A Multi-Detector Decision-Support Architecture for Air Traffic Control

Filipe Brito Ferreira | Independent researcher | https://www.fbritoferreira.com

August 2026 · v0.5

Version 0.5 · August 2026

Section 6 is a *diagnostic shape check* against a corpus of nine publicly-documented incidents. Sections 6.7 to 6.9 report the false-positive analysis, the negative-control corpus and the monolithic-baseline comparison, including a false-positive table that was withdrawn and re-measured after its harness was found to be testing a reimplementation rather than the system.

Reproducibility

The live deployment is at https://atc.fbritoferreira.com, and the nine historical-incident reconstructions load from the scenario picker in the application header. A reviewer opening that deployment may find live mode unavailable rather than merely stale. Checked on 2026-09-01 against the endpoint the client actually calls, for Atlanta and for Kennedy, the proxy answered HTTP 403 with a body naming both upstreams it had tried: `adsb.lol` refused with 429, which it does to Cloudflare's shared egress ranges, and `airplanes.live` refused with 403. Neither feed is open to this deployment at present. The refusal follows the caller rather than the call: the same adsb.lol endpoint, asked for the same thirty-mile radius around Atlanta from a residential connection on 2026-09-02, answered 200 with 86 aircraft. A feed that is down and a feed that declines one range of addresses look identical from inside the proxy, and only the second is happening. The failure is surfaced rather than hidden in this checkout, since the proxy passes a non-ok status through and names the sources that refused, the client raises on it instead of rendering an empty sky, and a panel explains what is unavailable. That panel is not on the deployed build: it was added on 2026-08-29 to the branch carrying these corrections, which is unmerged, and the three asset bundles the site serves do not contain it. What the older build shows when both feeds refuse has not been established, so this paragraph describes the repository and not the thing a reader clicks, which is the same gap the deployment section reports for the paper itself; the scenario reconstructions are built into the bundle and unaffected. It is stated here because an empty live view invites the reading that the detectors found nothing, which is a different claim from the feed being closed. `scripts/check-live-feed.sh` runs that probe against both fields, so the state is measured on the day a reader asks rather than taken from this sentence, which will age. One consequence is worth stating for anyone who opens the deployment while reading this. Publishing was blocked from 2026-08-28 to 2026-09-01 by an account billing failure rather than by anything in the code. It has since cleared, and a deploy landed on 2026-09-01, so the site no longer serves the build from the start of that block; it serves what has been merged, which trails this revision because the branch carrying these corrections is not merged yet. A reviewer visiting it today therefore gets an application predating some of the ingest corrections described above, reading magnetic heading where a true track is meant and placing an undistanced record at the field. The corrections are in the repository and in the measurements reported here, and they are not in the thing a reader can click. The size of that gap is measured rather than inferred: `scripts/check-deployed.sh` downloads the PDFs the site serves and compares each against what this checkout builds. The served copy of this document and the count it is measured against both move, the first when a deploy lands and the second with every revision, so the script reports the pair rather than this paragraph fixing either. The other published document shows no gap. It exists because three claims about the deployed site were made from commit dates inside two days and all three were wrong, two of them on their way into a paper. Those scenarios are deterministic and touch no network, so they reproduce

regardless of whether the public ADS-B feeds are serving. The source is licensed Apache 2.0 and public at `github.com/fbritoferreira/atc-decision-support`.

Abstract

This thesis introduces a multi-detector decision-support architecture for air traffic control (ATC) and exercises it against publicly-documented aviation incidents. The architecture is, in architectural terms, a constrained blackboard system: a thin orchestrator coordinates a population of independent rule specialists, each owning a single doctrinal concern (wake-turbulence spacing, runway conflict, runway identity, fuel reserve, crosswind, weather, runway surface, gate conflict, cascading delay, proximity, emergency squawk), and projects state forward in time to surface predictive alerts. Output is structured and carries explicit reasoning and recommended actions. The implementation is a proof of concept, licensed Apache 2.0 and intended for public release, fed by live Automatic Dependent Surveillance-Broadcast (ADS-B) data and live METAR weather reports for seventeen airports, ten of them in the United States. The work reported here is a *diagnostic shape check*: nine publicly-documented incidents are encoded as deterministic scenarios and the detector output is compared against the principal causes identified by the responsible investigation bureaux. In each of the eight detectable cases, among them the 2025 Potomac River mid-air collision (NTSB DCA25MA108), the 1990 Avianca 052 fuel-exhaustion accident (NTSB AAR-91/04), and the 1977 Tenerife runway collision (CIAIAC), the alert category the system produces corresponds to the official principal cause. The ninth, American 11 on 11 September 2001, is included as a documented blind spot: its hijack signatures are invisible to this architecture by construction, and a test enforces that silence. Sections 6.7 to 6.9 report the measured false-positive rate on nominal traffic, the negative-control corpus and the monolithic-baseline comparison; what those measurements do not amount to is a validation in the strict sense, and Section 6.1 says why. The false-positive analysis produced the thesis's sharpest empirical result: after three rounds of doctrine correction, the residual wake violations mark the boundary of what surveillance data can adjudicate, because accepted visual separation voids the radar minima (FAA JO 7110.65 §7-2-1), the acceptance is a voice-channel event no feed carries, and every violating pair across the two completed wake windows formed under the visual conditions where that regime applies. Instrument-condition sampling since has produced five pairs and two violations, the single observation that explanation does not reach, and Section 6.7.1 reports it rather than leaving a reader to find the exception. The contributions at this stage are: (i) the articulation of a Specialist Detector pattern situated within the blackboard tradition (Erman et al., 1980) and informed by automation-trust literature (Bainbridge, 1983; Parasuraman and Riley, 1997); (ii) an open implementation against real-world data sources; (iii) a reproducible historical-incident corpus suitable as a regression fixture for future ATC decision-support work, replayable in BlueSky; and (iv) the measured visual-separation boundary above, with the scripts to re-measure it on a fresh window, the windows behind the published figure not being distributed.

---

1. Introduction

1.1 Motivation

Aviation operates under decision-support systems whose maturity is uneven across the operational stack. The Traffic Collision Avoidance System (TCAS) is mandated in commercial transport aircraft and produces deterministic resolution advisories; ground-side, the picture is more fragmented. Controllers rely on a mix of integrated radar displays, separate flight-strip software, paper backup, voice coordination with adjacent sectors, and human pattern recognition. Recent high-profile events (the 29 January 2025 mid-air collision near Reagan Washington National Airport, the May 2025 LaGuardia aborted takeoff, the April 2026 JFK airborne near-miss) share a common shape: the operator received the inputs that, in retrospect, predicted the unsafe condition, but the system did not surface the prediction as an actionable alert before the event.

This thesis proposes that this gap can be closed, in part, by an architectural pattern that treats decision support as an orchestration problem rather than a single-model problem.

1.2 Problem statement

The decision-support functions a controller depends on are doctrinally distinct: wake-turbulence spacing, runway occupancy, fuel-reserve compliance, crosswind limits, gate availability, sequencing throughput, mid-air proximity. Each is governed by an explicit rule set published by a national or international authority (FAA, ICAO, AAIB, CIAIAC). A single algorithm cannot represent all of them well; a stack of monolithic algorithms produces unmanageable cross-cutting interactions.

The research question is: can these doctrines be implemented as independent, composable detectors under a common orchestration shape, such that the system's output is explainable, predictively useful, and verifiable against the historical record?

1.3 Approach

The approach is design-and-implement. A pattern is proposed (Section 3); operational practices are specified (Section 4); a complete implementation is built (Section 5); historical incidents are encoded as test scenarios and the detector output is compared against the published investigation findings (Section 6).

1.4 Contributions

1. **The Specialist Detector pattern.** A formulation of decision support as a population of independent rule specialists coordinated by a thin orchestrator over a shared blackboard, with explicit contracts for severity, reasoning, and recommended action. The pattern sits in the blackboard tradition of Erman et al. (1980) with stricter purity constraints justified by aviation regulatory requirements. The contribution is the articulation and the open implementation, not novelty of the underlying control structure.
2. **An open implementation against real-world data sources.** A web application fed by live ADS-B and METAR data, deployed to seventeen airports, ten of them in the United States, supporting both live operation and frozen historical reconstructions.
3. **A reproducible historical-incident corpus.** Nine publicly-documented incidents encoded as deterministic test scenarios, with mappings to the principal causes identified by the responsible investigation bureaux, plus eleven negative controls, three of them encoding specific NASA ASRS reports. Every scenario exports to BlueSky. The corpus is offered as a regression fixture, not a benchmark; the difference is discussed in Section 6.1.
4. **The measured visual-separation boundary.** A false-positive analysis across repeated live windows at five US airports, reported round by round in Sections 6.7.1 to 6.7.4, whose end state is a doctrinal boundary rather than a score: below the radar minima in visual conditions, surveillance data alone cannot distinguish a violation from legally delegated visual separation (Section 6.7). The scripts that produce the result ship with the implementation; the measurement windows do not, so a reader re-runs the analysis on a window they sample rather than replaying these. Section 6.10 states what that costs.

1.5 Outline

Section 2 reviews related work in ATC decision support and multi-detector architectures. Section 3 specifies the Specialist Detector pattern. Section 4 describes the operational properties the implementation must satisfy. Section 5 describes the implementation. Section 6 presents the diagnostic-shape check against historical incidents and reports the false-positive analysis, negative-control corpus and baseline comparison that earlier revisions listed as open. Section 7 discusses generalisability and limitations. Section 8 sets out a pathway to US adoption. Section 9 outlines future work. Section 10 concludes.

Where this could matter, counted rather than asserted. The FAA operates surface surveillance at 44 US fields: ASDE-X at 35 of them, and its ADS-B successor ASSC at nine more, which the agency names as Anchorage, Cincinnati, Cleveland, Kansas City, New Orleans, Pittsburgh, Portland, San Francisco and Andrews. The United States has 527 airport traffic control towers. So 483 towered fields have no dedicated surface surveillance, and their controllers work from visual observation and procedure.

That is not an argument that this prototype belongs at those fields. The architecture is alerting logic rather than surveillance: it cannot see what a surface radar sees, and none of it has been validated operationally anywhere. What the count establishes is narrower. The input it consumes, cooperative ADS-B, is present at fields where dedicated surface equipment is not, and the distance between the equipped 44 and the remaining 483 is the reason deterministic logic over already-available data is worth investigating.

2. Background and Related Work

2.1 Situation awareness and decision support

Endsley's three-level model of situation awareness (Endsley, 1995) remains the operative framework for human-in-the-loop systems in aviation. The model distinguishes perception (level 1), comprehension (level 2), and projection (level 3). The architecture proposed here corresponds principally to level 2 (the system interprets the operational picture and emits structured alerts) and level 3 (the orchestrator projects state forward in time and surfaces predictive alerts).

The decision-support systems literature (Power, 2002) predates aviation-specific automation by several decades and emphasises transparency, audit trails, and operator override. These properties are reified in this thesis as per-alert *reason* and *suggested action* metadata, persistent severity tiers, and the absence of any automated control action: the system advises, the operator decides.

2.2 Traffic Collision Avoidance System

TCAS II (RTCA DO-185B) is the canonical onboard collision-avoidance system. TCAS classifies conflicts using a time-to-closest-point-of-approach test (the *tau* threshold, typically 15 to 35 seconds depending on sensitivity level) combined with altitude-band-dependent distance floors (DMOD horizontal and ZTHR vertical). The threshold pair is not a single (0.5 NM, 200 ft) value; the actual thresholds vary by altitude per the DO-185B sensitivity-level tables. The ground-side proximity-conflict detector in Section 5.4 began as a distance-only test and no longer is: its *critical* tier computes time to closest point of approach from the pair's relative position and velocity and requires the horizontal miss distance at that point to fall inside DMOD, which is the same family of test TCAS runs. Its *warning* tier remains a distance box. Two differences from airborne TCAS remain, an instantaneous vertical test in place of a rate-based one and single thresholds in place of DO-185B's altitude-banded sensitivity levels. Section 5.4 gives both with the reason, and this paragraph used to give them again in the same words. An earlier revision of this section said the detector did not implement *tau* at all; that stopped being true when the convergence test was added, and the sentence outlived the change. The detector does not issue control instructions; it surfaces a projected conflict to the operator.

Closely spaced parallel approaches deserve separate mention, because they are the geometry this prototype handles worst and the field has purpose-built systems for them. The ILS Precision Runway Monitor supports simultaneous independent approaches down to a 3,400 ft lateral floor, and NASA's Airborne Information for Lateral Spacing concept, evaluated in simulation and flight test, targets runways as close as 2,500 ft using DGPS-grade path keeping with a flight-deck alerting algorithm rather than a distance box. Both exist because a proximity test that looks only at instantaneous separation cannot distinguish two aircraft correctly flying adjacent localisers from two aircraft in conflict. NASA ASRS CALLBACK carries controller and pilot reports of TCAS advisories on blunder-free parallel approaches for the same reason.

This prototype's proximity warning envelope is 2 NM. PRM operations run at 0.56 NM and AILS targets 0.41 NM, so the envelope is wider than the systems whose thresholds the detector's own reasoning string cites, and every legal simultaneous parallel approach falls inside it. Section 6.8 encodes that case as a negative control and reports what the detector does with it.

2.3 Deployed ground-based alerting

The FAA's NextGen and Eurocontrol's SESAR programmes drive incremental automation of US and European airspace. Both target enhanced surveillance accuracy, automated sequencing, and improved trajectory

prediction. Beneath the programme level, a specific population of ground-based alerting systems is already in service, and this work must be positioned against it rather than beside it.

In US terminal automation, STARS carries Conflict Alert and MSAW, and since 2011 the Automated Terminal Proximity Alert (ATPA): a monitor of in-trail separation on the final approach course that writes the live separation into the trailing aircraft's datablock and draws a cone sized to the pair's required separation, wake tables included, with a warning stage 45 seconds and an alert stage 24 seconds before predicted loss. ATPA is the deployed analogue of this work's wake-spacing detector, and no novelty is claimed here for distance-based in-trail alerting on final; the differences are that ATPA runs inside FAA automation with facility-authorized minima and suppression under visual separation, while the detector here runs from public ADS-B alone, with the visibility limits Section 6.7 measures. On the surface, ASDE-X and ASSC safety logic project the paths of arrivals and surface traffic at 44 equipped airports, and Approach Runway Verification (operational in roughly 85 towers since about 2024) raises wrong-surface alerts of the class the runway-identity detector implements. In Europe, EUROCONTROL's ground-based safety nets, STCA, MSAW, APW and APM, operate as four separate single-doctrine rule engines with defined prioritisation between them: operational practice already runs a specialist-detector population without naming it, each member shipped and adapted as its own system. The contribution claimed here is accordingly narrow: one thin orchestrator over the population, alert lifecycle semantics (demotion, subsumption, persistence) defined once and tested, and a historical-incident corpus that regression-tests the whole.

Two deployment facts locate the niche such a system could occupy. The FAA's Surface Awareness Initiative ships ADS-B-only traffic displays to towers lacking surface surveillance, roughly 490 towered airports have none, and SAI deliberately ships without alerting logic. And GAO reporting records Terminal Sequencing and Spacing, the projection-heavy terminal scheduling tool, as deferred without an implementation timeline, while the reactive short-horizon alerting tool, ATPA, deployed nationwide. An ADS-B-only detector population with short-horizon projection points at that gap: the surveillance-poor tail of the system, alerting rather than scheduling.

2.4 Blackboard and multi-detector architectures

The pattern presented here is, in architectural terms, a constrained blackboard system. The blackboard model, formalised by Erman, Hayes-Roth, Lesser, and Reddy in the Hearsay-II speech-understanding system (Erman et al., 1980), organises problem-solving around a shared data structure (the blackboard) that independent knowledge sources read and write under the coordination of a control component. The orchestrator in this work plays the role of the control component; each detector plays the role of a knowledge source operating over a shared `SimState`. The principal difference is that this work imposes strict purity on the knowledge sources: detectors read the blackboard but do not write to it, and all output is routed through the orchestrator. This restriction sacrifices some of the expressive power of the original blackboard model in exchange for determinism and auditability, properties required by aviation regulation.

Multi-agent systems (Wooldridge, 2009; Stone and Veloso, 2000) typically emphasise autonomous reasoners that coordinate via explicit communication protocols. ATC has been a testbed for that tradition since its beginning: Cammarata, McArthur and Steeb (1983) used distributed conflict resolution among cooperating planners as the canonical distributed-AI problem; the OASIS arrival manager trialled with Australian ATC assigned one BDI agent per aircraft (Ljungberg and Lucas, 1992); NASA's CTAS decomposed arrival decision support into cooperating tools over a shared trajectory core (Erzberger et al., 1990s); and negotiation-based conflict resolution continued through Wollkind et al. (2004) and AGENTFLY (Šišlák et al., 2007). The consistent axis in that literature is agency per aircraft or per airspace element. The decomposition here runs along a different axis, one specialist per piece of doctrine, and drops autonomy: each specialist is a deterministic function over the operational state, and the orchestrator owns all routing and persistence. The result is closer to a directed graph of pure functions over a shared blackboard than to a society of agents. The nearest recent neighbour is the rules-based tactical agent of Kent et al. (2026), which embeds an expert-informed deconfliction-strategy library in a UK digital twin; it plans and resolves where this work detects and alerts.

2.5 Conflict detection and trajectory prediction

The conflict-detection literature is foundational here. Kuchar and Yang (2000) review fifty-plus conflict-detection-and-resolution methods spanning state-based, intent-based, and probabilistic approaches; the detector population in Section 5 is broadly state-based with deterministic thresholds and corresponds to the simplest cell of their taxonomy. Modern trajectory-prediction work using ADS-B (Sun et al., 2020; Pang et al., 2021) demonstrates that data-driven projection can reduce the linear-extrapolation error this work currently exhibits; this is noted as a limitation in Section 7.4 and a future-work item in Section 8. Two facts bound how much that limitation costs at this work's horizons. Deployed practice converged on roughly the same window: EUROCONTROL's STCA guidance limits look-ahead to about two minutes because trajectory-prediction uncertainty grows beyond it, which independently corroborates the one-to-three-minute horizon chosen here. And in the adjacent vehicle-trajectory literature, constant-velocity baselines remain competitive with learned models at short horizons; no published ATC-specific comparison at exactly one to three minutes was found, so the claim is stated as corroboration rather than proof.

2.6 Automation and operator trust

Automation in safety-critical systems carries well-documented failure modes. Bainbridge's *Ironies of Automation* (Bainbridge, 1983) observes that the more reliable an automated system becomes, the less the operator practises the manual skills required to handle the residual failures. Parasuraman and Riley (1997) formalise the misuse / disuse / abuse triad of operator response to automation. Cummings (2017) and the broader human-factors literature catalogue *alert fatigue* in operational environments where the alert rate exceeds the operator's response budget.

The ATC-specific evidence is more precise than the general phrase, and it cuts in two directions. Deployed alerting runs high nuisance fractions: a study of operational FAA facilities classified 62% of en-route Conflict Alerts and 91% of en-route MSAW alerts as unnecessary, with 44% and 61% in terminal facilities (Friedman-Berg, Allendoerfer and Pai, 2008). Yet a field study of 497 operational en-route conflict alerts found *no* measurable cry-wolf effect at a 47% false-alert rate: controllers had usually anticipated the conflict before the alert fired, so unnecessary alerts were redundant rather than disbelieved (Wickens et al., 2009). The cost model this work adopts is therefore attention and workload rather than lost trust: every unnecessary critical claims a scan and a judgment from an operator with other uses for both. EUROCONTROL's STCA guidance material formalises the vocabulary used throughout Section 6: a *false* alert does not correspond to a real situation (split tracks, reflections), while a *nuisance* alert is correctly generated by the rule set but operationally inappropriate (EUROCONTROL, 2017). These failure modes are taken seriously in this work: the severity tiering (Section 4.4), the explicit reasoning attached to every alert (Section 4.2), and the absence of any automated control action (see Section 10) are all responses to this literature.

2.7 Why determinism

The reasons for the design choices in Section 3 are doctrinal. Aviation rules are deterministic; explainability requires that the same inputs produce the same outputs; audit trails require reproducibility; operator trust requires that the system's recommendations be inspectable. A non-deterministic reasoner (for example, a large language model) would violate these properties unless heavily constrained. The proposed pattern occupies the constrained end of the design space.

2.8 Aviation incident investigation

The diagnostic-shape check relies on the published reports of national investigation bureaux: the US National Transportation Safety Board (NTSB), the Italian Agenzia Nazionale per la Sicurezza del Volo (ANSV), and the Spanish Subsecretaría de Aviación Civil / Comisión de Investigación de Accidentes e Incidentes de Aviación Civil (CIAIAC). These reports identify principal and contributing causes following a standardised methodology and serve as ground truth against which detector output is compared.

---

3. The Specialist Detector Pattern

3.1 Definition

The pattern organises a decision-support system into three concentric layers (Figure 1):

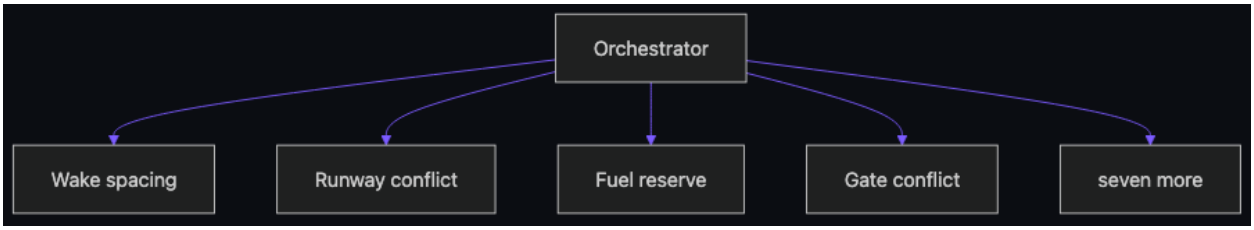

Figure 1: The Specialist Detector pattern. The orchestrator owns ingestion, projection, composition and emission. Each of the eleven detectors owns exactly one doctrinal concern and knows nothing of the others.

Figure 2 sets out the order in which those run. The orchestrator owns: state ingestion (ADS-B, METAR, route lookups), state projection (forward look-ahead), invocation of detectors over current and projected states, alert composition, de-duplication, and emission to the operator surface. Detectors own: a single doctrinal concern, the rule that operationalises it, a contract with the orchestrator, and the production of structured alerts with severity, reason, and recommended action.

