Why we should move FDIR onboard

James W. Masson

July 2, 2026

Abstract

Spacecraft fault detection, isolation, and recovery (FDIR) was designed for a small population of closely supervised satellites with frequent ground contact. That model assumed satellites few enough to supervise individually, ground contact frequent enough to intervene in time, and faults simple enough to catch with fixed limits. None of this still holds: the active-satellite population has risen roughly six-fold in six years [1, 2], while a satellite in low Earth orbit is beyond contact with any given ground station for more than 95% of the day, and even within contact the speed of light bounds a reaction time that fast faults exceed. Faults remain frequent, concentrated in a few subsystems, and preceded by gradual drift, yet the onboard logic in service detects them only after a hard limit is crossed. Interference, meanwhile, is now deliberate and documented across every orbital regime, so the same departure from normal may indicate either a failing component or a deliberate attack, and the reflex of entering safe mode may be what an adversary intends.

We argue that fault management must move onboard. It must sense departures from a learned nominal state before any limit is crossed, determine where a departure leads and whether it was induced deliberately, and select an authorized, auditable response within the seconds available. This paper presents NADE, the Neural Anomaly Detection Engine, an implementation in four parts: SENTRY, which senses on two independent planes, the optical inter-satellite link and platform health; CRUCIBLE, which calibrates the detectors against a physics-based twin; ARBITER, which fuses the two channels to attribute cause; and AEGIS, which recommends an authorized recovery policy. The economics favor the move: on the 2023 loss record, a single averted battery-class failure funds deployment across a 500-vehicle fleet for more than a decade.

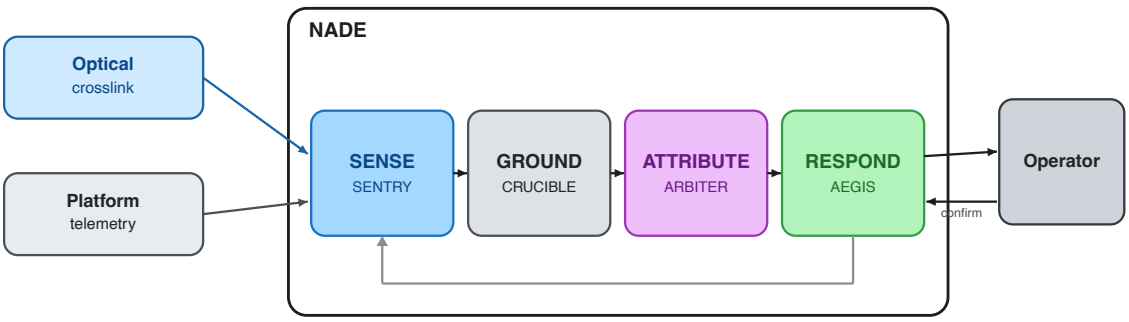

Figure 1: NADE at a glance: two independent sensing channels (the optical inter-satellite link and platform-health telemetry) feed an onboard four-stage loop (SENTRY, CRUCIBLE, ARBITER, AEGIS) that attributes cause and responds, with the operator in the loop.

1 Introduction

The space domain is being reshaped by three developments at once.

1. The active-satellite population has grown roughly six-fold in six years, to more than 14,000 operational spacecraft, faster than operators can be trained and hired to manage them [1, 2].
2. Those satellites now carry services that ground systems depend on continuously: imagery, signals intelligence, satellite communications, and the position, navigation, and timing (PNT) signals that synchro-

nize infrastructure from power grids to financial markets.

3. The satellites have become targets. The head of the U.S. Space Force frames the service's mission as the need to "defend U.S. space capabilities" and to "protect our forces from space-enabled attack" [3], and the threat is no longer hypothetical: the February 2022 cyberattack on the ViaSat KA-SAT network disabled thousands of terminals across Europe at the opening of the war in Ukraine [4, 5].

Those who buy, operate, and insure these systems describe the moment in direct terms:

> *"I believe that from a threat perspective, ground antennas and ground stations are, to me, something that's at risk in a conflict, so I think down the road, you want to take advantage of where the industry is going and really make your constellations as autonomous as possible and get human control out of the loop."*
> — Frank Calvelli, Assistant Secretary of the Air Force for Space Acquisition and Integration [6]

> *"The losses in the space insurance market are unsustainable."*
> — Melissa Quinn, General Manager, Slingshot Aerospace [7]

> *"Space superiority is our prime imperative."*
> — Gen. B. Chance Saltzman, Chief of Space Operations [8]

Together these developments expose a structural gap in how spacecraft manage their own faults. For most of every orbit a satellite is beyond the reach of any operator, so a fault that develops between contacts must be handled onboard or not at all. The logic that performs this task today was built for a smaller, closely supervised fleet, and the faults operators most want to catch, the gradual drifts that precede a loss, escape it until a hard limit is crossed. The remainder of this section examines that gap from both the internal and external sides.

