation process is supplied?

Some of these questions have consequential public answers. Flightradar24's API terms prohibit flight-critical or safety-critical decisions and supplementing or backfilling other near-time or real-time flight data. Those conditions must be reviewed before proposing it as an input to a live combined operational picture. The report does not infer permission for that workflow. (www.flightradar24.com) (www.flightradar24.com)

OpenSky similarly says commercial entities and continued operational users must obtain a license. Availability through a public endpoint is therefore distinct from authorization for sustained business use. FlightAware's Firehose documentation also identifies repurposing and redistribution scope as a pricing consideration. (opensky-network.org) (www.flightaware.com)

Specify blocked-aircraft and failure behavior

The FAA's **Limiting Aircraft Data Displayed (LADD)** program distinguishes filtering options. Under FAA Source filtering, FAA flight data are unavailable to external vendors; subscriber-level filtering allows selective monitoring arrangements with an internet vendor. These are US FAA distribution arrangements, not a global promise of private-fleet visibility through any chosen API. (www.faa.gov) (www.faa.gov)

A buyer should request an authorized demonstration for the relevant restricted aircraft and applicable data sources. A public display, an anonymized observation and an operator-authorized identified feed can represent different entitlements.

The fallback path also needs deliberate configuration. Microsoft documents that Event Grid dead-lettering is not enabled by default. This is an infrastructure example: if an adapter uses that service, failed deliveries require an explicit retention and recovery design rather than an assumption that rejected events remain available indefinitely. (learn.microsoft.com)

Packaged Products and Custom Adapters

The selection should start with the operational workflow already in place. An existing dispatch, flight-following or operations-management product is a category to evaluate when the requirement includes staffed workflows, acknowledgments and established operational processes. A data feed addresses a narrower input problem. Custom development becomes a candidate when the operator already has the required operational systems but cannot reliably connect their records.

A proposed decision sequence is:

- **Keep the existing product** when its licensed data, event definitions and workflow satisfy the requirement.
- **Add selected queries** when staff need bounded enrichment and the provider permits the intended access pattern.
- **Adopt a change feed** when intermediate changes and cursor-based recovery matter.
- **Adopt a continuous stream** when the subscribed event volume and workflow justify continuous consumption.
- **Build an adapter** when identifier mapping, provenance, conflict handling or integration is the unmet requirement.
- **Reassess the product category** when the missing capability is dispatch or flight-following service rather than data ingestion.

FlightAware's documented query and streaming options, and Cirium's pull and push distinction, support this delivery taxonomy. They do not determine which option fits a specific operation without samples, volumes and contract review. (www.flightaware.com) (www.cirium.com)

A custom adapter should have a bounded responsibility: ingest permitted data, normalize without losing meaning, preserve provenance, apply agreed reconciliation rules and expose unresolved disagreements. It should not manufacture missing actuals or hide uncertainty to make a dashboard appear complete. The design remains separate from any operator-approved procedure.

LANDING.AERO describes its work as tailored flight operations software, including integration, workflow automation and dashboards. Within this decision, that represents the custom-build category; it is not a competing flight-tracking feed or a dispatch service. No off-the-shelf platform, vendor partnership or tested integration is implied by that description. (www.landing.aero)

The practical deliverable before implementation is a customer-reviewed mapping specification: actual field names, permitted uses, sample records, missing-value behavior, recovery assumptions and reconciliation outcomes. That specification makes an adapter proposal reviewable and gives an existing product a fair basis for comparison. The choice should follow the verified gap, rather than the availability of a preferred technology.

Data Analysis and Evidence

Interpret documented quantities correctly

The public documentation provides useful constraints, but not comparable performance scores. Cirium describes Flight Status Feed dataset updates about once every **six seconds on average**, with a throttle of **3,000 requests per minute per user account**. The former is a dataset cadence; the latter is an interface limit. Neither is a tested maximum message lag for a customer's flights. (developer.cirium.com) (developer.cirium.com)

AeroAPI's product page describes a billing result set of **15 records**. Capacity and cost modelling should therefore distinguish Hypertext Transfer Protocol (HTTP) requests, returned records, pagination and billable sets, using the selected endpoint and contract. This report does not calculate a price from the result-set definition. (www.flightaware.com)