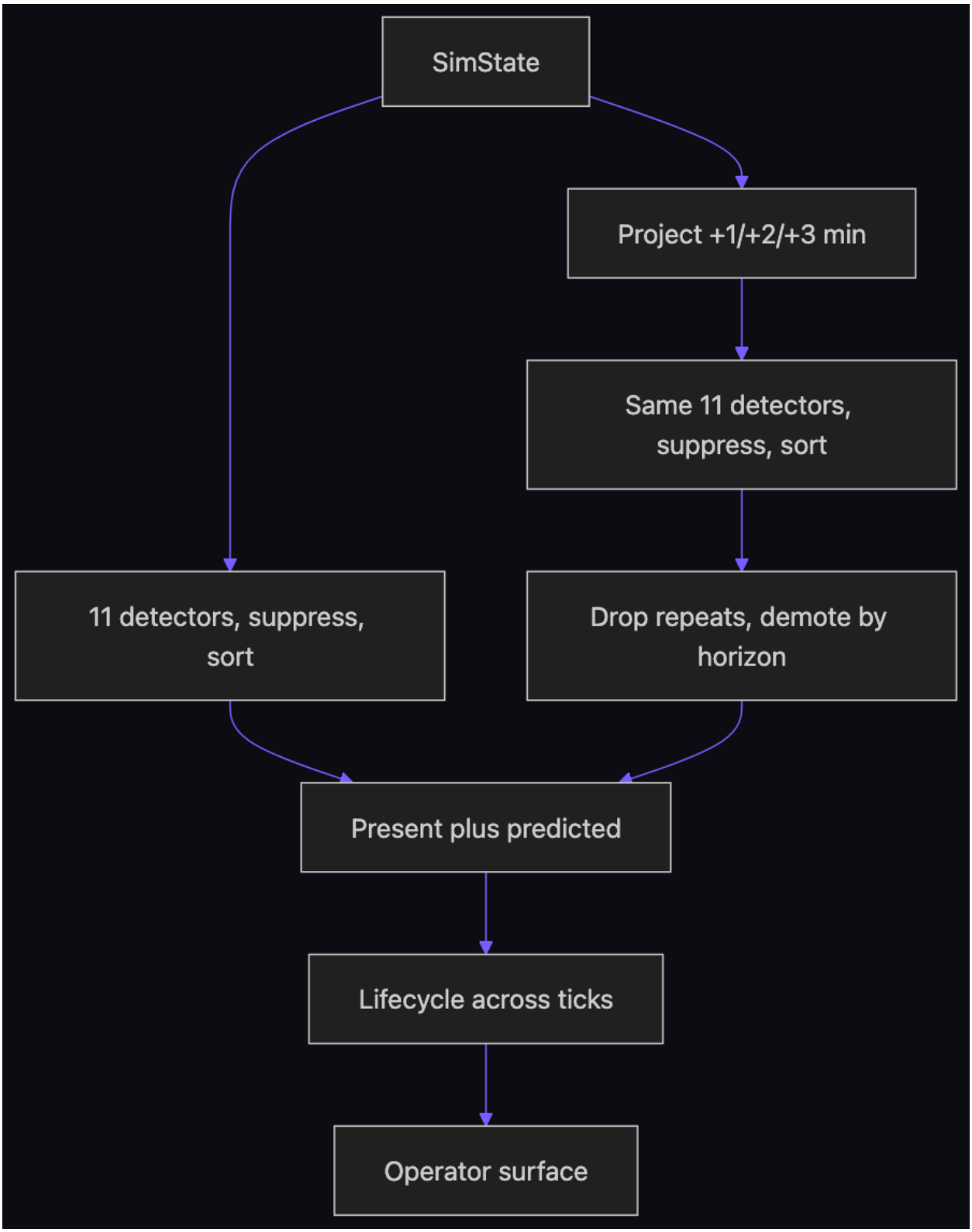

Figure 2: The orchestration pipeline. `SimState` *carries flights, runways, gates, one weather observation and a tick counter. The detector population runs unchanged on the present state and on each projection at plus one, two and three minutes, which hold heading and speed constant, advance arrival phase with the projected ETA and retire arrivals that land within the horizon. Only the orchestrator distinguishes them: it discards*

any alert id already reported at the present or a nearer horizon, and demotes projected severity one step at one and two minutes and two steps at three. Suppression marks lower-tier alerts as subsumed rather than deleting them, so a post-incident audit sees everything the detectors found, including what the display withheld. The lifecycle layer carries alerts across ticks within a grace period so a pair hovering at a threshold does not flicker. That layer measures its grace period in ticks, and until 2026-09-01 the live reducer never advanced one, so the difference between now and last-seen was always zero and nothing ever aged out: an alert absent for fifty consecutive polls was still held. The operator view was unaffected, because it filters stale alerts, which is why this survived; what grew without bound was the clearing list beside it, and the helper reporting how long an alert had been continuously true answered zero every time.

3.2 Contracts

Every detector presents the orchestrator with the same interface:

```
type Detector = (state: SimState) => Alert[];

type Alert = {
  id: string;
  severity: "critical" | "warning" | "advisory" | "info";
  category: string;
  title: string;
  detail: string;
  flightIds: string[];
  reason: string; // operator-facing explanation
  suggestedAction: string; // operator-facing remedy
  createdAtTick: number;
  lookaheadMin?: number; // set only when the alert comes from projected state
};
```

Detectors are pure functions over `SimState`. They have no side effects, no I/O, and no inter-detector communication. The orchestrator is responsible for any coupling: for example, suppressing a wake-spacing alert when a higher-severity runway-conflict alert already exists on the same runway.

3.3 Why centralised orchestration

A decentralised composition would require detectors to be aware of each other to avoid duplicate or contradictory advice. That awareness would couple them, defeating the maintainability argument. The orchestrator is the only component that needs to know the full detector population; detectors can be added, removed, or revised without touching the others.

3.4 Failure-mode profile

Failure	Single-algorithm system	Specialist Detector
Doctrinal change (e.g., FAA revises wake matrix)	Whole algorithm needs re-validation	One detector updated; others unaffected
Conflicting recommendations	Implicit; hard to detect	Explicit; orchestrator composes
Per-doctrine cost or latency analysis	Not separable	Per-detector telemetry available
Adding a new doctrine	Requires algorithm-wide change	Add one detector
Operator override	Override what?	Override at the doctrine that fired

4. Operational Properties

4.1 Determinism

All detectors are pure functions. The same `SimState` produces the same `Alert[]`. This property supports reproducibility, audit, and offline replay.

4.2 Explainability

Every alert carries a `reason` and a `suggestedAction`. The `reason` references the doctrinal source (e.g., "14 CFR §91.167 requires fuel to destination + alternate + 45 minutes at normal cruise consumption"); the `suggestedAction` is the operator-facing remedy. The operator can act on, override, or escalate any alert with the context needed to justify the choice in subsequent review.

4.3 Predictive look-ahead

The orchestrator projects state forward at multiple horizons (60, 120, 180 seconds) and runs the detector population against each projection. Alerts present in a projection but absent from the current state are tagged with their look-ahead horizon and rendered with a FORECAST indicator. This implements Endsley's level 3 in the situation-awareness model directly.

4.4 Severity tiering

Alerts are tiered. The current implementation uses three tiers actively: `critical` (immediate action required), `warning` (action required within the look-ahead window), `advisory` (operator attention required). A fourth tier (`info`) is defined in the type system but is not yet emitted by any detector in the current population; it is reserved for non-actionable observations (e.g., entry into a new sector). The tiering is set by the detector based on the magnitude of the rule violation. Tiering allows the operator surface to prioritise; tier `critical` alerts are visually distinguished and may produce auxiliary cues (audio, animation).

4.5 Observability

The system emits structured logs at three points per tick: ingestion (records the source, timestamp, and aircraft count), detection (records per-detector duration and alert count), emission (records the operator-visible alert set). The Cloudflare Pages Functions that proxy ADS-B and METAR upstream sources carry their own request logs. Together these produce an end-to-end trace suitable for incident reconstruction.

---

5. Implementation

5.1 Stack

The implementation is a single-page web application written in TypeScript on TanStack Start (a React-based framework). Live data is fetched via two Cloudflare Pages Functions that proxy `adsb.lol` (ADS-B aggregator) and `aviationweather.gov` (NOAA Aviation Weather Center). The application is deployed to Cloudflare Pages at `atc.fbritoferreira.com`. The source is licensed Apache-2.0 and public at `github.com/fbritoferreira/atc-decision-support`.

The choice of a browser-based client is deliberate: it lowers the barrier to inspection and reproducibility. The full system can be examined and exercised without installing anything.

5.2 Runtime topology

Figure 3 shows what runs where.

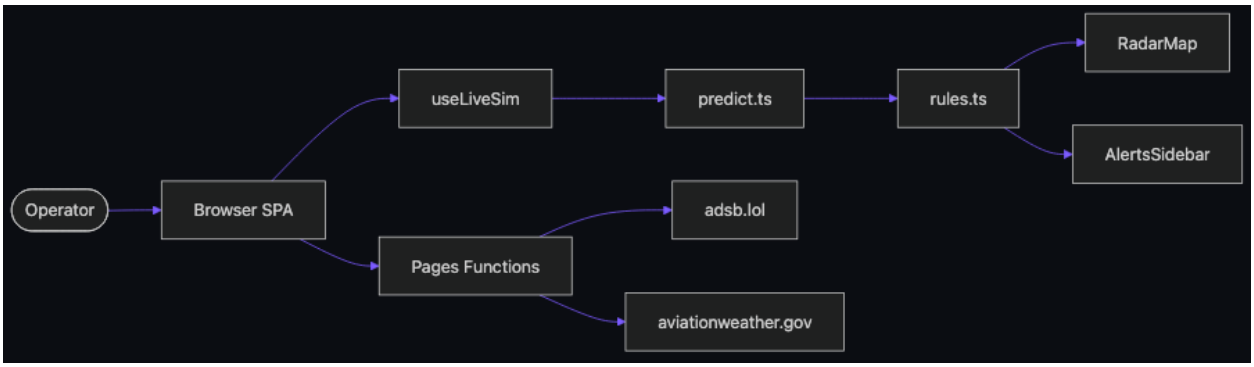

Figure 3: Runtime topology. Detection runs entirely in the browser, so one code path serves both the live feed and the frozen scenarios. The edge functions at `/api/adsb` and `/api/wx` exist only to cache and proxy the two upstream sources, ADS-B within 40 NM and METAR.

5.3 The orchestrator

The orchestrator is implemented as `runPredictiveRules(state: SimState)` in `src/sim/predict.ts`. It runs the detector population against the current state, then against three forward projections at +60, +120, and +180 seconds. The forward projection holds heading and ground speed constant, applies an 800 fpm descent profile to arrivals on approach and a 1500 fpm climb profile to departures once rolling, advances an arrival's phase with its projected ETA using the thresholds the engine applies tick by tick (approach inside ten minutes, final inside three), and removes an arrival whose projected ETA reaches zero, because a flight that lands within the horizon has left the traffic picture rather than arrived at an invented altitude. Projected alerts are de-duplicated against the current set; alerts unique to a projection are tagged with their look-ahead.

The last three of those behaviours are corrections, made after reading the projector against this section's own description of it, and each closed a projected state that could not physically occur. The climb previously applied to departures still queued for clearance, so an aircraft at speed zero on the ground was projected to 4,500 ft at three minutes without moving; the Tenerife scenario's two projected proximity advisories, cited in an earlier revision of Section 6.4, were artifacts of exactly this, two queued 747s lifted into the proximity detector's altitude filter by a climb they could not have flown, and they are gone from Table and text alike. Altitude previously clamped at zero, so an arrival projected past touchdown stayed in the population at 0 ft and the proximity doctrine's altitude filter silently dropped the aircraft closest to landing. And phase was copied while position, altitude and ETA moved, so every phase-gated doctrine was asked about states such as an arrival marked enroute two minutes from the threshold. The phase correction also cuts the other way: Linate now raises its runway conflict at the three-minute horizon as well as in the present, because the projected arrival reaches final within the horizon and arms a gate that phase-copying had kept cold. One artifact removed and one true forecast surfaced, by the same change.

5.4 The detector population

Eleven detectors are implemented. Each operationalises a specific doctrinal concern, and the count is checked against the orchestrator's own composition by `apps/atc/scripts/verify-paper-claims.mjs`, because this sentence has been wrong twice: it said seven when the population had reached eleven, and a reader met it immediately above a table listing all eleven.

Detector	Doctrinal source	Trigger	Caveat
Runway conflict	ICAO Doc 4444 §7.9; FAA JO 7110.65 §3-9 (Departure) and §3-10 (Arrival)	Two or more aircraft assigned the same runway in conflicting phases (e.g., arrival on final and departure on roll)	,

Detector	Doctrinal source	Trigger	Caveat
Wake-turbulence spacing	FAA JO 7110.65 §5-5-4 and TBL 5-5-2 (Consolidated Wake Turbulence, in effect for US ATC since 2022; the standalone JO 7110. The minima themselves were verified against the order on 2026-08-31, which had not been done before: the published table was fetched from faa.gov and compared cell by cell, all 81 agree, and nine representative types resolve to the categories TBL A-1 assigns them. A test pins the comparison, so a future edit to the matrix has to argue with the published minima. The table is not monotonic, since a category D leader requires 6 NM ahead of a category H where a category C leader requires 5, and that is doctrine rather than a transcription error: C is Pairwise Lower Heavy and D is Non-Pairwise Heavy, and a pairwise category exists because that pairing was studied and earned reduced separation.126 series is cancelled and 7110.65 is the citable authority)	Trailing aircraft below the separation the order requires for the pair, which is the CWT minimum for the leader and follower's categories where the table has one and standard terminal radar separation where it is blank. Both raise this category, which is why most alerts here are against a radar floor rather than a wake minimum, as Section 6 reports	The implementation uses the nine-category CWT matrix (A through I), with categories assigned from the ICAO type designator via the FAA's published table of 161 types. A blank matrix cell means no wake minimum applies and radar separation governs (3 NM, or 2.5 NM for aircraft established on final within 10 NM). The 2.5 NM case carries more conditions than that clause implies, and they are worth stating because none is observable here: JO 7110.126B paragraph j also requires FUSION or single sensor slant range mode, the aircraft within 40 miles of the antenna, a documented average runway occupancy time of 50 seconds or less, operational CTRDs, and turnoff points visible from the control tower. The last two are an administrative record and a property of the tower building, so no feed carries them and no prototype can know which floor applies at a field at a moment. The same paragraph requires wake separation to be applied per TBL 5-5-2 regardless, so the indeterminacy bites on the radar floor alone and leaves every wake requirement untouched. A type absent from the FAA table also falls to the radar floor rather than to a guessed category, and the sampling harness counts the unmapped rate: measured at 15% of the aircraft appearing in wake pairs across a

Detector	Doctrinal source	Trigger	Caveat
Gate conflict	Local ground-operations doctrine (airport-specific)	Inbound arrival with an occupied or double-booked gate assignment	Gate state is operator-supplied; the live ADS-B feed does not carry gate assignment, so this detector is exercised only by the historical / synthetic scenarios. The stand model behind it is synthetic too, which the doctrine column overstated until 2026-09-01: stands are generated per terminal from a declared size, so every stand in a terminal carries the same wake rating, and no detector reads that rating. The detector is airport-specific only in that stand counts and terminal names differ per field. No local ground-operations rule is encoded, and the ratings are not a stand register: Porto's terminals declare Super stands, and no A380 has operated there.

Detector	Doctrinal source	Trigger	Caveat
Fuel hold	14 CFR §91.167 (fuel to destination + fuel to alternate + 45 minutes at normal cruise consumption, IFR)	Remaining fuel below the §91.167 requirement at projected destination ETA	ADS-B carries no fuel state; this detector is exercised by the historical / synthetic scenarios where fuel state is known from the official report. That sentence understated the position until 2026-09-01, and the stronger version is this: the live ingest does not leave the field empty, it synthesises one, giving every arrival a figure of ETA plus 60 minutes. Since the doctrine fires below ETA plus 45, the synthesis guarantees a constant 60-minute margin against a 45-minute threshold, so the detector is arithmetically incapable of firing on live data rather than merely unexercised by it. The figure also reached the flight strip and the detail card, rendered in the same style as the measured altitude and ground speed beside it, so a viewer could not tell an invented number from a read one; it is shown only where fuel is genuinely known now. In a live deployment it would be exercised only by aircraft transmitting fuel state in MASPS-compliant messages or by a pilot-side data link.

Detector	Doctrinal source	Trigger	Caveat
Crosswind advisory	Type-certificate demonstrated crosswind components, applied as one representative value rather than per type	Computed crosswind component on the runway exceeds 25 kt, a single constant standing for narrow-body and most heavy limits; the system carries no per-type certificate data, so this is the one threshold in this table that is a chosen value rather than a figure read from a document	This is an *advisory*, not an enforcement detector; the type limit is enforced by the pilot in command, not by ATC. The detector surfaces the condition so the orchestrator can prompt runway-reassignment coordination.
Cascading delay	Operational throughput heuristic relative to the runway acceptance rate (RAR) for that field	Three or more arrivals sharing a five-minute bucket on the same runway. The threshold is a constant, not a rate: this row described it until 2026-09-01 as computed per-runway from OurAirports data and FAA-published runway acceptance rates, and no acceptance-rate data exists anywhere in the source. The alert's own detail line has always said runway throughput of about thirty an hour, which is the same constant stated plainly	An absolute "three arrivals in five minutes" threshold is normal operation at KATL, KORD, and KJFK; the implementation uses runway-relative thresholds and reports a default if RAR is not configured.

Detector	Doctrinal source	Trigger	Caveat
Proximity conflict	DO-185B sensitivity tables (coarse approximation)	Warning: two aircraft within 2 NM horizontal AND 1,000 ft vertical. Critical: converging, with *tau* (time to closest point of approach) inside 45 s, horizontal miss distance at that point inside DMOD 0.5 NM, and vertical separation already inside 200 ft. A non-converging pair never escalates past warning whatever its spacing, and a non-converging pair attributed to distinct parallel runways demotes to advisory	The critical tier is the same family of test TCAS runs, not a distance box; an earlier revision of this row described it as a distance box, which stopped being true when the convergence test was added. Two differences from airborne TCAS remain: the vertical test is instantaneous, because this ingest does not read the vertical rate the feed supplies, and the thresholds are single values rather than DO-185B's altitude-banded sensitivity levels. Appropriate for ground-side advisory use and inappropriate for any onboard role.
Weather shift	FAA JO 7110.65 §2-6 (weather information); CAT I minima as the reference point	Adverse conditions or an unstable wind: IFR at visibility below 3 NM or a ceiling below 1,000 ft, LIFR below 1 NM or 500 ft, and below CAT I minima at 0.5 NM or 200 ft. Advisory for IFR, warning for LIFR or below minima	Named for a shift it cannot see. The feed gives one observation at a time with no history, so this reports a standing condition rather than a change in one, and the name predates that recognition
Runway surface	Local ground-operations doctrine; braking-action reporting	A runway whose surface state is other than dry with traffic assigned to it. Advisory when wet, warning when snow or ice	Surface state is not carried by ADS-B or METAR in a form this ingests, so it is set per scenario and never fires on live data. A standing condition, so it does not escalate to critical

Detector	Doctrinal source	Trigger	Caveat
Runway identity	FAA JO 7110.65 §3-9 and §3-10 (takeoff and landing clearance); NTSB AAR-07/05	An aircraft above 40 kt whose track is outside 20 degrees of its cleared runway's course. Critical. This row said "while matching another runway's course" until 2026-09-01, which is what the detector's comment intends and not what it does: matching another runway changes the alert's wording and is not required to raise it	Motivated by Comair 5191, cleared for runway 22 at 7,003 ft and rolling on runway 26 at 3,501 ft, a 37-degree difference checkable from the moment thrust was applied. Critical despite the standing-condition rule, because a wrong-runway roll is an event rather than a condition. Compares true course against true course since 2026-08-29; it previously compared a true track against a magnetic heading
Emergency squawk	ICAO Annex 10 and FAA JO 7110.65 §10-2 (emergency codes)	Transponder 7500 unlawful interference, critical; 7600 radio failure and 7700 general emergency, warning	The radar surface rendered these codes from the first version while nothing in the alert layer covered them, so an emergency showed on the scope and never entered the alert ledger, the suppression logic or the lifecycle. It reads a code the crew sets: American 11 is in the corpus precisely because no code was set

Each detector is a pure function. One thing the per-incident counts do not separate: an alert raised during a reconstruction need not concern the event being reconstructed. Two of the nine carry a fuel-hold warning incidental to the accident, the Los Angeles runway collision of 1991 and the 2025 Potomac midair, because both place an arrival on short final with less remaining fuel than the destination, alternate and forty-five minute reserve the regulation requires. The alerts are doctrinally right and unrelated to what happened, so a reader asking whether the system detected the event should look at the category rather than the count. Encoding JFK from its report also exposed a rule that cannot separate the two cases it exists to separate. The proximity detector downgrades a pair assigned parallel runways to an advisory, deciding on the runway identifiers and never on the headings, so an aircraft deviating toward the traffic on the parallel runway raises what two aircraft flying their own localisers raise. The behaviour is pinned by test and deliberately not fixed here; requiring convergence before the downgrade applies is a doctrine change that wants a live measurement first. Two of the nine were also reconstructions of events they did not match, found by checking every incident's citation against its source after a wrong citation surfaced in this paper's opening. LaGuardia carried a flight number belonging to an unrelated accident and the wrong runway; JFK was encoded as two departures with the reported 350-foot separation treated as horizontal, when the event was two Embraer 175s on parallel approaches to 31L and 31R separated by 350 feet vertically. Both now follow the reports, and the JFK scenario drops from critical to advisory as a result, because 350 feet clears the critical gate and parallel approaches are downgraded deliberately. No scenario in the corpus now raises a critical proximity conflict. For a while nothing stated which alert is the detection for each incident. The suite pinned the top severity

each one reaches, which is a weaker thing, and an earlier version of this sentence claimed otherwise. It now declares all three classes per incident, detection, related and unrelated, and checks the declaration against what the detectors raise. The proximity-conflict detector is reproduced in Appendix A.3 as a representative example.

The measurement the downgrade needed, first result. The parallel-runway downgrade decides on runway identifiers and never reads geometry, and the heading-disagreement gate proposed to fix it turned out to fire on ordinary simultaneous parallel approaches. The quantity that separates the two cases, each aircraft's distance from the centreline of the runway it is assigned, was not recorded until 2026-09-02. Three windows have now collected it, and 67 proximity pairs carry it for both aircraft, 30 of them on parallel runways at Kennedy and Miami.

Aircraft in those pairs sit on their centrelines. The worst deviation in a pair has a median of 0.030 NM across the parallel subset and 0.000 NM across all 67, a ninetieth percentile of 0.070 NM, and a maximum of 0.330 NM. A gate at half a nautical mile would escalate none of the 30; at 0.2 NM it escalates four. That is the property the heading gate lacked: routine traffic does not trip it, so a rule keyed to centreline deviation has room to fire only on an aircraft that has actually left its approach.

Two things this does not establish. The sample contains no deviation event, so it measures the baseline rather than the detection rate, and nothing here shows the rule would have caught the event that motivated it. And 67 pairs from two airports over three windows is a first look, not a basis for changing a suppression rule that fires on forty per cent of proximity pairs. The doctrine stays as it is until the sample is larger and contains something to detect.

5.5 Data sources

- **ADS-B traffic.** `adsb.lol` open ADS-B aggregator, queried via the project's Cloudflare Pages Function proxy. Radius queries return all aircraft within 40 NM of the airport reference point. This said bounding box until 2026-09-01 and the two are not the same shape: a 40 NM box reaches past 56 NM at its corners, where a radius does not. The client builds the aggregator's radius endpoint, which is also why every record arrives carrying its own distance. The browser polls every 20 seconds and the proxy holds one copy per airport for 25 seconds, shared by every viewer, so the picture on screen is up to 25 seconds old in ordinary operation. The cache window is longer than the poll interval deliberately: at 15 seconds every poll expired the copy and went upstream, which defeated the purpose of caching at all. When the upstream refuses, a copy up to ten minutes old is served and labelled as stale on the operator surface rather than being replaced by an error, and its age is measured from the upstream fetch rather than from the browser's request.
- **METAR weather.** `aviationweather.gov` (US NOAA Aviation Weather Center) METAR API, queried via the project's Pages Function proxy. Refresh cadence: 300 seconds.
- **Route lookup.** `adsbdb.com` open callsign-to-route service, used on demand when the operator opens a flight's detail card.
- **Airport static data.** Runway threshold coordinates come from the FAA NASR 28-day subscription (the FAA's surveyed end positions, public domain) for the ten US airports, and from the `OurAirports` open-data CSV elsewhere; cross-checking the two found OurAirports missing KORD 09R/27L's 2021 extension by 3,589 ft, which is why the authoritative source matters. Runway headings, lengths, and airport reference points are bundled with the application for seventeen airports: KJFK, KLAX, KSFO, KORD, KATL, KDEN, KDFW, KSEA, KBOS, KMIA (United States); OMDB (Dubai), EGLL (London Heathrow), LPPR (Porto), LPPT (Lisbon), EPWA (Warsaw Chopin), EPKK (Kraków), EPPO (Poznań). Coverage of further fields is straightforward to extend by adding the airport's reference point and runway list.

5.6 Operator surface

The operator surface is composed of: a radar map with concentric range rings, compass cardinals, runway depictions, and aircraft datablocks (callsign / type / flight level / ground speed); a flight-strips panel listing arrivals and departures sorted by estimated time of arrival; a runway-gate-weather panel showing the live

METAR; and an alerts sidebar listing the current detector output with severity, reason, and recommended action.

Mouse-wheel zoom, click-to-inspect, and a weather overlay toggle are provided. Aircraft squawking the international emergency codes (7500 hijack, 7600 lost communications, 7700 general emergency) are visually flashed with a pulsing red halo on the radar.

5.7 Live-mode airport coverage

Live operation is supported for seventeen airports via the route `/live/{ICAO|IATA}`. Both ICAO (KJFK) and IATA (JFK) codes resolve. Each airport carries its own runway geometry; the same detector population is reused without modification, with one airport-specific exception that is a defect rather than a design choice. Runway attribution rejects any aircraft above 5,000 ft, and the altitude it tests is height above sea level rather than height above the field. KDEN sits above that cutoff, so an aircraft standing on its runway is already excluded and attribution never fires there at all, which leaves the runway-conflict detector and the parallel-runway demotion inert at that field. It is pinned by a test rather than corrected: the gate should measure height above the field, and no field elevation exists in this repository, so the fix is a data change recorded with the regeneration notes rather than a constant raised to make one airport work. That sentence was true of the registry and false of the live view until 2026-09-01, which is the kind of gap a claim about data cannot catch on its own. The live hook built its runway list, stand list and sector label in the initialiser React runs once per mount, while the polling effects re-ran whenever the airport changed, and the route renders the dashboard without a key. Choosing another airport from the picker is a client-side navigation, so the same component instance survived: the new airport's traffic arrived and the previous airport's runways stayed. Runway attribution, the runway-conflict detector and the crosswind check all read that list, so all three would have been reasoning about the wrong field. The repair is in the reducer rather than the route, because a key placed at the call site is invisible to the next call site, and it is covered by tests that switch between airports with different runway counts.

---

6. Diagnostic Shape Check Against Historical Incidents

6.1 Why this is a shape check, not a validation

This section deliberately avoids the word *validation* in its title because the exercise that follows does not validate the system in the strict sense the term is used in safety-critical software. A real validation would require: a substantial corpus of nominal-operations data with a measured alert rate; one or more documented near-misses that resolved safely, where the system must distinguish them from the incidents that did not; a baseline against which the architectural choice can be measured; and inter-rater agreement on what counts as "matching the principal cause." None of these are present here.

What follows is therefore a *diagnostic shape check*. The author encoded the position, weather, and known phase of eight publicly-documented incidents into the simulator and recorded which detector or detectors fired. The exercise tests whether the doctrinal rule set produces the alert kind that the responsible investigation bureau identified as the principal cause. It does not test whether the rule thresholds are correctly tuned, whether the detector population is complete, or whether the system would be useful in live operation. Sections 6.7, 6.8, and 6.9 report the false-positive analysis, the negative-control corpus, and the baseline comparison. Those measurements are done and they narrow the claim rather than settling it: what remains open is threshold work against the measured corpora, a broader negative-control set beyond the three ASRS records now encoded, and review by someone who holds or has held a control position.

A further limit on what any scenario corpus of this kind can establish emerged from two separate defects, and it belongs here rather than in a footnote. Scenarios are written in the full vocabulary the data model declares; a live feed produces only the subset its classification heuristics can assign. Where those diverge, the divergence is invisible to both the tests and the live measurement, because the tests never see live labels and the measurement has no expected output to disagree with. It has now happened three times. Inferred runway labels did not match the paired-strip identifiers the registry stores, so two corrections passed every

scenario and executed on no live traffic. And the proximity detector excluded aircraft in the `departed` phase, which live ingest assigns to every departure it classifies, so a whole traffic class was unreachable live while the scenarios that would expose it encoded the phase differently; that exclusion was removed once the sampling window it would have disturbed had closed. The third instance is not a label but a value, and it widens the limit. Ingest synthesises a fuel figure from each arrival's ETA such that the fuel doctrine's threshold cannot be crossed, and the smoothing layer then blends the ETA against the previous tick, before the detectors run, and leaves the fuel figure alone. The invariant holds where it is established and not where it is read. Six places recorded the doctrine as unreachable on live data on the strength of the first half, and the module that performs the smoothing recorded nineteen criticals measured in one Los Angeles window. So the divergence need not be in the vocabulary: any value a detector reasons about can be consistent at ingest and inconsistent by the time a rule reads it, and a scenario corpus cannot show it, because scenarios set the value directly and never run the synthesis. `apps/atc/docs/phase-vocabulary-audit.md` now records which of the nine declared phases live ingest can produce and how each detector's gate fits that subset, with a test pinning the reachable set by sweeping the input space. The general statement is that a passing corpus bounds the doctrine, not the deployment, and that the gap between declared and produced vocabularies is a place to look first rather than last.

6.2 Method

Nine publicly-documented aviation incidents are encoded as scenarios in the implementation, eight as detection shape checks and one, American 11 on 11 September 2001, as a documented blind spot the system is tested to NOT see. Each scenario fixes the simulation clock at the moment immediately preceding the event, places the aircraft at the positions established by the official investigation, sets the weather to the conditions reported by the official METAR for that timestamp, and freezes time (`speed: 0`) so the detector output can be inspected without temporal evolution.

The principal cause identified by the responsible investigation bureau is recorded for each scenario. The detector output is compared against that cause.

6.3 The corpus

The corpus is organised into a primary US set and an international supplementary set. The primary set (six entries) is drawn from incidents investigated by the US NTSB and FAA and is the basis for the thesis's claims about US-specific applicability; it covers airports in the New York Bracon (KJFK, KLGA, KDCA-adjacent) and one West Coast field (KLAX). The international supplementary set (two entries) provides longer-baseline coverage from the historical record (Tenerife 1977, Linate 2001); these incidents are doctrinally instructive but their investigations were conducted by non-US authorities and they are not relied on for any US-priority claim.

Primary US corpus

ID	Year	Location	Source	Principal cause (as published)	Highest-severity alert produced
Avianca 052	1990	KJFK approach	NTSB AAR-91/04	Fuel exhaustion after three holding patterns; crew failed to declare emergency	Fuel hold, critical

ID	Year	Location	Source	Principal cause (as published)	Highest-severity alert produced
USAir 1493 / SkyWest 5569	1991	KLAX	NTSB AAR-91/08	Position-and-hold clearance issued to SkyWest on 24L while USAir was cleared to land same runway	Runway conflict, **warning**
Potomac mid-air	2025	KDCA approach	NTSB DCA25MA108	Commercial CRJ on final to runway 33 collided with US Army H-60 on helicopter route 4	Proximity conflict, **warning** (see §6.4: the pair misses the critical tier by 0.006 NM of miss distance)
LGA aborted takeoff	2025	KLGA	FAA and NTSB investigations opened; no final report	American Eagle 4736, an Embraer 175 operated by Republic, was cleared for takeoff on runway 13 while United 2657 was still on it after landing	Runway conflict, critical
JFK near-miss on parallel approaches	2026	KJFK approach	FAA investigation opened April 2026	An Embraer 175 deviated from the 31L approach course toward another on the parallel approach to 31R; 350 ft vertical and about half a mile horizontal	Proximity conflict, advisory

ID	Year	Location	Source	Principal cause (as published)	Highest-severity alert produced
Comair 5191	2006	KLEX departure	NTSB AAR-07/05	Takeoff attempted from unlit runway 26 (3,501 ft) instead of assigned runway 22 (7,003 ft); crew failed to use available cues to confirm position; 49 of 50 died	`runway-identity` critical: heading on the roll (264°) matched runway 26, not assigned 22 (227°); flagged inside the survivable rejection window

International supplementary corpus

ID	Year	Location	Source	Principal cause (as published)	Highest-severity alert produced
Tenerife	1977	GCXO / Los Rodeos (ICAO; IATA TFN)	Spanish Subsecretaría de Aviación Civil / CIAIAC	Two heavy 747s on the active runway in fog; KLM commenced takeoff roll while Pan Am was backtaxiing	Runway conflict (multi-aircraft on runway), critical
Linate disaster	2001	LIML	Italian ANSV	Cessna entered runway 36R in fog while SAS MD-87 was on takeoff roll	Runway conflict, critical

In every case the alert category the system produces corresponds to the principal cause identified by the official investigation, which is the claim the shape check makes; the severity does not always reach critical, and the corpus table above records the two cases where it stops at warning. The diagnostic-shape claim of Section 6.1 is therefore supported for both the US-specific corpus and the international supplementary set. The strict-sense validation work outlined in Sections 6.7–6.9 remains open.

A known-blind-spot case: American 11, 2001 The corpus gains a third scenario class beyond shape checks and negative controls: a case the population is asserted NOT to see. American 11 is reconstructed at 08:30 EDT from the 9/11 Commission Report and the NTSB flight path study: transponder off since 08:21, a 100-degree route deviation, descending toward Manhattan, with United 175 still on its normal profile. The measured and test-pinned output is zero alerts.

That silence is the finding. None of the four hijacked aircraft that morning squawked 7500; the detectable signatures were transponder loss and flight-plan deviation, and this population holds no squawk history and no flight plans. Live mode is blind one level deeper: an aircraft without a transponder does not appear in

ADS-B at all, because ADS-B is the transponder. On 2001 signatures this system is exactly as blind as the 2001 system was, and the scenario exists so that fact is documented and enforced by test rather than left for a reviewer to discover. Closing the blind spot requires primary-radar correlation and route-conformance monitoring, both future work.

The case did motivate one real doctrine: `squawk-emergency` (7500 critical; 7600 and 7700 warning), covering the codes a crew sets. The radar surface had rendered these codes from the first version; nothing in the alert layer covered them. A synthetic drill scenario exercises the detector, kept deliberately separate from the AA11 reconstruction, where no such code was ever set.

6.4 Case discussion: Potomac mid-air (2025)

The 29 January 2025 collision between PSA/American Eagle flight 5342 and a US Army H-60 helicopter on the Potomac helicopter route is the most recent and most operationally relevant of the corpus. The NTSB's final report on DCA25MA108, NTSB/AIR-26/02, adopted 28 January 2026, gives the probable cause as the FAA's placement of a helicopter route close to a runway approach path, its failure to review that route and the data it generated, its failure to act on recommendations to mitigate the hazard, and the helicopter crew's ineffective use of pilot-applied visual separation. The count of prior encounters is from the urgent safety recommendation report of 11 March 2025 rather than from the final report: 15,214 close-proximity events between commercial aircraft and helicopters at the airport from October 2021 to December 2024, measured as lateral separation under one nautical mile with vertical separation under 400 feet, against 944,179 commercial operations. Both papers cited that figure to the final report until 2 September 2026, which named the wrong document by eleven months and left the threshold undefined.

In the reconstruction, the CRJ is placed at 350 ft on a one-mile final to runway 33; the H-60 is placed at 200 ft on a southeast track over the eastern shoreline of the Potomac. The proximity-conflict detector fires immediately, with horizontal separation 0.67 NM and vertical separation 150 ft.

It fires at warning, not critical, and the margin is six thousandths of a nautical mile. This was found on 2026-08-25 by checking the corpus table's severity column against detector output for the first time; the table had claimed critical since it was written. Measured on the encoded pair: vertical separation 150 ft, inside the 200 ft envelope; closing speed 3.81 NM per minute giving a time to closest approach of 6.9 seconds, far inside the 45-second gate; and horizontal miss distance at closest approach of 0.506 NM against a DMOD of 0.500 NM. Two conditions pass and the third fails by about 36 feet.

Two conclusions follow, and the second is the reason this is not fixed by nudging a coordinate. A verdict decided by 36 feet is decided by encoding noise rather than by doctrine: these positions were placed by hand from the investigation's narrative and are nowhere near that precise, so claiming this scenario demonstrates the critical tier firing on the accident would rest on a coincidence of hand-placement. More seriously, **the encoded tracks miss each other by half a nautical mile, and the actual event was a collision**, so the reconstruction reproduces the accident's proximity without reproducing its collision geometry. That was invisible while nothing measured the miss distance. The correct repair is to encode the positions and tracks the NTSB's final report publishes, which would produce a converging pair with a miss distance near zero and a critical that fires on doctrine; that is a data-sourcing task rather than an editing one, and adjusting a coordinate until the tier changes would be fitting the scenario to a desired output. A test now pins the measured severity and the two passing conditions, so a change to the geometry or to DMOD announces itself rather than quietly flipping the flagship case. See `apps/atc/docs/potomac-severity-margin.md`. The critical tier does compute *tau* and requires the miss distance at closest approach to fall inside DMOD, so it is the same family of test as an airborne RA rather than a distance box; it is still not equivalent to one, because the vertical test is instantaneous and the thresholds are not altitude-banded, and it remains a ground-side advisory. The `reason` cites the simplified threshold and the helicopter-corridor contributing factor; the `suggestedAction` recommends immediate vector or altitude change and re-routing of helicopter traffic away from the approach corridor.

The H-60 at 200 ft AGL is inside the regime in which onboard TCAS issues Traffic Advisories but inhibits Resolution Advisories. A ground-side advisory layer is precisely the gap an architecture of this kind addresses:

by surfacing the projected conflict to the controller rather than relying on an onboard RA, the system targets a regime that current onboard equipment by design does not cover.

This reconstruction depends on an encoding choice, and that dependency was invisible until 2026-08-25. The H-60 is encoded as a departure in the `enroute` phase. The proximity detector filters out any aircraft whose phase is `departed`, which in this model means airborne and climbing out, and live ingest assigns exactly that phase to every departure it classifies. Re-encoding the same helicopter as `departed`, which is what a live feed would supply for a departing rotor, produces no proximity-conflict alert at all: the reconstruction goes silent on the category the NTSB named, and the change is a phase label rather than any geometry. Two negative controls were passing for the same unrelated reason, including one written specifically to exercise the 1,000 ft vertical boundary. Three tests now pin both halves of the behaviour, the defect is recorded in `apps/atc/docs/departure-proximity-blindspot.md`, and the one-line fix waits for the current sampling window to close because it adds a traffic class to the highest-volume detector and the effect on that volume is unmeasured. The finding came from auditing whether each negative control exercises the doctrine it asserts silence on, which is the third time this corpus has been found asserting silence without the doctrine running, and the first time the audit reached inside a detector's own filters.

This is consistent with the corrective measures the NTSB recommended to the FAA. The system does not, and is not represented to, prevent the collision. It reaches the conclusion the official investigation reached, with the same inputs the controllers had.

6.5 Case discussion: Avianca 052 (1990)

Avianca 052 entered three holding patterns over the New York TRACON in deteriorating weather on 25 January 1990. The crew did not communicate the severity of the fuel state to ATC in standard terminology. The aircraft exhausted fuel during the final approach attempt and crashed in Cove Neck, Long Island. The NTSB cited fuel exhaustion compounded by the absence of a clear emergency declaration.

In the reconstruction, AVA052 is placed at 3,000 ft on approach with 8 minutes of fuel remaining and an ETA of 4 minutes. Three aircraft are held above. The synthetic-state fuel-hold detector fires `critical` with `reason` citing 14 CFR §91.167 and `suggestedAction` recommending prioritisation for direct approach. The cascading-delay detector fires `advisory` on the holding stack. The diagnostic shape of the output therefore corresponds to both the principal cause (fuel exhaustion) and the contributing factor (capacity-induced holding) identified by the NTSB. Two caveats accompany this finding. First, as recorded in Section 5.4, the fuel-hold detector depends on a fuel-state input that ADS-B does not carry, so this scenario exercises only the synthetic side of the detector and not a live-data path. Second, the NTSB also identified crew communication and phraseology as a contributing cause; the detector population in this work has no representation for phraseology and would not flag that contribution.

6.6 Coverage and de-duplication

Coverage across the corpus is measured by `apps/atc/scripts/baseline-compare.mjs`, which runs every scenario through the detector population and records the category and severity of each alert. The measured figures correct three statements previously made in this section.

The `Alert` type declared nine categories when this section was first verified, of which two, `weather-shift` and `missed-handoff`, had no emitter. Both are resolved, in opposite directions. `weather-shift` is now implemented (Section 5.4). `missed-handoff` was removed from the type: `SimState` models a single sector through `sectorId`, so there is no inter-sector handoff to miss and no honest way to emit it, and a declared category that nothing produces promises the reader coverage that does not exist. Reinstating it belongs with a multi-sector model. The type has since grown to eleven, gaining `runway-surface`, `runway-identity` and `squawk-emergency`, and all eleven have emitters, which the suite asserts on every run in "all eleven declared categories are reachable", checking each against the alerts the corpus produces through both the predictive wrapper and the baseline. Reachable is weaker than exercised, and the same is true of the projection branch, measured on 2026-09-01 rather than described: across the twenty-eight scenarios the look-ahead pass produces three alerts that the present state does not, in three scenarios, and all three sit at the furthest of the three horizons. The one-minute and two-minute projections never produce an alert the present tick

has not already raised. The projection itself is doing real work, which was checked rather than assumed: at the three-minute horizon it moves aircraft between ten and twenty-two nautical miles in the scenarios sampled, so the thin yield is a property of a corpus whose conflicts are already present at load rather than of a projection that barely advances the picture. That is not an argument against the branch, whose value is meant to show on live traffic rather than on scenarios built to be alarming now, but a corpus that exercises it three times is thin evidence for it and the reader should have the number. The category distribution is uneven in the same way: three categories fire in exactly one scenario each, `cascading-delay`, `runway-identity` and `squawk-emergency`, while `fuel-hold` fires in eight. A doctrine resting on one example is tested against that example rather than against a class, so for those three the corpus shows that they can fire and says little about when they should not. The suppression layer is thinner still: across the corpus it marks one alert of fifty-one as subsumed, in the single scenario its own test uses, against the 14.9 per cent of live alerts the false-positive analysis reports arriving pre-subsumed. So the layer is measured at scale on live data and exercised by one example in the corpus, and the corpus is what a reader can re-run. `scripts/baseline-compare.mjs` reports the same population per category when it is run, but nothing runs it: no test invokes it, no CI step calls it, and this sentence credited it with an assertion it only makes by hand. This sentence said eight until 2026-08-29, having been written when the type held eight and left alone through three additions; the assertion it cites had covered all eleven the whole time, so the code was right and only the count describing it was wrong.

Per-detector ablation. The category split counts what each detector produced, not what would be missing without it. Removing each in turn and re-running the rest through suppression, across the 28 scenarios and the 51 alerts the full population raises:

Detector	Alerts lost	Scenarios downgraded	Scenarios silenced
fuel-hold	16	2	0
runway-conflict	7	4	1
weather-shift	6	0	0
proximity-conflict	6	4	4
gate-conflict	4	1	1
crosswind	4	1	0
runway-surface	3	0	0
wake-spacing	2	2	0
runway-identity	1	1	1
squawk-emergency	1	1	1
cascading-delay	1	0	0

Volume and detection value come apart. Fuel-hold raises just under a third of the corpus and costs two scenarios a tier; proximity-conflict raises six and is all that stands between four scenarios and silence. Weather-shift, runway-surface and cascading-delay never change a tier when removed: they carry context rather than detection. No removal reveals an alert that suppression had hidden, because suppression drops none of the 51 in this corpus, for the reason given with the withdrawn in-trail doctrine. Ablation and the split agree here, and they agree for that reason rather than by coincidence. Produced by `scripts/ablation.mjs`.

Multi-detector firing is narrower than first reported, in one case and not the other. Avianca 052 produces four `fuel-hold` alerts and no cascading-delay alert, so the pairing claimed for it does not hold. Tenerife does now fire two categories, `runway-conflict` and `weather-shift`, which is the output previously claimed for it; that claim did not hold when it was made, because the low-visibility condition reached no detector until `detectWeatherShift` was written. The distinction matters: the claim was correct about what the architecture should produce and wrong about what the code produced, and it was made true by building the missing detector rather than by restating the measurement.

The de-duplication described here is implemented, in `runPredictiveRules` (`src/sim/predict.ts`): a projected alert is discarded if its identifier already appears in the current-state set or in an earlier projection horizon. Its observable effect on the corpus is to add alerts rather than remove them, because the projections

surface conditions that are not yet present. Linate goes from two alerts to three once the +1/+2/+3 minute projections are included, the addition being a projected `runway-conflict` at the three-minute horizon. An earlier revision cited Tenerife here, gaining two projected `proximity-conflict` advisories; those were artifacts of the projector climbing queued aircraft, a defect Section 5.3 reports, and the corrected projector does not produce them.

6.7 False-positive analysis

A meaningful validation requires that the detector population be exercised on a substantial sample of *nominal* operations and the resulting alert rate be reported. The implementation includes an offline analysis script (`apps/atc/scripts/fp-analysis.mjs`) that polls live ADS-B at a configurable interval, runs a subset of the detector population on each snapshot, and writes per-detector counts to a CSV.

Withdrawal of the first reported sample A 3-minute sample taken at KJFK on 27 May 2026 was previously reported in this section as Table 1: nine snapshots, a mean of 183 aircraft, 25.2 alerts per snapshot, 218 of the 227 alerts in the proximity-conflict category. Those figures are withdrawn. They did not come from the system this thesis describes.

The script that produced them defined its own `countAlerts()` function containing inlined re-implementations of two detectors, despite a docstring stating that it imported the live rule set. The reported numbers therefore characterise that re-implementation. Re-running the same sampling method against the production detector population inverts the two headline results: the real proximity-conflict detector fires roughly one alert per snapshot rather than 218 across nine, and `wake-spacing`, reported as never firing, is the most frequent category at roughly 2.7 alerts per snapshot.

Two of the three observations drawn from the withdrawn table are therefore unsupported. The claim that the proximity threshold is too loose for the NYC TRACON rested on alerts the system does not emit. The claim that wake, fuel and gate detectors fire zero times because ADS-B does not carry the necessary data is wrong for wake spacing: `wakeFromType` in `src/sim/wake-lookup.ts` infers a wake category from the ADS-B type code, and that inference succeeds often enough to make wake spacing the dominant category. The observation about the cascading-delay detector survives; it fires on roughly one snapshot in one.

The harness now imports the unmodified production modules and calls `runPredictiveRules`, the same entry point the operator surface uses, so detectors, projections and de-duplication are all exercised. The single deviation from the browser path is HTTP transport: `fetch` is shimmed to apply the rewrites declared in `vite.config.ts`, because the application fetches through relative dev-server proxy paths that do not resolve under Node.

Ingest coverage The corrected harness also records how many ADS-B contacts the live pipeline discards. `fetchLiveTrafficDetailed` reports the raw contact count alongside the classified flights, and the gap between them is large: of roughly 169 contacts inside the 40 NM radius, about 18 are classified as KJFK arrivals or departures, near 10 per cent. That is one window's reading and the low end of what has been measured here. The KJFK/KATL comparison later in this section puts KJFK at 17.6 per cent, and the two are not reconciled by arithmetic: coverage moves with the traffic mix in the radius at the time of sampling, so the honest statement is a range of roughly 10 to 18 per cent at this field, not a constant. Those coverage figures were measured before the heading fix of 2026-08-30, with a classifier that gave a record carrying no track a heading of due north and then judged it by geometry like any other. Such a record is declined now, which lowers the classified share on its own. The other half of the change does not move in one direction: preferring the true track alters the heading used by up to a local variation plus drift, and the classifier admits a contact whose heading falls within sixty degrees of the bearing to or from the field, so a record can cross that gate either way. The net was therefore unmeasured in size and not certain in sign, and a window running under the corrected ingest is what settled it. That has now been measured once, and it rose. A four-hour window at Atlanta closed on 2026-08-30 under the corrected ingest (`imc-2026-08-301304`, verified against its dump on 2026-09-01) classified 25.4 per cent of contacts, 3,376 of 13,270 across 158 snapshots, against 23.7 per cent in the twenty-four-hour window sampled at the same field under the old one (`tuned16`, 12,362

classified of 52,214 contacts). Hour-matched across the four UTC hours the two share, to remove the diurnal cycle the way every other comparison here does, the share rose in each of the four and by 1.7 points pooled. The direction is what the mechanism predicts rather than a surprise: the classifier compares a heading against a TRUE bearing to or from the field, so feeding it a magnetic heading imposed a systematic offset of about a local variation that pushed some contacts outside the sixty-degree gate, and removing that offset admits more than declining the records with no track at all loses. It is one field on one day, four matched hours, and the last of them holds four snapshots, so it is a first measurement and not a settled figure. Departure attribution moved too, and how much depends entirely on what is held fixed, which is worth showing rather than resolving quietly. Comparing the post-fix window against the twenty-four-hour window as sampled, the share of departure pairs where attribution reached both aircraft goes from 37.6 per cent to 92.4, which is a 55-point improvement and almost entirely an artifact. The earlier window covers three fields and the later one covers Atlanta alone, and Atlanta attributes far better than Dallas Fort Worth or O'Hare: at Atlanta by itself the earlier window is already 75.0 per cent. Hour-matching on top of that, over the four UTC hours the two share, leaves 85.7 per cent against 92.4, a rise of 6.7 points, and one of the four hours moves the other way while another rests on a single pair. So the honest reading is a modest improvement consistent with the same mechanism as the classified share, on evidence far weaker than the raw comparison suggests. Every figure in this paragraph was recomputed from the dumps on 2026-09-01 and reproduces: 406 of 1,079 departure pairs, 183 of 244 at Atlanta alone, 54 of 63 hour-matched against 122 of 132, with the shared hours numbering four as stated, the hour that moves the other way being 15:00 to 16:00 UTC at 90 against 88 per cent, and the hour resting on a single pair being 16:00 to 17:00. The recomputation is recorded because the same exercise found two figures elsewhere in this section that reproduce from nothing. It is recorded here because the raw comparison was the one the tooling printed first, and a 55-point claim would have survived any check in this repository: every number in it is correct, and the aggregation is what is wrong. An earlier revision of this sentence said the share could only fall, which is true of the declined records and not of the change as a whole. The paragraph carried "near 10 per cent" alone until this revision, leaving a reader to meet 17.6 per cent fifty-seven lines later with nothing saying they describe the same thing.