Retention is another independent axis: Flightradar24's API overview states that data should not be stored beyond **30 days from first receipt**. A requirement for a longer audit history must be resolved against applicable rights before selecting that interface. Historical data being queryable would not itself establish unrestricted storage permission. (fr24api.flightradar24.com)

Collection frequency and source precision also differ. InfluxData's flight-monitoring tutorial fetches data every **five minutes**; that is its example script's polling interval, not FlightAware delivery latency. A US Bureau of Transportation Statistics (BTS) reporting directive specifies **whole-minute** reporting, illustrating why a source's timestamp precision must be understood before comparing small timing differences. (www.influxdata.com) (www.bts.gov)

Conflicting-event timeline (Hypothetical Example)

All timestamps and events in Table 3 are **invented**, on the illustrative UTC date **2026-10-04**. The anonymous leg was scheduled from origin A to destination B and later diverted to alternate C. No row represents an actual flight, provider measurement or operator-approved procedure.

Table 3 shows proposed reconciliation outcomes. Times use UTC with a Z suffix; the example applies RFC 3339 timestamp principles. (www.rfc-editor.org)

Assertion received	Event or issue time, UTC	Adapter receipt, UTC	Proposed reconciliation outcome
Scheduled out from operator schedule	13:00:00Z	12:00:00Z	Keep scheduled out in its own field.
Actual out from designated movement source	13:06:00Z	13:06:08Z	Select as actual out; retain schedule.
Older estimated out from external snapshot	Issued 13:04:00Z; estimate 13:05:00Z	13:07:00Z	Retain as historical estimate; do not replace actual out.
Actual off from licensed movement feed	13:18:00Z	13:18:42Z	Record wheels off independently of gate out.

Assertion received	Event or issue time, UTC	Adapter receipt, UTC	Proposed reconciliation outcome
Repeat of the same actual-off event	13:18:00Z	13:19:10Z	Preserve receipt audit; avoid a duplicate movement.
Diversion status naming alternate C	Issued 14:10:00Z	14:10:05Z	Retain intended B and revised destination C.
Actual on at alternate C	14:41:20Z	14:41:32Z	Record landing at C; leave actual in unavailable.

The later receipt of the older estimate does not make it a better actual-out observation. Likewise, landing does not supply a missing gate-in event. These are proposed rules for this hypothetical contract, requiring agreement on source authority and correction behavior.

For the invented wheels-off message, **message lag = receipt time minus event time = 13:18:42Z minus 13:18:00Z = 42 seconds**. At an illustrative display time of **13:20:00Z**, the event's age is **120 seconds**. The repeated delivery must not reset the original event time or turn that age into a freshness claim. These quantities are arithmetic from the invented inputs, not benchmarks.

A proposed measurement plan would retain:

- **Event lag:** receipt minus occurrence, where occurrence is actually supplied.
- **Publication lag:** receipt minus provider publication, when documented.
- **Position age:** display time minus position observation time.
- **Processing delay:** display-ready time minus adapter receipt.
- **Recovery gap:** the interval missing before replay and reconciliation finish.

CloudEvents permits a substitute timestamp when occurrence time cannot be determined. Such a substitute must be identified before calculating event lag. Likewise, OpenSky's last-contact timestamp can advance without a new position; using it would measure a different kind of freshness. (github.com) (openskynetwork.github.io)

Implications and Future Directions

A useful airline operations control center flight-tracking evaluation should produce evidence about the operator's records, rather than a vendor league table. The next step is a permitted sample environment containing the actual fleet and workflows. Any conclusion about coverage, blocked-aircraft visibility or message lag should remain unmeasured until that exercise occurs.

Published developer examples offer a useful methodological contrast. Esri's flight knowledge-graph demonstration explicitly models connectivity rather than actual aircraft routes. InfluxData's tutorial uses periodic collection for a monitoring example. Both are concrete integration references, but neither establishes an operational-control workflow or a comparative service benchmark. (community.esri.com) (www.influxdata.com)

For future integrations, a common event envelope could make adding another permitted source easier to reason about. CloudEvents describes event data in a common way, but standardizing the wrapper does not standardize aviation semantics. An “arrival” still needs a milestone, location, source and revision policy. Interoperability work should therefore cover both schema and meaning. (cloudevents.io)

Recommended outputs from an evaluation are:

- **A source register** linking each displayed field to its origin and permitted use.
- **A reconciliation register** recording precedence, correction handling and unresolved cases.
- **A rights register** covering combination, storage, derivatives, backups and distribution.
- **A replay demonstration** showing recovery from an interruption without losing accepted history.
- **A measurement report** separating observation age, delivery lag, processing delay and missingness.