The cause is the classifier in `src/sim/live-adsb.ts`, which infers intent from geometry: a contact counts as an arrival if it is descending, close, and heading toward the field, and as a departure on the mirrored test. A contact matching neither is dropped and never reaches a detector. Inside 40 NM of KJFK that discards traffic belonging to LaGuardia, Newark and Teterboro along with high-altitude transits. This matters for the alert rates in this section, because the denominator is the classified count and not the traffic actually present. It also marks a structural difference from a deployed system, which would receive flight-plan-correlated tracks from the facility rather than inferring intent from position reports.

Three things about that classifier were found in the same pass, by listing the fields the feed actually returns rather than by reading this section. Its two predicates are named for a vertical behaviour neither tests: `isArrival` calls `alt > 0 && alt < 12000 && distNm < 30` descending, and `isDeparture` calls `alt > 0 && alt < 10000` climbing, so an aircraft level at 8,000 ft inside 25 NM pointing away from the field is climbing by this definition. The field that would let them test what they are named for, `baro_rate`, is present on 82 to 100 per cent of airborne records, 63 of 69, and read nowhere. And a helicopter transiting a corridor points neither at the field nor away from it, so geometry discards it, while the emitter category that identifies one, A7, is present on 96 to 100 per cent of records and is also unread. The coverage figures above are therefore measurements of a classifier weaker than its own identifiers claim, and the traffic class in the collision this thesis opens with is the class that classifier drops. A fourth field in the same records states how good each position is. Every record carries a navigation accuracy category, and when that was measured most sat at the two best values, under thirty metres, while about one airborne record in fifty reported half a nautical mile or worse and one reported the category meaning ten nautical miles or worse; a few positions are multilateration-derived rather than broadcast, which the feed flags separately. Neither field is read. The proximity doctrine treats every position as exact and its critical tier turns on 0.5 NM, so a false-positive count computed without reference to reported accuracy cannot separate a genuinely close pair from a badly reported one. The share is small, the effect on the published rates is bounded by it, and the field is there.

The 24-hour sample The 24-hour KJFK sample is complete. An earlier run was started and discarded: it had loaded the detector modules before `weather-shift`, suppression and the alert lifecycle were im-

plemented, so it was sampling a version of the system that no longer exists (partial data retained as `fp-kjfk-partial-precorrection.csv`). The sample ran against unmodified thresholds so that tuning follows the measurement rather than preceding it.

Table 1. 24-hour KJFK false-positive sample, untuned thresholds (03–04 August 2026, 19:41Z to 19:41Z; raw CSV not preserved; see Section 6.10 on what survives and what does not). This is the replacement table, not the three-minute one withdrawn above, which carried the same number before it was retracted and is referred to here only as the withdrawn Table 1.

Metric	Value
Snapshots	2,417 over 24.00 h (effective cadence ~36 s; upstream fetch latency stretched the 20 s target)
Aircraft per snapshot	18.7 mean, 0 minimum overnight, 58 peak
ADS-B contacts per snapshot	105.8, of which 17.6% classified as KJFK arrivals/departures
Total alerts	4,657
Alerts per snapshot	1.93 mean · 1 median · 7 p95 · 16 max
Snapshots with zero alerts	1,099 (45.5%)
Projected (look-ahead) share	989 (21.2%)
Suppressed	692 (14.9%); operator view 1.64 alerts/snapshot
Critical	1,655, 0.685/snapshot
Warning / advisory / info	1,553 / 1,268 / 181
wake-spacing	2,576 (55.3%)
proximity-conflict	1,322 (28.4%)
cascading-delay	536 (11.5%)
runway-conflict	223 (4.8%)
All other categories	0 (gate unreachable from ADS-B, fuel reachable but silent in this window; weather and surface saw a clear, dry 24 hours). Read at the time as two unreachable detectors, then as four; three are, gate, runway identity and runway surface, and Section 6.7 says why. Fuel was counted among them until 2026-09-01 and does not belong: its figure is synthesised so the threshold clears at ingest, and the smoothing layer rewrites the ETA afterwards, which is how nineteen criticals were once measured in a single window

Version note: the run loaded the ten-detector population, including `runway-surface` but predating `runway-identity` and `squawk-emergency`. This sentence pinned that to commit `73105897`, and that hash resolves to nothing in the published history: the branch it named was squashed on merge, which rewrites the hash and leaves the citation pointing at an object no reader can fetch. The detector population is the load-bearing fact and is stated directly here instead. Neither absent detector affects the numbers: live ingest never assigns runways to departures, and no emergency code appeared in the window.

Four findings, in the order they will drive tuning:

Per aircraft-hour. Every rate here is per snapshot, which is a unit of this prototype's polling rather than of anyone's workload. Across the 4,085 snapshots on disk at a sixty-second interval, 1,351 aircraft-hours of exposure, the detectors raise 5.74 new alerts per aircraft-hour, from 1.2 at Boston and 1.3 at Kennedy to 6.7 at Atlanta and 7.5 at Dallas Fort Worth. The distinction that makes the figure right is between the alerts standing at a snapshot, a stock averaging 2.92, and the alerts raised since the last one, a flow. Only the flow divides into exposure. Dividing the stock gives 8.82 and is the same units mistake recorded elsewhere in this paper.

1. **The critical rate is 0.685 per snapshot, roughly one critical every 52 seconds, around the clock, on ordinary traffic.** The median snapshot carries 1 alert and the daytime p95 carries 7. Section 2.6's alert-fatigue standard makes this the claim-limiting number; no operator keeps trusting a critical tier that fires by the minute.
2. **Wake-spacing is 55% of all volume.** Its trigger compares ETA differences that live ingest recomputes from distance and ground speed every poll, so the gap jitters across the `required − 1` critical boundary. Hysteresis on that boundary, or a smoothed ETA, is tuning target one. Which categories produce the criticals is not answerable from this CSV; the severity-by-category crosstab was added to the harness after this run started, and the KATL window will report it.
3. **The diurnal split is structural.** 45.5% of snapshots are silent, concentrated overnight; the earlier 8.7-hour daytime slice of this same window ran 3.38 alerts/snapshot against the 24-hour mean of 1.93. Tuning against the 24-hour mean alone would under-correct the period that matters.
4. **Suppression and projection behave as designed at scale.** 14.9% of alerts arrived pre-subsumed, cutting the operator view to 1.64/snapshot; 21.2% of alerts were look-ahead, none critical, per the demotion policy. The suppression half of that finding did not survive the corrections, and the reason is instructive rather than damaging: `applySuppression` triggers on a critical alert carrying a runway id, and three of its four declared pairs need that critical to be a runway-conflict; withdrawing the false "multiple arrivals on final" doctrine removed almost every one of them. The fourth pair, a critical wake-spacing suppressing a lower-tier cascading-delay on the same runway, is the one that still fires, and it is why the figures below are one rather than nought. This sentence said the trigger was a runway-conflict outright until the third window produced a suppression with no runway-conflict alert anywhere in the snapshot, which is a counterexample a smaller sample would not have contained. Measured under the corrected doctrine, 15 of 4,958 alerts arrive pre-subsumed in one 24-hour window, 1 of 8,501 in the next and 1 of 9,756 in a third closed on 2026-08-29, so the operator view and the raw count are now within one alert of each other. The layer is neither broken nor untested; two scenarios exercise it and its unit tests are real. Its live effect was removed as a side effect of a correction two doctrines away, which nothing in the test suite could report, because a test watches whether a feature works and not whether it runs. `apps/atc/docs/suppression-live-effect.md` records the measurement.

Table 2. 24-hour KATL sample, untuned thresholds, with the severity-by-category crosstab (05–06 August 2026, 05:56Z to 05:56Z; 819 snapshots, upstream latency stretched the cadence to ~105 s; raw CSV not preserved; see Section 6.10). This window measured the full eleven-detector population.

Metric	Value
Aircraft per snapshot	7.7 mean (0–35); 30.5 contacts, 25.3% classified
Alerts per snapshot	3.21 mean · 0 median · 15 p95 · 24 max
Silent snapshots	60.1%
Critical	786, 0.960/snapshot
Suppressed	11.8%; operator view 2.83/snapshot
Look-ahead share	39.4%, none critical
Volume by category	wake-spacing 64.8% · proximity 14.8% · cascade 8.5% · runway-conflict 6.0% · weather-shift 5.9%
Critical crosstab	**wake-spacing 79.1% (622) · runway-conflict 20.2% (159) · proximity 0.6% (5)**

The crosstab is what this window existed to produce, and it settles the tuning order.

1. **Wake-spacing produces four of every five critical alerts.** The detector compares ETA differences that live ingest recomputes from distance and ground speed on every poll, so the gap jitters across the `required − 1` critical boundary. Hysteresis on that boundary, or a smoothed ETA, is the first tuning change, and the KJFK and KATL windows are its before/after baselines.
2. **The runway-conflict criticals are dominated by a correctness bug, not a threshold.** `inferRunway` assigned arrivals by heading, and KATL's 08L, 08R, 09L and 09R all carry heading 092: legally separated parallel approaches got assigned to the same runway string and fired "multiple

arrivals on final" at critical. KATL, with five same-heading parallels, is the worst case in the registry, and the fingerprint is the disproportion inside its own window: runway-conflict is 6.0% of alert volume and 20.2% of criticals, 159 of 786, so a category producing one alert in sixteen produces one critical in five. The same comparison cannot be drawn against KJFK, because that window predates the severity-by-category crosstab and its critical split is unavailable; KJFK's runway-conflict share of volume, 4.8%, is the only comparable figure and it is close to KATL's. The interim fix was lateral-offset disambiguation between parallels; attribution itself has since moved to extended-centreline geometry (threshold coordinates for all 62 registry strips, nearest centreline within 0.6 NM of cross-track, a round-trip test pinning all 124 ends), and an aircraft near no centreline now receives no attribution rather than a heading-only guess. The live figures in this section predate that change and stand as measurements of the heading-only system.

3. **Proximity is nearly silent at the critical tier live** (five alerts in 24 hours). The tau-based rework anticipated here was subsequently built: the critical tier now requires convergence, which is what keeps this tier nearly silent while the warning tier carries volume.
4. **Part of that `weather-shift` volume was fabricated, and the cause was a substring match.** Live weather read present conditions by looking for two-letter codes anywhere in the raw observation, remarks included. `TSNO` is a routine remark on United States automated stations meaning thunderstorm information is *not available*, and it contains the letters that mean thunderstorm, so a clear day at a field whose storm sensor was out reported convective activity. The CSV behind the table above no longer exists, so its exact share cannot be recomputed; the surviving KATL window can be, and 120 of its 265 `weather-shift` alerts, 45.3% of that category and 5.40% of all 2,221 alerts in the window, are attributable to it. They were identified by elimination rather than by assumption: every one was raised while the field reported VFR, which excludes the instrument- conditions trigger, and none had a gust spread within 15 kt of the threshold, which excludes the other. **No critical alert is affected.** `weather-shift` emits no criticals by design, and the count of critical `weather-shift` alerts across every surviving window is zero, so the critical crosstab below and every claim resting on it stand as reported. What moves is the volume-by-category share: `weather-shift` is overstated at this field and the other categories correspondingly understated. The parser now reads the observation body rather than the remarks and matches whole present-weather groups.
5. **`weather-shift` fired on live data for the first time** (154 alerts, 5.9% of volume, none critical by design), confirming the standing-condition severity cap behaves on real weather.

The KJFK/KATL comparison also bounds the coverage claim: 17.6% classification in shared New York airspace against 25.3% at a single-airport field, with the earlier KORD spot check at 43%. Atlanta has since been measured a second time, at 23.7% over the 24-hour window closed 2026-08-29, 12,362 classified of 52,214 contacts across 835 snapshots; it is the only one of these figures whose window data still exists, so it is the only one a reader could check. Two measurements of one field 1.6 points apart is the same point that the Kennedy range makes: the classified share moves with the traffic mix in the radius at the time of sampling, and a single number is the wrong shape for it. Ingest coverage is airspace-dependent, and the KJFK figure is the worst case in the registry, not the typical one.

One figure set the agenda for the tuning that followed. The untuned population emitted 0.685 critical alerts per snapshot at KJFK and 0.960 at KATL, so ordinary traffic drew a critical on roughly every snapshot, against 2 in total across the nine snapshots of the withdrawn table. The withdrawal made that picture worse rather than better, which is the direction worth noticing: the harness that was wrong reported a system almost silent at the critical tier, and the corrected one reports a system that is not. Sustained critical alerting on routine operations is the alert-fatigue failure mode described in Section 2.6, and it is the finding most likely to determine whether the detector population is fit for any external use.

6.7.1 Measured effect of the corrections The corrections of Sections 6.7 and 6.8 were measured against a second live window at the same five airports. The obvious comparison, alerts per 20-second snapshot, is only valid when two windows cover the same hours of the day, and these do not: the second window ran 17.4 hours rather than 24 because both volunteer ADS-B feeds began refusing, and the hours it lost were US midday. Traffic came in 16 to 45 per cent below the baseline at four of the five airports, so a per-snapshot reading would credit the corrections for a reduction that thinner traffic produced. The comparison below

therefore restricts both windows to the hours the second one covers, and reports per thousand aircraft.

Airport	Baseline	Corrected	Change	Aircraft compared
KJFK	35.8	2.1	−94%	5,217 → 7,645
KATL	102.5	9.8	−90%	3,609 → 7,067
KORD	120.4	3.3	−97%	4,983 → 11,794
KDFW	105.3	1.4	−99%	5,937 → 10,283
KLAX	34.0	3.2	−91%	5,732 → 13,741

Table 3. Critical alerts per thousand aircraft, hour-matched, before and after the corrections

Critical alerting fell 90 to 99 per cent per thousand aircraft at every airport, against a corrected window carrying more traffic in the matched hours than the baseline, and the share of completely silent snapshots rose at four of the five. Each row of that table survives the test the rest of this work applies to smaller samples. Taking the counts back out of the published rates and denominators, the baseline and corrected proportions are 187 of 5,217 against 16 of 7,645 at Kennedy, 370 of 3,609 against 69 of 7,067 at Atlanta, 600 of 4,983 against 39 of 11,794 at O'Hare, 625 of 5,937 against 14 of 10,283 at Dallas Fort Worth and 195 of 5,732 against 44 of 13,741 at Los Angeles. Every pair of 95 per cent intervals is disjoint, at every field, by a wide margin. That check was not run when the table was first published, and it is run now because a percentage change carries no statement about whether two rates can be told apart: the same tool printed a 371 per cent rise elsewhere from one alert against six, where the intervals overlap across half their width. One limit on that test belongs with it. The interval assumes independent observations and these are not: an alert persists across consecutive snapshots, so the same event is counted more than once and the effective sample is smaller than the aircraft compared. The intervals are therefore narrower than the truth, which is safe in one direction only. An overlap would still be an overlap at proper width, so a verdict of indistinguishable holds; a disjoint verdict is the one that could be overturned by treating the autocorrelation properly. The separations in this table are two orders of magnitude wide and are not at risk from it. A marginal one would be, and the tool now says so where it prints them. The false runway-conflict criticals are gone entirely, which is the doctrine withdrawal of Section 6.7 rather than a threshold change. The comparison is produced by `apps/atc/scripts/hour-match-compare.py`.

The comparison above lost US midday, so it was repeated on a window covering all twenty-four hours at Atlanta, O'Hare and Dallas Fort Worth, 762, 763 and 758 snapshots respectively. Critical alerting per thousand aircraft fell 97 per cent at all three: 96.5 to 2.6 at Atlanta, 116.4 to 3.3 at O'Hare, 93.8 to 2.4 at Dallas Fort Worth. The share of completely silent snapshots rose from 16 to 45 per cent, 13 to 31 per cent and 21 to 39 per cent. Because every UTC hour is represented on both sides, no part of that reduction can be attributed to the daily traffic cycle, and the second window carried roughly twice the traffic of the first. Neither that window nor the baseline it is compared against is on the authoring checkout, which was checked rather than assumed on 2026-09-01: the fp dumps here cover `tuned16` and the instrument windows only, and `tuned16` holds 835, 834 and 834 snapshots across 23 hours rather than the 762, 763 and 758 quoted here. This result therefore stands on the general disclosure of the reproducibility section rather than on anything a reader can recompute, which is worth saying at the headline figure and not only where the limit is stated. The vortex-band figures reported earlier in this section come from the same window and settled at 267 candidate pairs.

Three results cut against this and belong beside it. Wake pairs per thousand aircraft fell 89 to 99 per cent in step with the alert count, so part of the reduction is narrower coverage rather than better precision: the detector pairs only aircraft the runway inference can attribute and that are established on final, which on live data is roughly 7 to 25 per cent of approach-phase arrivals. Proximity conflicts became the dominant category and did not improve under threshold work alone: a second window covering all twenty-four hours at Atlanta, O'Hare and Dallas Fort Worth put them up 3 per cent per aircraft at Atlanta, up 100 per cent at O'Hare and down 13 per cent at Dallas Fort Worth. What settled the category was identity, not thresholds. Re-measured on the first window collected under extended-centreline attribution (`scripts/prox-runway-split.py`), 77 per cent of proximity pairs have both aircraft attributed to a runway, and 71 per cent of all pairs are aircraft

on distinct parallel runways at a median cross-track separation of 0.87 NM, the runway spacing itself, with 8 criticals among 1,855 (a window this paper does not name, and one whose dump is not on the authoring checkout: no window here yields 1,855 pairs or a 77 per cent attribution rate, the closest being 161 pairs at 81 per cent, and `prox-runway-split.py` records only that the dump schema these figures need dates from `tuned11`, which constrains the window without identifying it): measurably, the dominant alert category was ordinary simultaneous parallel approaches. The wake pairing had a second route to the same fault, closed on 2026-09-01. Arrivals are split into lateral streams so that only aircraft on one extended centreline are paired in trail, with a 0.1 NM tolerance, and each aircraft was compared against the last one admitted rather than against the stream. That is single-linkage clustering: ten aircraft each 0.09 NM further off the centreline than the one before joined a single stream spanning 0.81 NM and were paired as though in trail, and since the closest US parallel centrelines sit 0.12 NM apart such a chain crosses several. Each aircraft is measured against its stream's first member now, which bounds the width by construction. Not observed in the corpus, where every stream has zero cross-track spread, so no measurement in this section moves; demonstrated synthetically and pinned both ways, since a rule that split every stream would satisfy the chaining test while destroying the detector. The detector now demotes a non-converging pair attributed to distinct parallels to *advisory*; the tau-based convergence test is untouched, and a test pins that a blunder between parallels still escalates to critical. The re-measurement of volume under this rule has since run, and the effect is a reclassification rather than a reduction: hour-matched and per thousand aircraft, proximity warnings fell 64 to 74% across the three fields while advisories rose 69 to 86% and total proximity volume moved by less than 8% at any of them. The pairs remain detected, listed and auditable; they have stopped asking for the operator's attention as conflicts. The share of completely silent snapshots moved *against* this argument at two of the three fields, because a demoted alert is still an alert, and reporting the tier change as a volume reduction would be the flattering reading rather than the true one. Whether an advisory tier of this size is worth displaying is a design question this thesis does not settle.

The residual violation rate among the pairs the vortex band admits was this work's largest open question, and it is now settled in two steps. First, a follow-up window under the corrected doctrine (the wake/radar-floor split of Section 5.4 and the CWT migration) recorded cross-track separation per pair. That window is not named here, which is a reproducibility gap found on 2026-09-01 and stated rather than papered over: the figures below cannot be checked against a dump without knowing which one to open. The only window on the authoring checkout answering that description holds 111 admitted pairs and 37 violations, 35 of them against the floor, under any of four natural readings of what counts as a violation. The numbers below are left as measured rather than replaced, because window data is not committed and assuming the paper counted the window that happens to be present would be an inference, not a correction. The next window closed under this doctrine should carry its tag in the sentence that reports it. As measured: 45 of the 46 remaining violations sit within 0.02 NM of a single localiser course, in-trail on one approach, and 45 of the 46 violate the 2.5 NM radar floor rather than any stated wake minimum; the corrected wake table itself was violated once in the window, and the share of admitted pairs judged legal rose from 11% to 63%. Second, the explanation is doctrinal rather than statistical. Under FAA JO 7110.65 §7-2-1 and AIM 4-4-14, once the trailing pilot accepts visual separation or a visual approach following traffic, the radar minima cease to constrain the pair and wake-avoidance responsibility transfers to the pilot; visual approaches dominate US operations in visual conditions, and the acceptance is a voice-channel event no surveillance feed carries. That explanation has a boundary the order draws and this paper had not stated. Paragraph 7-2-1 also says visual separation is not authorised when the lead aircraft is a super, so a violating pair behind an A388 or A225 cannot be explained this way at all, and would be evidence against the explanation rather than a case of it. That makes the claim falsifiable, which it was not before. The transfer of wake responsibility has a second boundary in the same place. AIM 4-4-14 makes acceptance an acknowledgment that the pilot takes on wake separation only in operations behind a heavy, or for a small aircraft behind a B757 or other large aircraft, so it does not transfer for every pair that accepts. Both boundaries can be checked against the surviving per-pair dumps, 286 pairs of which 59 violate, and the result is sharper than the claim it replaces. Thirty-eight of the forty violate the radar floor of 2.5 or 3 NM rather than a wake minimum, and visual separation voids the radar minima whatever the categories are. The remaining two violate a 5 NM wake minimum, and their leaders are CWT categories B and D, both heavy, which is exactly the case where the AIM transfers the responsibility. No pair in the set has a super leader. So every violation observed falls

inside the regime, by the clause that applies to it rather than by a general appeal, and the shape of a counter-example is now stated: a wake-minimum violation behind a leader that is not heavy, with a follower that is not small, would not be covered by this explanation, and neither would any violation behind a super. Of the same 286 pairs, 24 carry an unmapped leader category, 14 per cent, which is the rate reported above from a different measurement. The 3 NM figure is itself conditional, which is worth stating because the prototype's sampling radius sits on the condition. Paragraph 5-5-4 gives terminal separation as 3 miles when less than 40 miles from the antenna and 5 miles at 40 or more, with the same 3 miles for a FUSION target symbol and 5 when ISR shows in the data block. Live ingest queries a 40 NM radius around the field reference point, so every contact it sees is at or inside the distance where the floor changes, and a pair at the edge of the query sits exactly where doctrine moves to 5 miles. Nothing in the measurements turns on it, since the pairs these detectors form are established on approach and far inside that radius, but the floor is stated here as conditional rather than as a constant. Every admitted wake pair in both completed windows (189 and 124 pairs) formed under VFR METAR conditions, exactly the regime in which the minima being enforced may not have applied. The detector now caps a radar-floor violation at *warning* in visual conditions and states why, while a violation of a stated wake minimum keeps its severity in every condition; the deployed parallel is ATPA, which is suppressed when visual separation is applied. The contrast the hypothesis predicts, a lower violation rate under IMC, remains untested, and the attempt to test it is itself reportable. A watcher (`scripts/imc-watch.mjs`) samples METAR flight categories at the ten US registry airports and launches a window whenever any reports IFR or LIFR. It fired on 24 August 2026 when Boston went IFR; the condition cleared within roughly eighty minutes, yielding 55 instrument-condition snapshots out of 1,431 and no wake pairs at all. Pooled across every window collected under the corrected doctrine (`scripts/imc-pool.py`), the totals stood at that point at 55 instrument-condition snapshots and zero pairs, against 343 visual-condition pairs of which 36% violate. The prediction has nothing to test against yet, and the exposure arithmetic (`scripts/imc-pool.py`) turns that from a gap into a statement. Two measured factors bound how many instrument-condition pairs this sampling could produce: such conditions are a small share of snapshots, and they carry less traffic. Pair formation needs two aircraft on one final, so it scales roughly with the square of the aircraft count, and halving the traffic quarters the opportunity.

The watcher was then rewritten twice on what its first firing taught, and the second rewrite changed the exposure by an order of magnitude. It had refused to launch whenever any sampler was running, a guard that declined eleven instrument-condition events, one of them San Francisco reporting IFR while a window sharing none of its airports had two hours left to run. It now compares the airports reporting instrument conditions against those a running sampler actually covers and launches for the difference, in four-hour windows rather than twelve, because a long window blocks the next event for its whole duration and this contrast needs many short events rather than one long one. Within hours of the change it caught a morning marine layer at San Francisco: 174 instrument-condition snapshots carrying about 7.9 classified aircraft each, against the 0.8 of the two overnight windows before it.

Pooled across every window under the corrected doctrine, instrument-condition exposure stood at **532 snapshots at a mean of 5.0 aircraft** (the full collected total; `imc-pool.py` prints 347 on the authoring checkout, the difference being the windows whose dumps did not survive, and the same distinction applies to every pooled figure in this section), against 5,927 visual-condition snapshots at 22.5, which puts the expected number of instrument-condition wake pairs at **1.9** if the violation mechanism were identical in both conditions. The `tuned16` window, closed on 2026-08-29, adds **145 instrument-condition snapshots at a mean of 13.5**, against 2,221 visual at 25.0, and expects **2.1**. Exposure and expectation are given per window rather than pooled, because the model was run on each separately and a pooled exposure would imply a pooled run that was never made; the pair counts below are totals and are pooled.

Six have formed, and the movement across them is worth more than any one figure. An earlier revision of this section reported a single instrument-condition pair, legally separated, as pointing the way the hypothesis predicts, with an explicit caveat that one pair cannot be separated from sampling noise. Two further windows took it to three and reversed the direction of that reading. A third window closed 2026-08-29 added two more, one violating, and the Miami window of 2026-09-01 added a sixth, compliant. The instrument sample now holds six pairs of which two violate, 33% with a 95% interval of 10 to 70. That six is the running total across every window this project has collected, and it is not what `imc-pool.py` prints

on this checkout, which is three of one violating: the difference is the windows whose per-pair dumps did not survive, named in the reproducibility section. A reader running the script and reading this sentence would otherwise conclude one of them is wrong, against the visual-condition rate, 35% of 427 pairs in the earlier windows and 33% of 109 in the new one: still indistinguishable, since that interval contains both. The visual arm is quoted per window rather than pooled, because pooling one arm of a comparison and not the other makes a sample look larger than it is. Three of the six come from windows whose per-pair dumps survive and three from windows whose dumps did not survive a move between machines, so only the newer count is re-derivable here. The exposure model above expected 1.9 pairs in the earlier windows and three formed; it expects 2.1 in the new one and two formed. Both sit inside the sampling noise at counts this small, so it remains about right on how many pairs the sampling should produce while the rate it produced no longer points anywhere. An earlier revision put that expectation at 3.2 in this sentence while stating **1.9** a few lines above, and printing two values for one quantity is the defect corrected here rather than a choice between them. The figure this section's own exposure counts support is 1.9, checkable by multiplying snapshots by the square of the mean aircraft count in each condition; the window CSVs that would show which run produced 3.2 did not survive a machine move, so the derivable figure is the one kept. Moving from one pair to three took the reading from 0% to 33% and reversed the direction of the inference, which is what a sampling-noise caveat means when it is honoured rather than recited.

The first instrument-condition violation is the most interesting pair this work has sampled, because this thesis's own explanation does not cover it. A CWT category B leader (a heavy) was followed by a category F aircraft at 4.59 NM against a stated 5 NM wake minimum from TBL 5-5-2, under IFR conditions, and the same pair had been legal one snapshot earlier at 5.39 NM. Three features separate it from the residual population of Section 6.7: it violates a stated wake minimum rather than the radar floor. That distinction is recoverable from the pair dumps but not directly readable in them, which was checked on 2026-09-01 rather than assumed: the dump records the separation a pair needed without saying which rule set it, and while 2.5 NM is unambiguous because no wake minimum takes that value, 3 NM is both a possible wake minimum and the radar floor away from short final. Across all 286 recorded pairs, 33 require 3 NM and 22 of those have both categories resolved, so the file alone cannot say. Recomputing each from the aircraft types, which the dump does record, resolves it completely: **none** of the 33 comes from wake doctrine, 22 are the floor with both categories known and 11 are the floor because a category is missing. The same pass measures the reach of the category gap reported in Section 7.4 into data already collected: 11 of 286 pairs, one in fifteen, involve an aircraft the CWT table cannot categorise. Where 45 of the 46 residual violations sat and where the visual-separation argument applies; it was measured in the regime in which that argument stops applying, since accepted visual separation requires visual conditions; and it crossed its own requirement between consecutive observations, which is separation compression on final, the phenomenon European time-based-separation and optimised-runway-delivery work exists partly to manage.

One pair proves nothing. A 0.41 NM deficit exceeds ADS-B position error but not the compound uncertainty of two smoothed positions and an inferred runway assignment, and both aircraft must have been correctly attributed to the same final for the pair to mean anything. That reasoning was applied to one pair and not to the set, which was measured on 2026-09-01: of the 59 violations across the 286 recorded pairs, 11 sit within 0.05 NM of their requirement, 14 within 0.10 and 22 within 0.20, and the smallest deficit is 0.010 NM, about sixty feet, against a median of 0.310. So roughly one violation in five is within three hundred feet of compliance. Those counts are taken over every recorded pair by comparing the gap against the requirement directly. An earlier revision counted the same bands over the compliance-margin column, which gave 41 violations and 10, 12 and 18, because that column is not written for a pair outside the lateral admission band and eighteen violating pairs therefore carried no margin at all. That does not make those pairs compliant and does not make them violations; it makes the count sensitive to an error the model does not represent, and any allowance introduced later would move a quarter of them at 0.05 NM and nearly half at 0.20. It is recorded as a single well-formed observation the existing explanation does not reach, in the regime built to test that explanation; `apps/atc/docs/imc-first-violation.md` holds the detail. What would turn it into evidence is more exposure of exactly the kind the watcher now accumulates unattended. A second instrument-condition violation formed in `tuned16`, closed on 2026-08-29, and it is a different kind. A CRJ9 in CWT category G led a 737 MAX 9 at 2.36 NM against a 2.5 NM requirement on Atlanta's 08L, a deficit of 0.14 NM. That requirement is the radar floor on final rather than a wake minimum, and it is

the floor precisely because the MAX 9 is one of the types the 2021 CWT table omits, so the wake table had nothing to say about the pair. It is a floor violation under instrument conditions, where the visual-separation explanation does not apply, so it is a second observation that explanation does not reach, arrived at by a different route from the first. What has changed is the character of the gap: the prediction is no longer untestable at this sampling rate, since the instrument works and exposure is accumulating, and roughly thirty instrument-condition pairs would make the comparison meaningful. A window at Miami opened at 02:07Z on 2026-09-01 and closed 55 minutes later at 03:02Z, holding 56 snapshots and one admitted pair: the first instrument-condition pair this project has formed away from San Francisco. While it was filling this passage said every snapshot was in instrument conditions, which was true of the opening minutes and not of the window: it closed with 18 IFR snapshots against 36 marginal and 2 visual, the condition having lifted rather than held. It was excluded from the figures above while it ran, because the pooling script refuses a window whose newest snapshot is minutes old and a window still filling is not a measurement; on closing it entered the pool, which moved the pair count from 165 to 167 and the admitted count from 116 to 117, and left the violation count at 40 because both of its pairs are compliant. It also carries the wake-category gap reported in Section 7.4 into live data within a day of that gap being found: the trailing aircraft is a 737 MAX, which the CWT table does not carry, so its category column is empty and the requirement fell to the radar floor. That did not change the answer here, because the categorised aircraft of the same generation sits in the same category as the leader and the table is blank for that combination, so 2.5 NM applies either way; the point is that the empty column is visible in the recorded data rather than silently filled, so the ambiguity can be audited. About six times the pairs now held would be needed. One limitation of this comparison sits in the ingest rather than the sample. A METAR arriving without a flight category is recorded as visual conditions, `m.fltCat ?? "VFR"`, and nothing downstream distinguishes that from a measured VFR: the windows collected so far store one condition string per snapshot, so the defaulted share cannot be recovered from any of them afterwards. The sampler now writes a second column recording whether the category was observed, added on 2026-09-01, so a window collected from here on can separate the two. Every dump this analysis rests on predates that column, which was checked rather than assumed: the newest of them does not carry it. The direction of the bias is knowable even though its size is not. Every snapshot mis-filed this way inflates the visual arm and removes exposure from the instrument arm, which is the starved one. Absence is neither visual nor instrument conditions, so it belongs in neither bucket, the way marginal conditions already sit in neither; changing it needs the rate measured first, and measuring it needs a field the window schema does not yet carry. That was tested on 2026-09-01 rather than assumed, by trying to recover the defaulted share from the visibility and ceiling the schema does carry: a snapshot filed as visual while reporting under three miles or a ceiling under a thousand feet would be a defaulted one caught in the act. There are none, across 3,065 snapshots, and the attempt fails for a reason worth stating rather than for the reassuring one. The seeded weather block sets ten miles and twenty thousand feet, and `lowestCloudFt` returns twenty thousand for a sky with no cloud layers at all, so a seeded triple and a clear day are the same three numbers. Every snapshot carrying them is filed visual and no snapshot outside them is, which is 1,512 of 4,501: the three fields move together because they are defaulted together. Most of those are certainly real clear days, and the share that is not cannot be recovered, because auditing one fabricated field with two others that share its fabrication recovers nothing. The sampler now records which airport the weather block was observed for, empty while the seeded block is still in place, which the application has tracked since the carry-across repair and the sampler was dropping. That closes the seeded-against-observed ambiguity for windows from 2026-09-01 and does not close this one, a distinction worth stating because the first attempt at this paragraph ran the two together. A METAR arriving without a flight category still sets the observed marker, since an observation did arrive; what it lacks is the category, and nothing yet records that separately. That second field was then added on the same day: the ingest records whether the observation carried a category, and the sampler writes it beside the condition. Windows closed from 2026-09-01 can therefore bound the defaulted share directly, which is the measurement this paragraph described as impossible, with one qualification found by opening the first window collected after the change: the flag recording whether a category was present works, and the companion field naming the airport an observation was made for wrote blank on every row, because the application sets it in its store and the sampler builds its own state. That is fixed, and windows opened before the fix answer the category question and not the seeded-block one. The decision the measurement was blocking, whether an absent category belongs in neither bucket the way marginal conditions already do, can be made against a rate

rather than a guess. The 3,065 snapshots already collected still cannot answer it, and no later analysis will change that, because the distinction was discarded at write time rather than lost afterwards. The limitation is not that the field was impossible to record. It is that nobody asked what would distinguish a real reading from a default until after the data was collected, which is the cost of a plausible value standing in for an unknown one. The window closed on 2026-08-30 shows why this accumulates so slowly, and it is not only that instrument conditions are rare. The watcher launched on Atlanta going IFR and sampled 158 snapshots over 3.86 hours, of which 12 were IFR, 108 were marginal and 38 were visual: the condition degraded out of instrument minima within twenty minutes and spent most of the window in the margin. A window opened at Kennedy at 10:04Z on 2026-09-01 repeated it: 54 instrument snapshots over 53 minutes, then the marginal band, with no wake pair formed in the instrument spell at all. Two windows is not a pattern, but it is the second time the same shape has been measured, and it is the mechanism behind the scarcity rather than an accident of one afternoon: the watcher launches on a field entering instrument minima, and a field entering them is usually a field about to leave them. Marginal conditions sit in neither arm of the contrast by design, since visual separation may or may not be in use there, so two thirds of a window launched specifically to catch instrument conditions contributed nothing to either side. It produced no admitted wake pairs at all, leaving the instrument sample at five, and its contribution was exposure rather than pairs. That default had a second, sharper instance, found on 2026-09-01 and fixed. Visibility was parsed with a float conversion, which stops at the first character it cannot read, and METAR writes the values that matter as fractions. A half mile was read as one mile, three quarters as three, and `M1/4`, the notation for less than a quarter mile and the worst category reported, matched nothing and took the ten-mile default: forty times the actual value. The thresholds it feeds are visibility under a half for below-minima, under one for low instrument conditions and under three for instrument conditions, so the numerators land exactly on the boundaries they should cross, and each test failed by the margin the misreading created. The alert survived only because the flight category triggers it independently, which is the same field defaulted optimistically above: two defaults failing in one direction, each concealing the absence of the other. The parser now reads mixed numbers, bare fractions, the bound prefixes and the unit suffix, and a test asserts both the parsed values and that each crosses its own threshold. The same function held the fault twice more. Ceiling was taken from broken and overcast layers only, and an obscured sky forms one as well: the vertical-visibility report is what fog gives, an indefinite ceiling the FAA treats as the ceiling for minima. Being neither, an obscured sky at 100 ft fell through to the same 20,000 ft clear default, and a layer whose base was missing made the result NaN, which fails every ceiling comparison, so an unreadable ceiling raised nothing rather than raising doubt. Both are fixed and pinned. The sweep that found the visibility defect had been organised by mechanism, numeric conversion, and these two wear a different one, so the shared property was never parsing: each had a benign default and no branch reaching it for the worst input. One further consequence of that default was found on 2026-09-01, and it is about how the evidence is gathered rather than how it is read. The watcher that decides when to open an instrument-condition sampling window reads the same `fltCat` field and treats its absence as *unknown*, so it neither launches nor pretends, while the ingest reads the same absence as visual. Two consumers of one field, two answers, in one codebase, and the more careful of the two was arrived at independently rather than by reconciling them. The watcher also holds the visibility and ceiling a category is computed from, and whose parsers were corrected the same day, so it could derive what it declines for; that is recorded rather than changed, because widening the condition that opens a window alters how the instrument sample is collected. That default is one of a set. Every weather fallback in the ingest is optimistic: a missing flight category reads as visual, a missing visibility as ten statute miles, no opaque cloud layer as a twenty-thousand-foot ceiling, and a missing wind speed as calm. Absent weather therefore reads as the best weather, and the crosswind and weather-shift detectors stay silent exactly when the data that would trigger them did not arrive. For an advisory layer that is the safer direction to fail, because a fabricated alert costs more than a missed one in a tool a controller has to trust; it is reported here because the direction was chosen and never stated, and because none of these values can be told from a real reading in the window afterwards. That remains the strongest argument in this thesis for historical-archive access, because an archive permits choosing daytime instrument-conditions periods instead of waiting for them to arrive at a sampled field.

6.7.2 The measured cost of removing the departure blind spot The proximity filter excluded airborne departures until 2026-08-25 (Section 6.4). Removing that exclusion carried a prediction, recorded in `apps/atc/docs/departure-proximity-blindspot.md` before the confirming window ran: proximity volume

should rise, because a traffic class was being added to the highest-volume detector, and most arrival-versus-departure pairs would be non-converging and therefore outside the parallel-approach demotion. The record stated that if volume did not rise, something else was wrong.

Volume rose. Hour-matched, per thousand aircraft: proximity from 114.8 to 133.3 at KATL, 92.6 to 148.7 at KORD, and 79.2 to 151.1 at KDFW. Departures now account for 3,352 of 7,548 pairs, 44% of the population.

The prediction missed the part that matters most. Criticals rose as well, from 1.4 to 1.9 per thousand aircraft at KATL, 1.1 to 2.6 at KORD, and 0.9 to 3.9 at KDFW; warnings rose between 176% and 362%. A prediction that gets the direction right and misses a fourfold move in the critical tier was not specific enough, and that is recorded here rather than smoothed over.

The new criticals concentrate where doctrine is absent:

pair kind	pairs	critical	rate
arrival to arrival	4,196	77	1.8%
departure to arrival	1,949	62	3.2%
departure to departure	1,403	52	3.7%

Departure-involved pairs reach critical at roughly twice the arrival-to-arrival rate, and the largest single group in the whole population is two airborne departures, 1,403 pairs at a median 1.32 NM and 350 ft. This is structurally the same finding as the parallel-approach result earlier in this section, where 71% of arrival-to-arrival volume proved to be ordinary simultaneous parallel approaches. A second measurement reaches that conclusion without sharing its method. The parallel-approach figure comes from runway attribution, asking which strip each aircraft is on; classifying the 7,288 pairs held in the twelve settled window dumps on this checkout by pure geometry instead, on the along-track share of their separation, gives 19.6 per cent in trail, 11.7 per cent abreast and 68.7 per cent diagonal. Two routes with no common step agree that the envelope is dominated by aircraft that are not following one another, which is worth more than either alone. Successive departures are separated by doctrine too, by diverging courses or by time intervals depending on category and runway (JO 7110.65 §5-8), and this detector models neither, so it cannot distinguish procedural separation from conflict.

The same remedy is unavailable, for a structural reason rather than a want of effort. The parallel-approach demotion works by comparing the two aircraft's assigned runways, and at the time departures carried no runway attribution: live ingest assigned runways only to arrivals, which was the same limit that prevents the runway-identity doctrine firing on live data (`apps/atc/docs/phase-vocabulary-audit.md`). The departure population therefore cannot be triaged the way the arrival population was, because the discriminator is absent from the data.

The trade is nonetheless the right one. A detector that cannot see airborne departures is wrong in a way no volume figure excuses, and the demonstration is in Section 6.4: the Potomac reconstruction detected its own collision only because of how one aircraft's phase was encoded. The cost is now quantified rather than assumed, and the route named here was taken.

Departure attribution landed on 2026-08-26, built by running the arrival geometry backwards: an arrival sits on the approach side of a threshold and so has positive along-track distance, while a departure has crossed the one it left and has negative along-track distance, so the difference between attributing the two is the sign of one scalar and a cap, set at 8 NM, on how far past the threshold to keep looking before a turn on course has probably happened. A round-trip test places a departure past all 124 runway ends at three distances and requires its own label back; a companion test asserts that the approach and departure sides of one end do not claim each other, which also closed an arrival case nobody had checked, where an aircraft past its own threshold had landed and was still being attributed to that approach.

Building it exposed a units error that would have shipped a false critical, and it is a units error rather than a threshold one, which is why it is reported here. Attributing a departure still on the takeoff roll produced a critical wrong-runway alert at KBOS. Attribution compares the aircraft's

heading against the coordinate-derived TRUE course inside a 30-degree gate; the runway-identity doctrine compares the same heading against the registry's MAGNETIC heading inside a 20-degree gate; and KBOS's stored heading for 04L is 24 degrees from its coordinate-derived course, a discrepancy the generation-time validation had already flagged. An aircraft correctly lined up on the runway it was cleared for was therefore reported as rolling on the wrong one, at the severity a controller acts on immediately.

A second fault in the same doctrine, found on 2026-09-01 and smaller: it could fire correctly and still decline to say which runway the aircraft was on. A strip stores one course, its low end's, so a field operating the reciprocal way sits 180 degrees from the stored value. The assigned side of the comparison resolves the end before reading the course and was right; the search for the runway actually being used read the strip's stored course directly and was not, so at KSFO, where the 28s are the arrival runways, an aircraft lined up on 28R matched no runway and the alert said the heading did not match the assigned strip rather than naming 10L/28R. Half of any field's configurations are the reciprocal ones, so this was the ordinary case. Both ends are compared now; nothing about when the alert fires changed, only what it can say.

The same units error was present in a third place, found by sweeping every heading comparison in the detector file rather than by anything reporting it. The filter deciding which arrivals count as established on final, which is what the wake-spacing pairing draws from, compared the aircraft's true track against the registry's magnetic heading inside a 30-degree window. At KLAX the two differ by 14 degrees, so an aircraft established on final had 16 degrees of real tolerance rather than 30: one 20 degrees off the true final course was excluded at 34 against the gate, and is admitted at 20 once both sides are true. Excluded arrivals never enter the in-trail pairing, so this understates rather than invents. No scenario runway carries a true course, so all 93 of them keep comparing magnetic against magnetic through the fallback and no scenario measurement in this paper moves; the live windows were sampled before the correction.

A third instance, and the one that errs in both directions: the crosswind component takes the angle between a METAR wind direction, which is referenced to true north, and the registry's magnetic runway heading. At KLAX a 30 kt wind along the magnetic heading computed as no crosswind where the true geometry gives 7.2 kt, and a wind along the true course computed 7.3 where the answer is none, so it invents crosswind on runways that have none and hides it on runways that do, by up to a quarter of the wind speed against a 25 kt limit. All three were found by listing every heading comparison in the detector file and reading each one. The paper already reported the first as a units error rather than a threshold one; that framing was right and did not go far enough, because it was a class rather than an instance.

A fourth instance, and the only one a viewer could see: the radar canvas places aircraft by projecting latitude and longitude about the airport reference point, a true-north frame, and drew the runway beneath them from the registry's magnetic heading. An aircraft correctly established on the approach therefore appeared beside the runway it was tracking. At KLAX, KSFO and KJFK that is about 1.2 NM off the drawn centreline at a 5 NM final and over 2.3 NM at ten; at KORD and EGLL it is under a twentieth of a mile. Not all of that spread is magnetic variation, since at some fields the stored heading also contradicts the surveyed geometry, and the displacement is the difference between the two whatever its cause. This one was not in the sweep that found the other three, because the display compares no angles at all: it draws one thing in one frame and another in another, and there is no comparison to inspect.

Attribution is consequently scoped to *airborne* departures. That is the scope it was built for, since its purpose is airborne proximity triage and a ground departure gains nothing from a runway label, and it leaves the runway-identity doctrine unreachable on live data exactly as Section 6.1 and the phase-vocabulary audit describe, so the false critical cannot arise in production. The defect was pinned by a test rather than fixed for three days, on the grounds that carrying coordinate-derived true courses on the runway record is a change to a shared data structure and belonged with the next such change rather than bolted to an attribution fix. It is now fixed. `Runway` carries the true course of each strip's first-named end, live mode attaches it wherever surveyed thresholds exist, and the doctrine compares true against true on both sides. Runways without coordinates, which is every hand-written scenario, fall back to magnetic against magnetic, so all twenty-eight scenarios are unchanged. The pinning test was inverted rather than deleted: it asserts that nothing fires with true courses attached and that the false critical returns without them, which keeps the reason for the field visible if a later change stops preferring it. Boston's stored headings are still wrong and are deliberately left

alone, since correcting them moves live attribution and the cross-track axis and so waits for a gap between measurement windows. Two doctrines comparing degrees drawn from different reference systems is an error that survives indefinitely, because both sides look like angles. `apps/atc/docs/departure-fix-cost.md` and `apps/atc/docs/departure-attribution-identity.md` record the sequence and the containment.

6.7.3 What attribution bought, what it cost, and what the aggregate hid The window that measured departure attribution produced a mixed result and one methodological lesson worth more than either half of it.