These are proposed evaluation artifacts, not claims that a provider supplies them. Internal replay remains dependent on retention configuration: Kafka, for example, explicitly makes retention configurable. An architecture diagram should therefore state which history is retained, by whom, for what purpose and under which rights. (kafka.apache.org)

The same discipline applies to automation. A generated summary or recommendation should be traceable to the underlying assertions and visibly preserve uncertainty. The initial engineering objective should be explainable reconciliation; additional automation should be evaluated only after identity, timing and recovery behavior are demonstrated in the customer environment. Operator-approved procedures remain a separate source of requirements.

Conclusion

Flight status data feeds are most useful when their role is precisely defined. The decision should begin with the existing schedule and movement records, the missing information, the intended workflow and the rights to combine and retain it. Delivery technology follows those requirements.

The proposed provenance matrix gives operations and IT teams a shared review tool. It separates scheduled and estimated times from actual out, off, on and in; distinguishes positions from movement events; and requires audit information for corrections, diversions, stale messages and replay. Its central recommendation is to preserve competing assertions before deciding which value a screen should show.

The hypothetical timeline demonstrates why message arrival order cannot stand in for event meaning. A later-received estimate need not replace an actual movement, a duplicate should not create another movement, and landing at a diversion airport does not establish gate arrival. Transparent UTC calculations make the difference between delivery lag and event age reviewable.

A query API can fit bounded enrichment. A change feed or stream can fit continuous ingestion when access, recovery and operational requirements justify it. An existing operations product deserves consideration when the missing requirement is a workflow or service. A custom adapter fits a verified integration gap. In each case, the defensible choice is the one whose data contract, permitted use and customer-environment behavior can be demonstrated.

Source links

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

1. https://www.flightaware.com/commercial/data/
2. https://www.cirium.com/data/aviation-api/
3. https://www.aspm.faa.gov/aspmhelp/index/OOOI_Data.html
4. https://developer.cirium.com/apis/cirium-sky-api/flight-status-feed
5. https://www.flightradar24.com/terms-of-service
6. https://fr24api.flightradar24.com/docs/endpoints/overview
7. https://www.oag.com/flight-info-api
8. https://opensky-network.org/about/faq
9. https://www.iata.org/en/publications/manuals/standard-schedules-information/
10. https://knowledge.oag.com/docs/schedules-api-response-layout-descriptions
11. https://developer.cirium.com/apis/flightstats-apis/flight-status
12. https://openskynetwork.github.io/opensky-api/rest.html
13. https://fr24api.flightradar24.com/docs/getting-started
14. https://learn.microsoft.com/en-us/azure/event-grid/delivery-and-retry
15. https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/standard-queues-at-least-once-delivery.html
16. https://www.flightaware.com/commercial/firehose/faq.rvt
17. https://aspmhelp.faa.gov/index/ASPM__Estimation_Techniques.html
18. https://www.bts.gov/explore-topics-and-geography/modes/aviation/number-39-technical-directive-reporting-time
19. https://www.flightaware.com/commercial/firehose/documentation/summary
20. https://fr24api.flightradar24.com/docs/endpoints/flight-summary
21. https://github.com/cloudevents/spec/blob/v1.0.2/cloudevents/spec.md
22. https://www.flightaware.com/commercial/firehose/documentation/connection
23. https://airtechzone.iata.org/aidm_model/21.3/EARoot/EA5/EA3/EA4/EA2/EA11061.htm
24. https://cloudevents.io/
25. https://landing.aero/articles/en/oooi-event-integration-acars-flight-times
26. https://www.landing.aero/
27. https://www.rfc-editor.org/rfc/rfc3339.html
28. https://developer.cirium.com/apis/cirium-sky-api/flight-track
29. https://www.flightaware.com/commercial/firehose/documentation/commands
30. https://kafka.apache.org/41/getting-started/introduction/
31. https://www.faa.gov/pilots/ladd
32. https://www.landing.aero/about/
33. https://www.flightaware.com/commercial/aeroapi/

34. https://www.influxdata.com/blog/monitoring-flights-influxdb/

35. https://community.esri.com/en/discussion/1625837/create-and-maintain-a-knowledge-graph-with-one-etl-tool

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.