It works, and it is measurable. Attribution reaches 28% of the departure slots appearing in proximity pairs, against nothing before, and departure-to-departure pairs eligible for the parallel-approach demotion went from zero to 389 of 1,311. Proximity criticals per thousand aircraft fell at every field:

field	before	after
KATL	1.93	1.20
KORD	2.61	1.72
KDFW	3.89	3.36

It cost coverage elsewhere, through a guard added in the same change. An arrival past its own threshold has landed, and it no longer holds the attribution of the approach it just flew. That guard is correct and it reduced arrival attribution: the share of arrival pairs eligible for the demotion fell from 68% to 60% overall, and from 63% to 47% at KDFW, the field with five parallel strips and therefore the most to lose. KDFW gained least from the change for exactly that reason. A correct guard and a loss of coverage are both true at once, and reporting only the first would be the flattering half.

And the aggregate would have supported the opposite conclusion. The aggregate critical rate at KDFW rose 90% across this comparison. Read alone, that says a change to the proximity detector made one field much worse. The rise is 100 crosswind criticals across 20 snapshots, wind 26 kt gusting 33 from 270° against a runway family on 184°: a category that produced nothing in the earlier window and that tracks the wind rather than any detector, while proximity criticals at that same field fell. `hour-match-compare.py` now reports criticals by the category that produced them and names a crosswind swing as weather-driven.

That is the third layer of aggregation in this work to conceal a real effect. Total alert volume hid movement between severity tiers, which the tier split exposed; the tier split hid movement between categories, which this split exposes; and the category figure still aggregates over pair kinds, which is where the next one came from, as the following section reports. In each case the concealing number was arithmetically correct and answered a question nobody had asked.

6.7.4 Following the prediction into pair kinds The previous section names pair kinds in print as the aggregate most likely to hide the next effect. Following through produced a sharper result than the prediction, and it changes what this chapter can claim about the residual.

Criticals by pair kind, before and after departure attribution:

pair kind	before	after
arrival to arrival	1.8%	1.5%
departure to arrival	3.2%	2.2%
departure to departure	3.7%	3.2%

Attribution improved every kind, and departure pairs remain the worst at roughly twice the arrival-to-arrival rate. Read at that level the conclusion is the one Section 6.7.2 already reached: the detector models no departure doctrine, so departure pairs alert more.

Splitting the departure pairs again, by whether the parallel-approach demotion could reach them at all, says something different:

attribution state	pairs	critical	rate	95% interval
both attributed	447	1	0.2%	0.04 to 1.3%
one attributed	27	0	0%	0 to 12.5%
neither attributed	837	41	4.9%	3.6 to 6.6%

Where both aircraft carry a runway label the critical rate is one pair in 447. Where neither does it is 4.9%, and 41 of the 42 remaining departure-pair criticals sit in that group. The two intervals do not overlap, so this window separates the groups rather than merely ordering them; `prox-runway-split.py` now prints that test alongside the rates, because at a few hundred pairs per bucket the ordering of two rates and the ability to tell them apart are different questions, and only the second supports the reading that follows. That split has since been measured a second time, on a window sampled for the purpose rather than found in data already collected. A 24-hour run at the same three fields, closed 2026-08-29 16:23Z, 835 snapshots and 6,352 proximity pairs:

attribution state	pairs	critical	rate	95% interval
both attributed	406	0	0.0%	0.0 to 0.9%
one attributed	35	0	0.0%	0.0 to 9.9%
neither attributed	638	17	2.7%	1.7 to 4.2%

The interval for the unattributable group is disjoint from the interval for the fully attributed one and sits above it, so the separation replicates in direction and in significance. The magnitude does not: 2.7% against 4.9% in the first window, and those two intervals overlap between 3.6 and 4.2, so the gap between the windows is not itself a result. It does not separate from the middle group, which holds 35 pairs and an interval running to 9.9 per cent that contains both the others: that group is too small to tell apart from anything, and saying the intervals are disjoint, which this sentence did until 2026-09-01, claimed a three-way separation the sample does not support. It is what two samples of a few hundred pairs each look like when they agree about a direction. The claim this section rests on is the separation and its sign, and that is what carried over; the size of the effect is the part that moved, and reporting the replication without saying so would overstate it. It is worth naming what kind of replication this is: the two windows were collected 9.75 hours apart and share no snapshot, so the second is out of sample with respect to the first. Two external reviews in September 2026 both listed out-of-sample evaluation as missing from this work, which it was not; the words a reader looks for were.

The timing matters as much as the numbers. Run against the partial window the same comparison overlapped at seven and a half hours and separated at nine and three-quarters, so a reading taken before the window closed would have been a different claim depending on the hour someone asked. The figure above is from the completed run and no interim value is quoted anywhere.

The demotion is not failing on departure traffic; it is not reaching it. That is a coverage problem in the data rather than a doctrine problem in the detector, and the distinction is not cosmetic, because the two have different repairs. A doctrine problem would need divergence and time-interval separation modelled inside the rule, a substantial addition to a detector that deliberately holds one concern. A coverage problem needs the runway label present more often, which is an ingest change and leaves every doctrine alone.

The same split bounds the approach. Attribution reaches 28% of departure slots because extended-centreline geometry can follow a climb-out only until the aircraft turns on course; past that turn the departure is near no centreline, and no refinement of the geometry recovers it. The 28% is one window's coverage and should be read as one: how much of a climb-out stays near the centreline depends on the runway configuration in use and on the departure procedures flown off it, both of which change with the wind. What generalises is the mechanism that bounds it, not the figure. Closing the remainder needs the departure's cleared route

rather than its position, which means flight-plan-correlated tracks from a facility rather than a public ADS-B feed. Section 6.7.1 argues for historical-archive access on the grounds that an archive permits choosing instrument-conditions periods instead of waiting for them to arrive; this measurement makes a second and more specific argument, for facility data, and states what having it would close: 41 of 42 criticals in one pair kind, at a coverage figure that this data source held to 28% in the window measured.

6.7.5 The runway records disagree with themselves, and so does ingest The five paragraphs below were written into 6.7.4 as they were found, leaving that section running pair kinds and then registry data quality under a heading about neither, with the ingest paragraph landed in the middle of the registry ones. They are about the records the attribution figures are computed from rather than about the split those figures came out of, so they have their own heading.

A separate limit sits underneath all of these numbers, in the runway records themselves. Each strip carries a magnetic `headingDeg` and, where surveyed thresholds exist, a coordinate-derived true course; their difference is the magnetic variation at that field, which changes by well under a degree across one airport and must therefore be the same for every strip on it. It is not. The spread across strips was 14.8 degrees at KSFO, 11.9 at KBOS, 10.5 at KORD, 10.0 at KDEN, 8.9 at KDFW and 7.0 at KATL when first measured, against 0.0 to 2.2 at KJFK, KLAX, KSEA, KMIA, EGLL and OMDB. KDEN reads 0.0 now: it is not one of the three fields the open measurement window samples, so its correction could land immediately, and five fields remain. Where the spread is ten degrees, one of the two numbers is wrong for at least one strip. Attribution gates on 30 degrees and the runway-identity doctrine on 20, so an inconsistency of that size spends a third to a half of the budget before an aircraft is off centreline at all, and KATL, KORD and KDFW are the three fields every sampling figure in this section comes from. The test that detects this uses no external source, only the record's agreement with itself, and it is new in this revision.

A second, sharper test says which of the two numbers is wrong for part of that, and needs no external survey either. Two strips with the same surveyed true course are physically parallel, and parallel strips at one field cannot have different magnetic headings, because there is one magnetic north for the field. Two families broke that rule. KATL's five parallels are all 90.0 degrees true; four of them read 92 and `10/28` reads 99. KDEN's four parallels are all 180.5 true and split 174 against 184, and that one is fixed: four of its six strips gave a variation of -6.5 degrees against the two 17s at +3.5, the wrong sign for a field where magnetic runs about eight degrees east of true, so the 17s were corrected to 174 and the field now agrees with itself exactly. In both cases the offending value is the runway designator rather than the heading, and both fields are ones where the designator has been shifted away from the magnetic bearing because there are more parallel strips than L, C and R can label. So `headingDeg` was taken from the strip's name instead of its survey at exactly the fields where the name stops tracking the bearing. That does not account for every field listed above. For KBOS the answer was already in the repository and unapplied: `runway-ends-data.ts` records that the registry's headings are wrong rather than the coordinates, derives 020 true for 04L/22R, notes Boston's variation of about 13 west giving a magnetic course of 033 against a published ILS course of 035, and observes that the nine-degree error against the registry's 44 had been absorbed by the 20-degree inference tolerance. The measurement was made and written down beside the number it contradicts.

The repair is structural rather than per-field. `headingDeg` is hand-entered while the threshold coordinates beside it are generated from NASR, and NASR publishes a true alignment per runway end together with a magnetic variation per airport, whose difference is the magnetic heading. Deriving it the way the coordinates are already derived removes the class rather than the two instances a test can pin, and that recommendation is recorded in `scripts/gen-runway-geometry.py` where the next regeneration will meet it.

A third check on the same records compares the surveyed thresholds against the stated `lengthFt`. Small differences are legitimate, because threshold-to-threshold is shorter than the paved length wherever a threshold is displaced, but five strips differ by 6 to 50 per cent and one of those is not a displacement. KORD 09R/27L states 7,500 ft against 11,231 ft between its own thresholds, and `runway-ends-data.ts` records that this strip's east end moved 3,589 ft when the coordinates were rebuilt from NASR, because of the 2021 extension. The coordinates were corrected for that extension and the length beside them was not. The consequence reaches further than an earlier revision of this paragraph said. It claimed `lengthFt` feeds no detector and appears in exactly one place, the runway-identity alert's message. It appears in two: `RadarMap.tsx` divides

it by 6,076 to obtain the drawn length of the strip, so KORD 09R/27L is drawn a third short on the radar display as well as named wrongly in an alert. The same function takes the strip's angle from `headingDeg`, so the heading defects above are drawn too, and KATL's `10/28` is rendered seven degrees off parallel from the four strips it is parallel to. No detector reads either field, so the alerting figures in this chapter stand; the picture a controller reads does not. A third display defect sits beside them and was found by asking which modules the suite names. The runway palette table held six entries and the lookup wrapped, while KORD carries eight strips, so 04L/22R and 10C/28C drew in one pink and 04R/22L and 10R/28L in one violet, at the field carrying the most runways in this registry. The colliding pairs are differently aligned rather than parallel, which is what makes it a display fault rather than a separation one, and it still leaves a controller two strips of one colour on a map whose purpose is telling strips apart. That one is corrected rather than pinned, since a colour carries no survey: the table has eight entries and a test ties its size to the widest field in the registry, so adding an airport with more parallels fails instead of silently wrapping. A fourth was in the motion rather than the drawing. The simulator advanced every arrival by `x + dx * Math.cos(0)` with the y coordinate untouched, and `Math.cos(0)` is one, so heading was ignored: aircraft moved due east at their own speed whatever they were pointing at, and one on heading 321 gained no northward component. Against a display that rotates headings by minus ninety, which is north-up, that drew aircraft sliding east while pointing west. Correcting it exposed the other half, that the seeded arrivals sit west of a field landing westbound, so honouring their headings sent them away; both were moved together. None of this reaches the figures in this chapter, because detectors read `headingDeg` and never call the tick function, which is the same division as the length and palette defects: the alerting is measured, and the picture was not. The length is still not corrected here, because this repository holds no source for a published runway length and 11,231 ft is a threshold separation rather than one.

A related fault sits upstream of all of it, in ingest rather than in the registry. An aircraft's heading was resolved as `rec.mag_heading ?? rec.track ?? 0` until 2026-08-30, while every comparison downstream expects a true ground track. `mag_heading` is magnetic where the comparison is true, and it is the heading rather than the track, so it differs by drift as well as by variation; the `?? 0` gives a record with neither field a heading of due north rather than none. Measured against adsb.lol on 2026-08-29 the field is present on a small share of records, absent at KATL and KORD, one in 62 at KJFK and two in 22 at KDFW, with one aircraft carrying both and reading track 180.0 against magnetic heading 174.4. So the attribution figures above are right for most aircraft and out by up to a local variation for that share. It was pinned by a test rather than corrected for the same reason as the registry headings, that changing it moves live attribution while a window is open, and it is now fixed: tuned16 closed on 2026-08-29 and nothing is sampling, so ingest reads the true ground track first and returns no heading where the record carries neither field. A contact that cannot be oriented is declined instead of being pointed north and compared against runway courses like a measured value. The pinning test was inverted rather than deleted, and has since been replaced. The inverted version read `live-adsb.ts` as text and asserted the expression it contains, which checks the wording of a fix rather than the fix, and would have passed against any rewrite that preserved the characters while changing the behaviour. Two behavioural tests stand in its place, driving a stubbed provider response through `fetchLiveTrafficDetailed` at Kennedy: a record carrying both fields must resolve to the track and not the magnetic heading, a record whose track is exactly zero must keep due north rather than fall through to the other field, and a record carrying neither must be counted as a contact and left out of the classified flights. The current state is pinned per airport rather than corrected, because changing the registry moves live attribution and cannot land while a measurement window is open. What the correction does to the attribution figures above is not yet known, and this paragraph is here so that a reader knows it is outstanding rather than settled. The same fallback shape survives nine lines below the one just fixed, and measuring it reversed which correction to make. Distance from the field is read as `rec.dst ?? 0`, where zero means at the field, so a record arriving without a distance passes every distance gate and `phaseFromAlt` reads it as on final below 2,000 ft. Sampled on 2026-08-31 across fourteen of the seventeen registry airports, which the word all claimed otherwise until 2026-09-01, the registry having held seventeen on that date as well, 1,181 records carried the field and none omitted it: the ingest queries adsb.lol's radius endpoint, which reports a distance per aircraft because that is what a radius query is for, so the branch is unreachable on this path rather than merely unlikely. That leaves two possible corrections, and the more attractive one is the worse one. Latitude and longitude are already required two lines above and `airports.ts` exports `distanceNm`, so

the distance can be computed instead of read, which removes the fallback and declines nothing. It should not be done. `distanceNm` is an equirectangular projection, exact at the reference point and drifting with range, and against the provider's own figure over those records the residual reached 0.154 NM at the 40 NM query radius. Four tenths of one per cent, and still enough to change the answer to one of the 4,032 gate comparisons in the 672-record half of the sample, because twenty-four of those contacts sat within 0.2 NM of one of the six distance gates the file applies. Swapping a fabrication that fires zero times for a projection error that fires once is a bad trade, so the correction was to let absence decline, and it is made. It waited for the watcher to be idle with every field in visual conditions and no collection running, on the same rule that held the heading fix, since changing what ingest classifies part-way through a window splits that window across two versions of the code. Ingest reads the distance the feed supplies and declines a record carrying none, and a test pins both halves: a record with no distance is counted as a contact and left unclassified, and a distance of exactly zero is kept, because an aircraft over the field is a reading rather than an absence.

A second self-consistency check on the same records needs no coordinates and therefore reaches every field, and getting it right took two attempts. A runway's identifier is its magnetic heading rounded to the nearest ten and divided by ten, so 04 means 035 through 044 magnetic; identifier and stored heading record one fact twice. Compared end by end, 34 of 124 runway ends across seven fields disagree, and most of those records are correct. An airport with more than three parallel strips cannot label them all with one number, because the suffixes are only L, C and R, so it uses the adjacent number for the rest: Dallas Fort Worth carries 17C, 17L, 17R, 18L and 18R on five parallels that all store heading 184, and Denver carries 16L, 16R, 17L and 17R on four at 174. An end-by-end comparison calls the 17s at Dallas and the 16s at Denver wrong when they are the convention working as intended. The sound comparison is per heading group, because the convention always leaves one strip on the rounded number and shifts the others off it, so a group where NO identifier matches cannot be the convention. Two groups fail that test: San Francisco's 10L and 10R, which store 119 against an identifier implying 100, and Seattle's 16L, 16C and 16R, which store 174 against 160. Both offsets are close to the local magnetic variation, which is the signature of a stored true course where a magnetic one is meant, and both are invisible to the threshold comparison above. Pinned rather than corrected, for the same reason as the rest: the right values need FAA NASR data this project does not hold.

6.8 Negative controls

A separate corpus of near-misses that resolved safely, typically documented in the FAA's Aviation Safety Reporting System (ASRS) database or in NTSB incident-only reports, is required as a negative-control corpus. The system must not produce *critical* alerts on these scenarios; an *advisory* or *warning* tier is acceptable.

A first negative-control scenario is included in the implementation as `negative-control-asrs`. It places two arrivals on parallel KSFO approaches (10L/28R and 10R/28L) at 1.5 NM horizontal and 600 ft vertical separation, inside the simplified warning threshold but well outside the critical threshold. The expected behaviour was originally a *warning* proximity-conflict alert and not a *critical* one; under the parallel-approach demotion introduced with centreline attribution it now draws an *advisory*, which still satisfies the control's requirement. Until 2026-08-29 both aircraft in it carried an unresolvable clearance: the scenario assigns the runways `10L/28R` and `10R/28L` while declaring its own by single end as 28L, 28R, 01L and 01R, so neither assignment matched anything. The behaviour is unchanged by the correction, because the demotion works from geometry rather than from the clearance, and that is what let it sit unnoticed: a control asserting silence was silent for a reason unconnected to the input it was testing. A test now requires every assigned runway and gate in the corpus to name something its scenario declares. The scenario can be loaded from the scenario picker in the application header and inspected by the operator.

The corpus now holds eleven scenarios. Five of the earlier additions encode shapes that recur in ASRS controller narratives: a go-around with spacing restored (briefly 1.6 NM / 800 ft, warning acceptable, critical a failure), sequenced crossing-runway departures, three arrivals at legal wake spacing, a VFR corridor with 1.9 NM lateral but 1,500 ft vertical separation, and a sustained crosswind component of ~22 kts against a 25 kt limit. Measured: all six produce zero critical alerts; four are completely silent; the go-around flags its pair at warning. The VFR-corridor case is the geometry the withdrawn Table 1 miscounted 218 times, so it

now stands as a regression test against that class of error. The silent four assert more than the sub-critical two: a population that murmurs on legal traffic drowns its own signal.

A corpus that asserts silence has to prove the doctrine ran, and for most of this one's life it did not. Measured on 2026-08-18, all six controls then in the corpus formed **zero wake pairs**: every scenario passed its zero-critical assertion without the wake detector, the component carrying the largest model correction in this work, ever reaching a pair to judge. The corpus could not distinguish "evaluated and found legal" from "never evaluated".

The cause was the lateral-stream pairing introduced during tuning. The staggered control's three arrivals sat at y = 0.5, 1.5 and 0.0, which is 1.5 NM of lateral spread against a 0.1 NM stream threshold, so they split into three separate streams and no pair could form. Its brief also still described the withdrawn time-based doctrine, four- and five-minute gaps against a three-minute requirement, months after wake separation became distance-based.

Rebuilding it surfaced a tension between two constraints that both have to hold before a wake pair exists: cross-track offset within the stream threshold, and the follower no more than 1,000 ft above the leader. On a 3 degree glideslope those two fight each other. Nine nautical miles of in-trail separation puts the follower roughly 2,900 ft higher, outside the vortex band, so the pair is skipped. Legal spacing and the vortex-band gate are therefore in tension on any descending profile, and the pairs the detector does see are biased toward aircraft close together along the approach. Measured against live traffic, the bias is real and large. Fewer than one in ten of the pairs the band admits are legally separated, against roughly nine in ten of the pairs it excludes, and the excluded pairs sit a median of about 4.5 NM apart in trail against about 2.5 NM for the admitted ones. The effect appears separately at each of the three airports sampled. The margin distribution of Section 6.7 therefore describes a population the band has already filtered toward tightness, which is a property of the instrument rather than of the traffic.

The rebuilt control represents aircraft level on an intermediate segment before glideslope intercept, the ordinary case, and forms two pairs at 9.2 and 10.2 NM against 5 and 3 NM requirements, both judged legal, alerts still zero. A seventh control was added for the boundary the corpus lacked: a medium 5.3 NM behind a heavy, 0.3 NM inside the requirement, which must form a pair and stay silent. Two tests now pin the pairs forming, so a change that stops the wake doctrine running fails there rather than reading as a clean corpus pass. An eighth control places two arrivals abreast on KSFO's 28L and 28R at 4,300 ft, 0.708 NM apart, the geometry of a normal simultaneous parallel approach; it must draw at most a warning, never a critical, which pins the proximity doctrine's tau test against the nuisance-alert class that ASRS CALLBACK reports on blunder-free parallel approaches.

Three further controls now encode specific NASA ASRS reports rather than synthetic shapes, closing part of the gap this section previously declared. Each carries the record's own geometry and a test pinning the expected tier. NASA ASRS ACN 2071367 (Wake Turbulence Encounters report set, update 24, 2024): an A319 on base to ILS 22L at Newark in IMC, put within 5 NM of a heavy 787, inside the CWT category-B-over-F minimum, resolved by discontinuing the approach; the encoded pair must form, draw a wake *warning*, and never a critical. ASRS CALLBACK Issue 461 (June 2018): a departure level at its assigned 2,000 ft with advised traffic crossing 1,000 ft overhead, all three pilots visual, TCAS raising a climb RA anyway; the crossing separation is normal operation sitting exactly on the proximity doctrine's vertical boundary (the predicate requires a gap strictly inside 1,000 ft), and the encoded state must be silent. NASA ASRS ACN 2069720: an EMB-175 descending behind a heavy 787 en route into Miami airspace, ATC wake caution issued, a 600 ft excursion, flight resumed normally; the approach minima of TBL 5-5-2 do not apply to an en-route pair, so the wake doctrine must form no pair. Where a record does not state a figure the encoding says so in place rather than inventing one; the CALLBACK report's aircraft types are unstated and encode as unknown, and the in-trail distance in ACN 2069720 is unstated and encoded wide of every envelope so the assertion cannot hinge on it. A fully representative corpus across multiple US TRACONs remains future work, but the assertion has changed kind: three controls now regression-test the detector population against events that actually occurred and resolved safely.

One property of these controls runs against the corpus and is worth stating here rather than in a footnote. Each carries a `sectorId` naming a real facility, and they are built on one default runway set, 28L, 28R, 01L

and 01R, which is San Francisco's. A control labelled "KBOS TWR" therefore flies runways Boston does not have, and the same holds for the ones labelled for Denver, Los Angeles, Kennedy and O'Hare. The label is shown on the operator surface, so a reviewer checking geometry against it finds the two disagree. One of the seven was a contradiction rather than a looseness and is corrected: the parallel approach control read "KORD TWR" while this section describes it as arrivals abreast on KSFO's 28L and 28R, and the geometry is San Francisco's, so the label moved to KSFO. The label was load-bearing, which the word does not suggest, because the committed BlueSky export derives position from it and had placed both aircraft at O'Hare while they flew approaches to San Francisco.

The remaining six are left as they are, deliberately. Making each control fly the airport it names is a design change to the corpus rather than a correction to a claim, and it would move geometry these controls were tuned against. What has changed is that the choice is enforced instead of merely recorded: the six are pinned in the suite, a seventh appearing fails, altering one of the six fails, and a further test fails if a pinned entry quietly becomes clean, so the list cannot rot into a comment. The nine reconstructions were checked in the same pass and use their own fields' runways. The exception is the American 11 blind spot, whose sector is ZNY, an en-route centre controlling no runways at all; nothing in that scenario reads a runway, since its purpose is to assert silence.

6.9 Baseline comparison

A monolithic-threshold baseline implementing the same doctrinal rules in a single function is included as `apps/atc/src/sim/rules-baseline.ts` and is the reference against which the orchestrated population is compared. The architectural claim of Section 3 is that the orchestrated population is more maintainable, not necessarily more accurate; the comparison is therefore primarily structural.

The comparison is run by `apps/atc/scripts/baseline-compare.mjs` over every scenario. One claim is confirmed and three are withdrawn.

Confirmed, with one qualification: the two implementations agree on category and severity across the incident corpus, on the doctrines the monolith implements. Three doctrines postdate the baseline freeze, weather-shift, runway-identity and emergency-squawk, and exist only in the orchestrated population, so it is a superset wherever the weather is adverse, and on Comair 5191 the monolith is blind outright: zero alerts against the population's one critical, because the wrong-runway doctrine has no block in the frozen function. They diverge only on the synthetic scenarios. Recounted on 2026-09-01, the `crisis` case yields seven alerts from the monolith and eleven from the orchestrated population, where this previously read eight and ten. Two of the four are `weather-shift`, one of the three doctrines named above as postdating the freeze. The other two are `gate-conflict`, and the reason given here was wrong: the thresholds are identical in both implementations, and both alerts come from a second gate rule that fires when more than one arrival is assigned to the same gate, for which the frozen function has no block at all. So the undeclared divergence is a fourth doctrine rather than a threshold, and the `gate-gridlock` case diverges by one alert for the same reason, which went unmentioned. Every divergence found is the population raising something the monolith cannot; none is a disagreement about a shared rule. The baseline's header comment anticipates this, describing its behaviour as close to but not bit-identical with the orchestrated population.

Confirmed: the baseline has no projection layer, and the de-duplication in `runPredictiveRules` is implemented as described.

Built rather than withdrawn: suppression. When this section was first verified against the code, no suppression logic existed; `runAllRules` concatenated detector output and sorted by severity. It is now implemented in `applySuppression`, and Appendix A.4 records both what it does and the two respects in which the original specification was wrong. Its measured effect on this corpus is narrow: one suppression, on the `runway-conflict` scenario, where a `critical` runway-occupancy alert subsumes a `warning` alert describing the same occupancy on the same runway.

Corrected: the Tenerife figures. The monolith produces one alert; the orchestrated population produces two, `runway-conflict` and `weather-shift`. The earlier claim of "four overlapping alerts against two" is still wrong on the monolith's count, and the direction of the comparison is the reverse of what was claimed: the

orchestrated population emits more, not fewer, because the monolith implements no weather doctrine. The categories cited for the orchestrated output are now the categories it emits.

Withdrawn: the maintainability conclusion, on the strength of the drill that was meant to support it. Revising the FAA wake matrix requires editing one declaration and one use site in the orchestrated population (`CWT_ON_APPROACH` in `src/sim/rules.ts`) and one declaration and one use site in the monolith (the mirrored `CWT_ON_APPROACH` in `src/sim/rules-baseline.ts`). The cost is identical, so this drill does not distinguish the two architectures. That much has held; the numbers around it had not. This sentence cited four line numbers until 2026-08-29 and all four had rotted, pointing at an import and a comment in one file and an import and a runway-name helper in the other, which is what line numbers do in a living file and why the symbol is named here instead. On size the comparison runs against the architectural argument. Counting non-blank, non-comment lines, the orchestrated population is 702 against the monolith's 290. The paper carried 360 against an "unchanged" 212 by an unrecorded method, and neither figure survives any obvious counting of the current files, so the method is stated here and the numbers can be re-derived. The monolith is smaller because it implements less: eight doctrines to the population's eleven, with no suppression and no alert lifecycle. A size comparison between implementations of different scope measures scope, not maintainability, which is a further reason the drill in this section needs redesigning.

A further finding bore on Section 3 rather than on this comparison, and has since been fixed. The pattern defined in Section 3.1 assigns one detector to one doctrine, but `wake-spacing` was emitted from inside `detectRunwayConflicts`, which at 76 lines was the largest of the six functions and carried two doctrines. It is now split: `detectWakeSpacing` and `detectRunwayConflicts` share a `groupByRunway` helper and carry one doctrine each. The population is eleven detectors, and the pattern holds across all of them.

What remains of the maintainability argument is the registration mechanism, and the redesigned drill has now been run. A new doctrine, runway-surface contamination, was added to both implementations in a single commit: `surfaceFriction` was already modelled on every runway and read by nothing, so the doctrine is real rather than contrived. Measured from the commit's own diff, the orchestrated population took 44 added lines (a 43-line self-contained detector plus one registration line) and the monolith took 39 (the same logic inlined). Line cost is near-equal, and the drill is reported that way.

The difference the drill did expose is isolation. The monolith block had to rename its `using` variable to avoid colliding with the crosswind block's scope and required care about loop-variable collision with the blocks above it; the orchestrated detector shares nothing with its neighbours. At one added doctrine that composition cost is small, and the claim it supports is correspondingly modest: the architecture buys isolation between doctrines, not fewer lines. Both implementations emit identically on the scenario that exercises the new doctrine (five alerts, same categories and severities on the crosswind-storm case, which gained wet pavement, rain without a wet surface having been a latent inconsistency the new doctrine made observable).

The drill also produced a finding about equivalence maintenance itself: every doctrine added from here doubles its implementation cost, because the monolith exists only to be compared against. That burden is the honest price of keeping Section 6.9 measurable, and it is why the corpus stops growing the baseline after this drill.

6.10 Reproducibility

The scenarios are deterministic, and determinism is now enforced rather than asserted: `src/sim/sim.test.ts` runs every scenario twice and compares the alert identifiers and severities, and separately checks that the detector pass does not mutate the state it is given. Each scenario loads with `speed: 0` so the output can be inspected without temporal evolution, through the picker in the application header and without writing code. Every scenario additionally exports to a BlueSky `.SCN` file via `scripts/export-bluesky-scn.mjs`, with a test pinning the committed exports against the scenario source, so the corpus geometry is written in the format the open ATM simulator the research community already uses reads (Hoekstra and Ellerbroek, 2016) rather than only in this project's harness. The file is what is established, not the replay: the 28 exports carry 77 aircraft and every latitude, longitude, heading, altitude and speed is inside its valid range, checked with a validator that fails on a planted out-of-range coordinate, but none has been loaded into BlueSky, and claiming they replay would be a claim about a tool this work has not run; the export

preserves traffic geometry, not detector semantics, and states its approximations in its header. The live-window figures are regenerated by named scripts rather than ad-hoc analysis: `fp-analysis.mjs` produces the windows and per-pair dumps, `hour-match-compare.py` the traffic-normalised before/after comparisons, and `wake-condition-split.py` the visual-separation split of Section 6.7 (the legal share, the floor-versus-wake-minimum division, the cross-track distribution and the METAR condition of every pair), so the section's central claim is one command to re-derive from a window, and the window is the part that is missing. There are no archived window files: an earlier revision of this sentence said there were.

Two claims previously made in this section were not true when written and are corrected here.

The deployment URL `atc.fbritoferreira.com` resolves and serves the application as of 2026-08-04: a proxied CNAME managed in Terraform (matching how the zone's other Pages hostnames are managed) plus a custom-domain binding on the `fbf-atc` Pages project. This hostname was cited here while it had no DNS record at all.

The application is open-source as of 2026-09-02, at `github.com/fbritoferreira/atc-decision-support` under Apache-2.0. It could not simply have its visibility flipped, because it lives in a private monorepo alongside unrelated material, so publishing it meant extracting it into a standalone repository, which `scripts/extract-standalone.sh` does and verifies by installing, testing, typechecking and building the result. This paragraph said the opposite until the day it stopped being true. A `CITATION.cff` is in place; archived on Zenodo with the concept DOI 10.5281/zenodo.22245225, which always resolves to the latest deposited version. A DOI is an archived artifact and an identifier that survives a rename, not peer review, and it is not claimed as one.

The two analyses in Sections 6.7 and 6.9 are re-runnable from the repository by the commands documented in `apps/atc/README.md`, and that is a weaker claim than an earlier revision made. It said the analyses were reproducible from the repository, which was not true and is worth stating plainly, because reproducibility is one of the properties this work argues for.

What is in the repository is every script and every detector. What is not is the measurement windows: `apps/atc/data/` is gitignored, the two CSVs this section previously cited as preserved under `atc-data/` were never committed and no longer exist, and the windows behind the published figures did not survive a move between machines. Running `imc-pool.py` pools the windows whose per-pair dumps survive, names the ones it cannot open, and names any dump in the data directory its window list does not mention, a check added after two closed windows sat unlisted for two days, rather than printing one number over a set it read only in part. It refuses to report zero for them, because printing zero against the 532 quoted in Section 6.7.1 would report nothing found where it had found nothing to read. Until 2026-08-30 it could open none of them and said so in those words.

So a reader can re-run the analyses and obtain the numbers for a window they sample themselves; they cannot re-derive the sampled numbers printed here. The claim checker behaves accordingly and says so out loud: six of its checks recompute figures from the window dumps, and on a checkout without them they skip with the reason printed rather than passing. Verified on 2026-09-01 by moving the data directory aside and running it, which prints eleven skip lines, reports no problems and leaves every other check running. A reader cloning this repository therefore gets a green result that means five things were not checked, and the way to know which is to read the output rather than the exit code. The exception is the scenario comparison of Section 6.9, which is computed rather than sampled and is re-derived by the suite on every run. The remedy is to commit the windows rather than to soften the claim: a complete 24-hour three-airport window is 1.2 MB of CSV, measured on `tuned16` rather than estimated, which is a normal thing to keep in a repository, and doing so would make the original sentence true. The cost of not doing so is sharper than a number that cannot be re-derived. Attempting on 2026-09-01 to identify the window behind the headline comparison from its own figures failed: the only comparable window here gives 1.7, 1.6 and 3.0 criticals per thousand aircraft against the 2.6, 3.3 and 2.4 reported, and inverts which field is quietest. That shows nothing about the reported figures, because they are hour-matched against a baseline and the recomputation is a raw whole-window rate, and the baseline needed to build the same quantity is absent. So a figure cannot be matched to a window at all, and a reader who suspects an error has no way to test it, which is also true of the author. The 1.4 MB this passage gave until 2026-09-01 counted the run log beside the dumps, and a

run log is not CSV. Until then this one is accurate and the other was not.

Window references in this paper named a closing date until 2026-09-01, and on one date that is not unique: three windows closed on 2026-08-29, `tuned16` at 16:09Z, `imc-2026-08-291331` at 15:14Z, and `imc-2026-08-290944`, whose pair dump holds a header and no rows. The phrase "the window closed on 2026-08-29" stood above for three separate measurements without saying which of the three produced each, and the sentences carry tags now because of what follows. Every window carries a tag and the tags are what the analysis scripts take as arguments, so the repair is to quote the tag rather than the date; all three have since been resolved by re-reading the data rather than by inference, and each named `tuned16` in the end. The first looked like it belonged to the two instrument windows that closed that day, which hold 138 instrument-condition snapshots between them against the 145 the sentence reports, close enough that substituting their tags and moving the figure to 138 would have read as a tidy correction. Atlanta in `tuned16` holds 136 IFR and 9 LIFR snapshots, which is 145 exactly, and no other window or combination reaches it; the tidy correction would have replaced a right citation with a wrong one. The second was settled by measuring the window's files. The third was settled by joining every violating pair to the weather recorded at its own timestamp, which finds exactly two under instrument conditions, both on Atlanta's 08L against the 2.5 NM floor, the second of them the CRJ9 and MAX 9 pair described above; the two instrument windows that closed that day contributed no violating pairs at all, one of them no pairs at all. That the ambiguity resolved in one direction every time is not evidence it was harmless. Three windows did close that day, the prose could not say which, and the resolution took measurement rather than reading. A later paragraph in this same section pools the surviving per-pair dumps and reports 286 pairs of which 59 violate; that figure was recomputed on 2026-09-01 and reproduces exactly, including the split of 38 against the radar floor and 2 against a 5 NM wake minimum behind heavy leaders, and the 23 unmapped leader categories at 14 per cent. The 46 above does not reproduce from any window or combination of windows still on the authoring checkout. Neither does the 63 per cent beside it: the share of admitted pairs judged legal is 66.7 per cent across every closed window, 78 of 117, and 66.7 per cent within `tuned16` alone, 74 of 111, whether pooled over pairs or averaged over snapshots. Both figures come from the same sentence, so it is the sentence rather than either number that belongs to a measurement this checkout cannot reconstruct. Where the two disagree, the 167-pair figure is the one to use: it is the later measurement, it states its own pool, and it is the one that can be checked. `imc-pool.py` prints it as one line on a checkout that holds the windows, giving recorded pairs and violations beside the admitted subset, because those two counts differ and were confused for each other while this paragraph was being checked.

Neither analysis requires an API key: traffic comes from the volunteer adsb.lol feed and weather from the NOAA Aviation Weather Center.

One class of claim is checked automatically. `scripts/verify-paper-claims.mjs` regenerates the countable figures from the code that produces them, the detector population by reading the dispatch calls in `rules.ts` and the scenario, incident and negative-control counts from the corpus itself, and fails the test suite when a document disagrees. It reads every document that makes a claim about this artifact, which is a larger set than it sounds: this thesis, the repository README, the `CITATION.cff` that feeds the Zenodo deposit of Section 8.4, the subtitles the site renders its research cards from, the other paper this site publishes, every engineering note under `apps/atc/docs`, and the long-form write-ups kept in a sibling checkout. The notes are read from the directory rather than named, so the set tracks the directory instead of falling behind it, which is the failure it had: the list grew around this work rather than around the question of which documents exist, and the documents with the most riding on them were the ones it did not read. Adding them found stale counts in every one. A count of documents is deliberately not given here, because that number moved four times in a single day and this sentence would be the next thing to go stale. Those last two are read when that repository sits beside this one and named in the output when it does not, so a checkout without it reports which documents went unread instead of returning a clean result over a smaller set. It covered only the first four until the same revision. The figures it guards are the ones that have gone stale in practice, and nothing it does reaches a measured number: no check in it can tell whether 4.9% was ever true, only whether every document that quotes the detector count quotes the same one the code dispatches.

One published table is exempt from all of the above, and it is now guaranteed. The scenario-by-scenario comparison of the monolith against the detector population is produced by code that is committed rather

than sampled from a window that is not, so it can be re-derived on any checkout. A test parses that table out of the paper, matches each row to a scenario by the label's own words rather than by a hand-kept alias list, runs all three configurations and compares; all ten scenario rows reproduce, as does the aggregate over the eleven negative controls including its claim that none of them yields a critical alert. The first version of that test resolved its path one directory short, found no paper, and skipped on every run, so it passed while reading nothing. It was caught by planting a wrong number in the table and watching the suite stay green, and it now reports a skip as a skip.

The count of tests those claims are quoted against is itself checked now. It is a hand-maintained constant, because a suite cannot count itself without recursing, and the comment above it asserted that CI verified it while nothing anywhere did. A separate script runs the suite under a machine-readable reporter and compares, in CI beside the suite.

A green suite is not a compiling one, and the distinction went unnoticed for an hour. The tests added for the smoothing path annotate their fixtures with a type they never import. Vitest transpiles rather than type-checks, so the whole suite passed while `tsc --noEmit` reported the name unbound in three places. Writing this paragraph then reproduced the defect it describes: quoting the test count as it stood made the claim checker read a stale current claim, the third time a finding has tripped the guard by quoting what it found. The count is not what the sentence is about, so it is gone. The workflow does run the typecheck, so a merge would have caught it, and the workflow has not executed a step since 28 August for a billing reason: the net that would have caught this is the one that is switched off, which is a more concrete cost of that block than a stale page count. The repair is a script that runs all seven commands the workflow runs, in its order, with a guard that fails when the workflow grows one the script lacks. That guard's first run found a defect in itself. It compared only the first word after the package name, so a `pnpm exec` passthrough reduced to a fragment no script would ever contain, and it would have failed permanently against correct input. It compares whole command lines now, with one exemption that carries its reason. A guard that always fires is read as noise, which is the failure it exists to prevent.

---

7. Discussion

7.1 What the diagnostic-shape check does and does not show

The shape check shows that the system's diagnostic output, when given the inputs the controllers had at the time of each event, corresponds to the conclusion the official investigation reached. It does not show that the system, if deployed, would have prevented any of the events. Each event has a chain of upstream contributing factors (clearance phraseology, procedural failures, sector workload) that are outside the detector population's scope. It also does not show what the system would emit on nominal traffic; the false-positive analysis required to characterise that is open work and is described in Section 6.7.

7.2 Determinism as a design constraint, not a limitation

A reader from the broader machine-learning community might object that the system uses no learned components. This is intentional. Aviation operates under regulatory regimes (FAA, EASA) that require deterministic behaviour, traceable decision logic, and reproducibility under audit. A learned reasoner in the alert loop would have to satisfy these requirements before being eligible for any operational role. The architecture proposed here is suitable for the present regulatory environment; a future extension with learned components is discussed in Section 8.

7.3 The role of look-ahead

The forward projection is the single most consequential design choice in the architecture. Without it, the system would only describe the present. With it, the system surfaces the situation the controller will face in 60, 120, and 180 seconds. Endsley's level 3 is operationalised directly. Several of the historical incidents (Avianca 052 on its first hold extension, the JFK near-miss on the second clearance) would have triggered look-ahead alerts ahead of the moment the controller was forced to act.

7.4 Limitations

1. **The ETA filter delays escalation, and the cost is quantified.** The smoothing described in Section 4 trades responsiveness for stability, and the price is stated here rather than only in the code. At the shipped weight of 0.4 on the newest observation, the filter's mean lag is 1.5 polls and a genuine, steadily closing gap crosses the critical boundary two polls later than it would unsmoothed, which is 40 seconds at the 20-second live cadence. That delay sits inside the margin the doctrine already carries, because the wake critical tier fires a full minute before the required separation is breached, so a delayed escalation still precedes the violation. It is a real cost all the same, and it scales with the weight: a test measures the two-poll figure directly against the constant, so tuning the filter fails the suite rather than silently moving this paragraph out of date.

2. **Reconstruction, not deployment.** The system has not been integrated into any operational ATC environment and is not represented as suitable for one. The diagnostic-shape check is by archive review, not by live trial.

3. **Data access bounds any live study of this shape, and the reason is not the one this section first gave.** Both public ADS-B sources stopped serving this client after roughly ten days of continuous five-airport sampling, and an earlier revision read that as goodwill exhausted by volume. Re-examined on 28 August 2026, the two refusals have different causes and only one is about volume. `adsb.lol` refuses a request whose `User-Agent` does not identify its operator, and says so in the body: "User-Agent too generic; include valid contact info." Node's own `fetch` is refused on the same grounds, so this is a standard a client must meet rather than a header it must merely set. The serverless runtime this proxy runs on sends no `User-Agent` at all unless the caller sets one, so every request from the deployment was refused while the identical URL from a laptop returned a full traffic picture; that was a defect in this client, since fixed, and it had been recorded here as an upstream withdrawal. The uncomfortable detail is that the project had already met the standard elsewhere: `scripts/fp-analysis.mjs` has sent a contact-bearing agent string since 9 August 2026, under a comment observing that an anonymous poller is indistinguishable from abuse. The edge proxy was written separately and never received the same treatment, so a correct fix sat in the repository for nineteen days while the deployment was refused for want of it, and nothing compared the two callers of the same API. With the header in place `adsb.lol` still returns 429 intermittently and for minutes at a time to the deployment, while the same request from a laptop succeeds at the same instant: serverless platforms egress through address ranges shared with every other tenant, so a project sending a few requests a minute is charged against an allowance it can neither observe nor control. `airplanes.live` refuses this client from every address and agent tried, with a body directing the operator to request access by email, which is an access policy rather than a throttle. Sampling was stopped rather than circumvented, which is why the post-correction window covers 17.4 hours rather than 24. The conclusion is unchanged and better founded: a study needing sustained multi-airport sampling requires a facility data agreement or a locally operated receiver. What changes is why. Public aggregation is not simply finite, it is granted per identity, and a client that cannot present a stable identity to it has no claim on it at all. Measured again on the evening of 2026-09-01, with that fix in place for days, the deployment is still refused and the cause has moved. `adsb.lol` answers 429 rather than 403, three times over forty seconds with an identical body, while the same URL carrying the same contact-bearing agent string from an ordinary address returns about 15,800 bytes of traffic. That locates the limit on the shared egress addresses the serverless runtime leaves through rather than on this project or its request shape, and it means no further header will clear it. The weather half of the same proxy answers 200 from the same deployment at the same moment, serving a live observation, so neither the proxy nor Cloudflare's addresses are refused in general: the limit is this one upstream's. The remedies are an authenticated endpoint, a different network path, or the policy conversation already pending with the second source, which refuses from any address.

4. **Detector population scope.** The eleven detectors cover the principal doctrines exercised by the corpus. Additional doctrines are not implemented: wake encounter on departure, RNAV path conformance, and controlled-flight-into-terrain proximity. Approach-path alignment with runway centrelines was listed here as unsupported until 2026-08-29 and is not: the runway record gained surveyed thresh-

old coordinates for all 62 strips in Section 6.9, and `alignedWithRunway` in `rules.ts` gates the wake detector on exactly that test. What the Air Canada 759 taxiway-overflight case needs is a different thing that remains absent, which is taxiway geometry; the model carries runways and no other surface, so an aircraft lined up on a taxiway parallel to its cleared runway is aligned with nothing the registry knows about.

5. **Linear projection.** The forward look-ahead holds heading and ground speed constant. Real arrival trajectories curve; the projection is therefore most accurate at short horizons and degrades beyond two minutes. A stronger objection used to apply, that the projected state was not always a physically possible state (queued aircraft climbing in place, arrivals held at zero altitude past touchdown, phase copied while everything it gates on moved); the corrections in Section 5.3 closed it, and what remains is the path model.

6. **Open data limitations.** ADS-B coverage is non-uniform; aviationweather.gov has occasional outages. The system degrades gracefully (last-known-good state) but does not yet have a formal data-quality alert.

7. **No formal latency budget.** The system is intentionally browser-based for inspection and reproducibility; an operational deployment would require a different runtime with a published latency budget.

8. **A fifth of catalogued aircraft types carry no wake category, and the requirement collapses to the radar floor for them.** Separation reads a table of 161 ICAO type codes; a pair whose leader or follower is absent from it yields no wake minimum and falls back to the radar floor. An earlier draft of this item said the detector could not fire for such a pair, which was wrong and was caught by running it rather than reasoning about it. The detector fires; the requirement and the tier collapse. At the same 1.0 NM spacing an `A388` leader is a critical against a 7 NM requirement and an `A38F` leader a warning against 2.5 NM, and the alert detail prints `CWT ?`, which is the one part of this that behaves well. 21 of the 77 types the display's older four-class table names are missing from it, and the list is not marginal: the entire MAX and neo families, which are among the most numerous narrowbodies at these fields, the Embraer regional jets, and the freighters. A freighter is the same airframe as its passenger sibling and usually heavier, so the sharpest statement of the gap is that `A388` requires 7 NM and `A38F` requires none. The volumes reported in Section 6 were measured over traffic in which those types could not produce a wake alert at all, which biases the wake share downward by an amount this repository cannot recover, since the windows record categories rather than type codes. It is pinned by tests rather than filled, because filling it means assigning CWT categories and those belong to TBL A-1 of the order rather than to inference from a sibling airframe.

9. **Stand wake rating is declared and not modelled.** `Gate.maxWake` exists on the type and is populated for every stand at every airport in the registry, and no detector reads it: the gate doctrine compares occupancy against arrival time and nothing else. Until 2026-08-29 the alert's suggested action read "reassign to a compatible wake-rated gate", which told a controller the system knew which stands take which aircraft. It now says another gate and states that the rating is not modelled. `Flight.wake` is unread in the same way, because the wake detector resolves the category from the aircraft type through the CWT table rather than from the declared field.

---

8. Pathway to US Adoption

This section addresses how the architecture proposed here might be brought into operational use within the US National Airspace System (NAS) given current procurement, certification, and research-funding mechanisms. It does not represent a current programme of work; it identifies the steps a US adoption pathway would entail and the federal stakeholders relevant to each.

8.1 Alignment with named US priorities

Three named US priorities provide the relevant policy context:

- **NTSB Most Wanted List 2025–26** identifies *Implement Comprehensive Runway Safety* and *Improve Surface and Approach Safety* as continuing priorities. The runway-conflict, gate-conflict, and cascading-delay detectors directly serve these priorities. The proximity-conflict detector addresses the *Reduce Mid-Air Collision Risk* item added in response to the Potomac incident.
- **FAA NextGen Implementation Plan** identifies *Surface Operations and Data Sharing* and *Trajectory-Based Operations* as continuing investment areas. The orchestration shape proposed here is complementary: NextGen brings improved surveillance and trajectory data, this architecture surfaces conflict diagnoses against that data.
- **FAA Aviation Safety Action Plan (2025)**, issued in the aftermath of the Potomac collision, includes a specific commitment to review helicopter-fixed-wing interaction corridors at congested fields. The proximity-conflict detector with helicopter-corridor reasoning, as exercised in Section 6.4, is directly responsive.

8.2 Funding mechanisms

The relevant federal funding mechanisms are:

- **Small Business Innovation Research (SBIR), DOT/FAA topics**. The FAA publishes SBIR solicitations under DOT-wide and FAA-specific topic areas; relevant topic codes in recent cycles have included surface safety, controller decision support, and AI/ML for ATM. Phase I awards are typically $150,000 over six months; Phase II awards extend to $1,000,000 over two years. SBIR is an obvious first step for the present work because it is small-business-friendly, does not require prior contractor history, and aligns the architecture with a specific topic statement.
- **FAA Broad Agency Announcement (BAA)**. The FAA periodically issues BAAs for research with no fixed topic. BAAs are best suited to mature architectures with a deployment partner.
- **NASA Aeronautics Research Mission Directorate (ARMD)**, *Airspace Operations and Safety Program*. NASA Ames Research Center has a long history of ATC decision-support research (CTAS, SARDA, surface operations); ARMD partnerships with industry are mediated by Space Act Agreements.
- **FAA William J. Hughes Technical Center**. The Tech Center operates the laboratory environments where new ATC decision-support concepts are integrated and tested before any operational consideration. A Cooperative Research and Development Agreement (CRADA) with the Tech Center would be a natural mid-stage step.
- **MITRE Center for Advanced Aviation System Development (CAASD)**. MITRE CAASD is the FFRDC that supports FAA ATM modernisation. Engagement with CAASD is typically through FAA-directed task orders rather than direct contracting.

8.3 Certification posture

The proposed architecture is a *decision-support* system, not an active separation-assurance system. This places it outside the certification regime that governs onboard collision-avoidance systems (TCAS) but within the regime governing ATC automation tools. Relevant standards are:

- **DO-278A**, Software Integrity Assurance Considerations for Communication, Navigation, Surveillance, and Air Traffic Management Systems. Applicable to ground-based ATM software. Equivalent assurance levels for ground software are SWAL-1 through SWAL-6; a decision-support advisory of the kind described here would be assessed at SWAL-3 or SWAL-4.
- **NAS Cybersecurity and Privacy Program (NAS-CSEPP)**, the FAA programme that governs cybersecurity for ground-based NAS systems.
- **FAA Order 1370.121**, Information Systems Security, the operative FAA IT-security order for ground-based systems.
- **NIST SP 800-53**, Security and Privacy Controls for Information Systems and Organizations, the federal baseline that NAS-CSEPP and FAA Order 1370.121 inherit from.

Note: DO-326A and DO-356A are airworthiness-cybersecurity standards applicable to airborne systems and are *not* the operative regime for ground-based ATM software. They are mentioned here only to clarify the boundary; this work falls under the ground-based regime above.

The properties stressed in Sections 4.1–4.5 (determinism, explainability, severity tiering, observability, audit trail) are the properties DO-278A would require evidence of in a SWAL-3 or SWAL-4 assessment. The architecture is designed with these properties from the start, which lowers (but does not remove) the cost of a certification engagement.

8.4 Concrete next steps

The concrete short-term steps that move the work from proof of concept toward a US adoption pathway are:

1. **Publish the source code** under an OSI-approved licence (Apache 2.0 or MIT) with a citable DOI via Zenodo, so federal contracting officers can evaluate it without procurement friction.
2. **Submit an arXiv preprint** of this thesis; the measurement work in Sections 6.7–6.9 that this step waited on is closed as of this revision.
3. **Submit a derived paper to the Digital Avionics Systems Conference (DASC) or the ATRD Symposium** for peer review and US ATM-research community visibility. Derived, because this document cannot be submitted as it stands: DASC caps a full paper at ten pages including figures, references and appendices, the ex-ICRAT symposium that became ATRD caps at ten, SESAR Innovation Days at eight, and the AIAA Journal of Air Transportation's regular-article band is ten to twelve thousand words against this thesis's twenty thousand across forty pages. Every viable venue wants roughly a quarter of the current length, so the submission is a separate paper written from this one rather than a condensed version of it, choosing which findings carry the argument and citing the long version for the rest. This document stays the reference artifact.
4. **Engage MITRE CAASD informally** through the public ATM research community before any FAA contracting conversation.
5. **Identify a current FAA SBIR topic** under the DOT solicitation and submit a Phase I proposal if a topic matches; otherwise wait for the next cycle.
6. **Letter of interest from a US aviation-safety research lab** (NASA Ames, FAA Tech Center, MITRE CAASD, or a Tier 1 university programme such as Georgia Tech ASDL, Purdue PEGASAS, or MIT Lincoln Laboratory).

These steps establish the architecture's US-priority alignment with verifiable third-party signal, which is the gap identified in the two completed rounds of external-persona review.

---

9. Future Work

1. **Wake-encounter on departure.** Add a detector covering wake encounters during departure climb, complementing the existing arrival-side wake-spacing detector.
2. **Trajectory-based projection.** Replace the constant-heading model with a published-flight-plan and STAR-aware trajectory projector.
3. **Cross-airport awareness.** Treat the New York TRACON (KJFK + KLGA + KEWR), the Bay Area (KSFO + KOAK + KSJC), and similar groupings as single decision domains, surfacing inter-airport conflicts.
4. **Learned anomaly detector under a formal verification envelope.** A learned component that flags anomalies inside the orchestrator's input stream, with a verifiable upper bound on false-negative rate, would be a candidate for inclusion within the existing regulatory constraints.
5. **Audit-format export.** Formalise the alert + reason + action + override-history trace into a compliance-grade audit format suitable for FAA or NTSB review.
6. **Replay and counterfactual analysis.** Record live-mode sessions and replay them offline with rule variations to support post-incident review and detector iteration.

---

10. Conclusion

This thesis has proposed a multi-detector decision-support architecture for air traffic control and exercised it against nine publicly-documented historical incidents, eight of them detectable by construction. The Specialist Detector pattern organises operational risk detection as a population of independent rule specialists coordinated by a thin orchestrator over a shared blackboard, with explicit reasoning and recommended actions attached to every alert. The implementation is licensed Apache 2.0, public since 2026-09-02, and operates on live ADS-B and METAR data for seventeen airports, ten of them in the United States. A diagnostic-shape check against the historical-incident corpus shows that the system's output corresponds to the conclusion the responsible investigation bureau reached in every case. The false-positive characterisation on live traffic, the negative-control corpus and the baseline comparison, listed as open work in earlier revisions, are now measured and reported in Sections 6.7 to 6.9, every round of correction measured against a fresh live window rather than argued; what they do not amount to is a strict-sense validation, for the reasons Section 6.1 states, and the sharpest of their findings is a boundary rather than a score: below the radar minima in visual conditions, surveillance data alone cannot say whether a pair is in violation or under legally delegated visual separation. The architecture is offered as a starting point for production decision-support work in US sectors of national importance, with a concrete pathway to US adoption described in Section 8.

References

Agenzia Nazionale per la Sicurezza del Volo (ANSV) (2004). Final Report on the accident at Milano Linate, 8 October 2001.

Bainbridge, L. (1983). Ironies of Automation. *Automatica*, 19(6), 775–779.

Cammarata, S., McArthur, D., and Steeb, R. (1983). Strategies of Cooperation in Distributed Problem Solving. *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI-83)*, 767–770.

Cummings, M. L. (2017). Operator Interaction with Centralized Versus Decentralized Unmanned Vehicle Architectures. *Journal of Aerospace Information Systems*, 14(7), 376–388.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32–64.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. (1980). The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Computing Surveys*, 12(2), 213–253.

EUROCONTROL (2017). EUROCONTROL Guidelines for Short Term Conflict Alert (Guidelines 159, Parts I–III).

Federal Aviation Administration (2024). Aeronautical Information Manual, §4-4-14, Visual Separation.

Federal Aviation Administration (2024). Order JO 7110.65, Air Traffic Control. US Department of Transportation. (§5-5-4 wake turbulence application and TBL 5-5-2; §7-2-1 visual separation.)

Federal Aviation Administration (2024). Title 14 of the Code of Federal Regulations, Part 91.167, Fuel requirements for flight in IFR conditions.

Friedman-Berg, F., Allendoerfer, K., and Pai, S. (2008). Nuisance Alerts in Operational ATC Environments: Classification and Frequencies. FAA William J. Hughes Technical Center.

Government Accountability Office (2024). GAO-24-105254: FAA Actions Needed to Better Support Air Traffic Controller Efforts (Terminal Sequencing and Spacing deferral).

Hoekstra, J. M., and Ellerbroek, J. (2016). BlueSky ATC Simulator Project: an Open Data and Open Source Approach. *Proceedings of the 7th International Conference on Research in Air Transportation (ICRAT).*

International Civil Aviation Organization (2016). Doc 4444, Procedures for Air Navigation Services, Air Traffic Management (16th ed.).

Kent, T., De Ath, G., Everson, R., et al. (2026). A Future Capabilities Agent for Tactical Air Traffic Control. arXiv:2601.04285.

Kuchar, J. K., and Yang, L. C. (2000). A Review of Conflict Detection and Resolution Modeling Methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4), 179–189.

Ljungberg, M., and Lucas, A. (1992). The OASIS Air Traffic Management System. *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence (PRICAI '92).*

Lockheed Martin (2011). Lockheed Martin-Developed Automated Tool Points Way to Safe Separation of Aircraft on Final Approach (Automated Terminal Proximity Alert, Minneapolis TRACON). Press release, 7 September 2011.

NASA Aviation Safety Reporting System. *CALLBACK*, Issue 461, June 2018. Reports of TCAS resolution advisories with traffic in sight and under visual separation.

NASA Aviation Safety Reporting System. *CALLBACK*, Issue 524, September 2023. Reports of wake encounter and TCAS resolution advisory during visual approaches to parallel runways.

NASA Aviation Safety Reporting System. ASRS Database Report Set: Wake Turbulence Encounters, Update 24, July 9, 2024. (ACN 2071367, ACN 2069720.)

NASA Langley Research Center. *An Analysis of the Role of ATC in the AILS Concept.* NASA/TM-2000-210091. NTRS 20000052924.

NASA Langley Research Center. *Description of the AILS Alerting Algorithm.* NTRS 20000040456.

NASA Langley Research Center. *Flight Test Evaluation of the Airborne Information for Lateral Spacing (AILS) Concept.* NASA/TM-2002-211639. NTRS 20020043800.

NASA Langley Research Center. *Simulator Evaluation of Airborne Information for Lateral Spacing (AILS) Concept.* NTRS 20010028813.

National Transportation Safety Board (1991). Aircraft Accident Report AAR-91/04: Avianca, the Airline of Colombia, Boeing 707-321B, HK 2016, fuel exhaustion, Cove Neck, New York, January 25, 1990.

National Transportation Safety Board (1991). Aircraft Accident Report AAR-91/08: Runway collision of USAir Flight 1493, Boeing 737, and SkyWest Flight 5569, Fairchild Metroliner, Los Angeles, California, February 1, 1991.

National Transportation Safety Board (2026). Final Report DCA25MA108: Midair Collision between PSA Airlines/American Eagle Flight 5342 and US Army Black Hawk H-60, Reagan Washington National Airport, January 29, 2025.

Pang, Y., Zhao, Y., Yan, H., and Liu, Y. (2021). Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach. *Transportation Research Part C*, 130, 103326.

Parasuraman, R., and Riley, V. (1997). Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors*, 39(2), 230–253.

Power, D. J. (2002). *Decision Support Systems: Concepts and Resources for Managers.* Quorum Books.

RTCA (2008). DO-185B, Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II).

Spanish Comisión de Investigación de Accidentes e Incidentes de Aviación Civil (CIAIAC) (1978). Investigation of the collision between Pan American World Airways Boeing 747 and KLM Royal Dutch Airlines Boeing 747 at Los Rodeos Airport, Tenerife, 27 March 1977.

Stone, P., and Veloso, M. (2000). Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3), 345–383.

Sun, J., Ellerbroek, J., and Hoekstra, J. M. (2020). OpenSky report 2020: Analysing in-flight emergency situations. *Proceedings of the 39th Digital Avionics Systems Conference (DASC).*

Wickens, C. D. (1992). *Engineering Psychology and Human Performance* (2nd ed.). HarperCollins.

Wickens, C. D., Rice, S., Keller, D., Hutchins, S., Hughes, J., and Clayton, K. (2009). False Alerts in Air Traffic Control Conflict Alerting System: Is There a "Cry Wolf" Effect? *Human Factors*, 51(4), 446–462.

Wollkind, S., Valasek, J., and Ioerger, T. R. (2004). Automated Conflict Resolution for Air Traffic Management Using Cooperative Multiagent Negotiation. *AIAA Guidance, Navigation, and Control Conference*.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2nd ed.). Wiley.

Appendix A, Implementation Details

A.1 Detector contract

```
type Alert = {
  id: string;
  severity: "critical" | "warning" | "advisory" | "info";
  // Every category below has an emitter in rules.ts. A previous revision also
  // declared "missed-handoff", which was removed: SimState models a single
  // sector, so there is no inter-sector handoff to miss.
  category:
    | "runway-conflict"
    | "wake-spacing"
    | "gate-conflict"
    | "fuel-hold"
    | "crosswind"
    | "weather-shift"
    | "cascading-delay"
    | "proximity-conflict"
    | "runway-surface"
    | "runway-identity"
    | "squawk-emergency";
  title: string;
  detail: string;
  flightIds: string[];
  reason: string;
  suggestedAction: string;
  createdAtTick: number;
  lookaheadMin?: number;
};

type Detector = (state: SimState) => Alert[];
```

A.2 Orchestrator

```
const DEMOTE: Record<Alert["severity"], Alert["severity"]> = {
  critical: "warning",
  warning: "advisory",
  advisory: "info",
  info: "info",
};

const demoteForHorizon = (severity: Alert["severity"],
  lookaheadMin: number,): Alert["severity"] => {
  const once = DEMOTE[severity];
```

```
  return lookaheadMin >= 3 ? DEMOTE[once] : once;
};

export const runPredictiveRules = (state: SimState): Alert[] => {
  const present = runAllRules(state);
  const presentIds = new Set(present.map((a) => a.id));
  // Base ids already reported at a nearer horizon: the nearest horizon
  // carries the report, so the operator sees one alert per condition.
  const reported = new Set<string>();
  const predicted: Alert[] = [];
  for (const lookahead of [1, 2, 3]) {
    const projected = projectState(state, lookahead);
    for (const a of runAllRules(projected)) {
      if (presentIds.has(a.id)) continue;
      if (reported.has(a.id)) continue;
      reported.add(a.id);
      predicted.push({
        ...a,
        id: `predicted-${lookahead}-${a.id}`,
        severity: demoteForHorizon(a.severity, lookahead),
        title: `IN ${lookahead} MIN: ${a.title}`,
        detail: `Forecast: ${a.detail}`,
        lookaheadMin: lookahead,
      });
    }
  }
  return [...present, ...predicted];
};
```

A.3 Representative detector: proximity-conflict

Reproduced from `src/sim/rules.ts` at this revision, with the long explanatory comments trimmed. An earlier revision of this appendix printed a version that predated three documented changes: it still excluded the `departed` phase that Section 6.8 shows removed the whole departure population, it decided the critical tier with a static distance box rather than the tau test of Section 5.4, and it had no parallel-approach demotion. A stale listing is worse than a stale sentence, because a listing looks like the code.

```
const detectProximityConflict = (state: SimState): Alert[] => {
  const alerts: Alert[] = [];
  // Airborne departures are included: "departed" here means airborne and
  // climbing out, exactly the state in which a departure meets an arrival
  // descending through the same altitude. The at-gate and ground exclusions
  // stay.
  const active = state.flights.filter(
    (f) => f.phase !== "at-gate" && f.altitudeFt > 0,
  );
  const seen = new Set<string>();
  for (let i = 0; i < active.length; i++) {
    for (let j = i + 1; j < active.length; j++) {
      const a = active[i];
      const b = active[j];
      const horizNm = Math.hypot(
        a.positionNm.x - b.positionNm.x,
        a.positionNm.y - b.positionNm.y,
      );
```

```
      const vertFt = Math.abs(a.altitudeFt - b.altitudeFt);
      if (horizNm < PROXIMITY_HORIZONTAL_NM && vertFt < PROXIMITY_VERTICAL_FT) {
        const key = [a.id, b.id].sort().join("-");
        if (seen.has(key)) continue;
        seen.add(key);
        // critical := converging (tau in (0, 45 s]) AND horizontal miss
        // distance at closest approach inside DMOD (0.5 NM) AND already
        // inside the vertical envelope (< 200 ft). A pair with no closure
        // never escalates past warning, whatever its spacing.
        const critical = isConvergingCritical(a, b, vertFt);
        // A non-converging pair on distinct parallel runways is a
        // simultaneous parallel approach, not a conflict.
        const parallel = !critical && parallelRunwayPair(a, b);
        alerts.push({
          id: `prox-${key}`,
          severity: critical ? "critical" : parallel ? "advisory" : "warning",
          category: "proximity-conflict",
          title: parallel
            ? `${a.callsign} and ${b.callsign} abreast on parallel approaches`
            : `${a.callsign} and ${b.callsign} converging`,
          flightIds: [a.id, b.id],
          reason: parallel
            ? `Attributed to distinct parallel runways with no convergence.`
            : `TCAS RA threshold ~0.5 NM horizontal / 200 ft vertical.`,
          suggestedAction:
            `Issue immediate vector or altitude change to one aircraft.`,
          createdAtTick: state.tick,
        });
      }
    }
  }
  return alerts;
};
```

A.4 Orchestrator coupling and suppression logic

This appendix previously described four orchestrator responsibilities, of which two were implemented. Verification against the code exposed the gap; the two missing behaviours have since been built, and the two inaccurate descriptions corrected. The list below states what the code does, and records what each item previously claimed, because Section 6.9 turns on the difference.

1. **De-duplication. Implemented, and the cross-horizon half was found dead.** A projected alert is dropped if its `id` appears in the present-state set, and dropped if a nearer horizon has already reported the same condition; both checks are set lookups. The second check is newer than it should be. As originally written, the predicate compared a horizon-prefixed identifier against an unprefixed one and could never match, so the same conflict reached the operator at up to three horizons at once: measured before the fix, Tenerife and Linate each reported one proximity pair twice and the runway-conflict synthetic reported it three times. No existing assertion could see the bug, because the duplicate identifiers differed by prefix and severities were identical across horizons. The fix keeps the nearest, most severe report, a regression test now fails on any base identifier seen at two horizons, and the corpus counts in Section 6 reflect the corrected output.

2. **Severity demotion of projections. Horizon-dependent, as of this revision.** Demotion deepens with the projection horizon: one step at one and two minutes out, two steps at three. This closes the open work recorded in the previous revision, which found the implementation demoting identically at

every horizon. One deliberate deviation from the original design note stands: that note let a one-minute projection keep its base severity, but here a projected alert is never critical at any horizon, because the Section 6.8 negative-control guarantee and the Section 2.6 alert-fatigue argument both reserve the critical tier for conditions holding in the present state. Weighting demotion by how far the threshold was exceeded, rather than by horizon alone, remains open.

3. **Suppression of lower-severity alerts. Implemented, after being described without being built.** `applySuppression` in `src/sim/rules.ts` holds a table of `(higher_category, suppressed_category)` pairs. When a `critical` alert carrying a `runwayId` is present, a lower-severity alert of a paired category on the same runway is marked `suppressedBy` that alert's identifier.

   Two details came out of building it. First, the pairs originally specified, a critical runway-conflict suppressing wake-spacing and cascading-delay, fire on no scenario in the corpus: they need a critical runway-conflict co-occurring with a lower-tier wake or flow alert on the same runway, and that combination does not arise. They are retained because they are correct, not because they are exercised. Second, the specification missed the redundancy that does arise. `detectRunwayConflicts` emits up to three alerts per runway, and on the `runway-conflict` scenario a `critical` "multiple aircraft on runway" sits beside a `warning` "simultaneous arrival and departure intent" describing the same unsafe occupancy at a lower tier. That pair is now in the table and is what the suppression test exercises.

   Suppressed alerts are marked rather than deleted. An operator surface that hides an alert must be able to say what hid it, and a post-incident review has to see everything the detectors found, including what the display withheld. `activeAlerts` derives the operator view by filtering them out.

4. **Alert lifecycle and flicker suppression. Implemented, in `src/sim/lifecycle.ts`.** `reconcileAlerts(previous, current, tick)` carries `firstSeenTick` and `lastSeenTick` across ticks and holds an alert that has vanished from detector output for `ALERT_GRACE_TICKS` (three, a one-minute grace period at the 20-second live poll interval) before dropping it, marking it `stale` in the interim. Severity is refreshed from the current pass, so an escalating conflict shows its new tier immediately rather than waiting out the grace period.

   This addresses a real defect rather than a documentation gap. Before it, each tick's alert list was computed from scratch with no cross-tick state, so a pair of aircraft sitting either side of a separation threshold produced an alert that appeared and vanished on consecutive polls. Flicker of that kind is among the fastest ways to lose an operator's confidence in a display, and it was invisible in scenario playback because scenarios are static.

   The function is pure: state lives with the caller, in the live store, and not inside a detector, so the determinism of Section 4.1 survives. `SimState.alerts` now carries the operator view and `SimState.trackedAlerts` the full set.

An earlier claim that these properties are unit-tested in `src/sim/predict.test.ts` was false: that file did not exist, and the repository contained no test files at all, with `vitest` configured and nothing for it to run. `src/sim/sim.test.ts` now holds 321 tests, one per verifiable claim in this document and, since 2026-09-01, a set of invariants the scenario corpus must satisfy that no claim in this document asserts: assigned runways and gates existing in the state that holds them, callsigns unique within a scenario, alerts naming only aircraft their own state carries, including the suppression and lifecycle properties above. Where a claim proved false the test pins the actual behaviour and says so, so the two cannot silently diverge again.

A.5 Historical-corpus scenario builders

Each historical scenario is a pure function `() => SimState` registered in `src/sim/scenarios.ts`. Loading a scenario freezes time (`speed: 0`), sets the airport runway geometry to the airport in question, sets the weather to the conditions reported by the relevant METAR, and places the aircraft at the positions established by the official investigation.

A.6 Reproducibility checklist

- Source: `github.com/fbritoferreira/atc-decision-support`, public since 2026-09-02, licensed Apache 2.0. Extracted from a private monorepo by `scripts/extract-standalone.sh`, which verifies the result by installing, testing, typechecking and building it.
- Sampling windows: `github.com/fbritoferreira/atc-joas-paper`, the open-data archive for the journal submission. They are not in the source repository, so the analyses there run against a locally sampled window unless fetched.
- Live deployment: `https://atc.fbritoferreira.com`. **Live traffic is currently unavailable**: both public feeds refuse the deployment, one rate-limiting the shared addresses the serverless runtime leaves through and one blocking by policy, so a visitor sees an empty radar scope. Checked 2026-09-01. The weather half of the same proxy answers normally.
- Historical scenarios accessible via the scenario picker in the header, and these are the part to open. They need no feed and are the evidence behind every detection-shape claim in Section 6.
- Data sources documented in Section 5.5; all are public and free, and none needs a key.
- **The sampling windows are distributed, and not from this repository.** `data/` here is still not committed, so the figures recomputed from the window dumps, the pooled pair counts, the proximity geometry, the violation margins and the seeded-weather share, still cannot be re-derived from a clean checkout of this repository, and six checks in the suite skip for want of them and say so by name. What changed on 2026-09-02 is that the windows themselves were published: 36 files and 1.58 MB at `github.com/fbritoferreira/atc-joas-paper`, the open-data archive for the journal submission. A reader who wants the numbers re-derived can fetch them from there and point the analysis scripts at them. Two windows still filling when the archive was assembled were left out on purpose rather than silently.
- **The headline reduction cannot be reproduced at all**, by anyone, including the author. It needs a baseline window and a 24-hour window that were not preserved. Section 6 says so where the figure appears rather than only here.
- What can be reproduced from a clean checkout: the whole test suite, the scenario corpus, the monolith-against-population comparison, and every claim the checker verifies against the code.