1.1 Faults From Within

A satellite can operate for years without physical contact. Its components age under conditions the ground cannot inspect directly: launch vibration, thermal cycling between sunlit and eclipsed phases, sustained exposure to ionizing radiation, and the gradual erosion of battery cells under repeated partial-discharge duty. These conditions induce drift in the sensors that track each subsystem, and that drift is the precursor to failure.

The failures this produces are neither scattered nor sudden. They concentrate in a small set of subsystems, among them reaction wheels, momentum-control hardware, batteries, inertial measurement units, and transmit chains, that recurs as the dominant source of faults in incident reports across operators and orbital regimes [12]; because that population is known in advance, it can be instrumented and measured. And they announce themselves: a reaction wheel slows before it seizes, a sensor bias grows before it is noticed, and a battery cell loses a fraction of its rated capacity over weeks of cycling before any single discharge crosses a hard limit. The telemetry that ground stations already collect carries an early signature of the failure; what is typically absent is anything onboard able to interpret that signature before a threshold is crossed [13].

The faults operators most want to catch are therefore the ones limit-based logic is least equipped to catch: it is organized around fixed thresholds, and the signature arrives as a gradual departure from baseline long before any threshold is crossed. NADE closes this gap for the spacecraft's internal subsystems. The next subsection turns to the case in which the same telemetry signature is produced deliberately, from outside the spacecraft.

1.2 Adversarial Interference

Interference with satellites now spans nearly every orbital regime and takes many forms: jammers directed at satellite communications and GPS, anti-satellite weapons, on-orbit grapplers, optical dazzlers, and cyberattacks on ground stations and spacecraft alike [14]. GPS jamming and spoofing have become near-daily events across the Arctic, Eastern Europe, the Middle East, and parts of South Asia; in a single year, spoofing incidents rose roughly 500% and broader satellite-navigation (GNSS) interference roughly 175% [15, 16]. The capability extends to the highest orbits: in January 2022 the Chinese satellite Shijian-21 docked with a defunct navigation satellite and towed it to a graveyard orbit roughly 3,000 km above the geostationary belt, demonstrating the ability to take physical hold of another spacecraft [17, 18].

The effects can be significant even when a satellite is never touched. During a 2026 internet shutdown in Iran, GPS jamming degraded Starlink terminal performance by as much as 80% in parts of the country, because the terminals rely on the GPS signal to acquire the constellation [19]. A fault can now be induced on purpose, and on the spacecraft's own telemetry a lost lock, a drifting attitude, or a degraded link reads the same whether a component failed or an adversary made it fail. Several of the subsystems that fail most often are also those an adversary can disrupt, a star tracker dazzled, a link jammed, so the cause of a deviation can no longer be read off its symptom.

2 The Case for Autonomous FDIR

2.1 Scalability

The number of vehicles an operator must oversee has outrun any pool of people that could be hired to watch them. Starlink alone now operates roughly 10,400 spacecraft [20], more than five times the entire active satellite fleet of 2018 and well over half of all working satellites today. The pipeline is larger still: filings to the International Telecommunication Union (ITU) from China total on the order of 200,000 satellites, and U.S. regulators have authorized a 7,500-spacecraft second-generation Starlink constellation [21]. A trained operator can attend to only a fixed number of vehicles, so the operator-to-satellite ratio falls with every launch. The routine judgment of whether a satellite is healthy must therefore be made automatically, with people reserved for the rare decisions that require human authority.

2.2 Geometry of Contact

A satellite in low Earth orbit (LEO) is visible to any single ground station for only five to twelve minutes per pass and for four to six passes per day, yielding perhaps thirty to sixty minutes of contact and leaving the vehicle unobserved by that station for more than 95% of each day [9]. When a link is available, the speed of light establishes a latency floor that bandwidth cannot reduce: round-trip propagation is approximately a quarter of a second to geostationary orbit (GEO) and roughly 2.6 seconds to cislunar distances [22, 23]. For the majority of every orbit, therefore, no operator is positioned to observe the spacecraft or to issue a command, and even during a contact a closing conjunction or a developing electrical fault may evolve faster than a command can be returned. The industry is already moving this work onboard rather than expanding ground contact: processing is migrating to the spacecraft as constellations grow, and a single constellation has executed on the order of 50,000 autonomous collision-avoidance maneuvers over six months, each determined onboard without human intervention [24].

2.3 Frequency and Cost of Faults

Of small satellites launched between 2000 and 2016, 41.3% failed or partially failed and 24.2% were total losses [10], and across a reliability analysis spanning 1,584 satellites the failures concentrate in attitude control and telemetry [12]. The cost of missing the drift that precedes such a loss is now substantial: 2023 was a record year for space-insurance losses, with about $995M in claims against about $557M in premiums, including a single $348M claim for an on-orbit battery failure [11]. A loss of that size dwarfs the cost of equipping a fleet with NADE. And the early signal is usually there to be caught: in the design scenario of Appendix A, NADE flags a degrading battery roughly 30 minutes before it would trip the low-charge limit, while intervention is still cheap. A limit check waiting for the hard threshold would forfeit that entire margin.

Table 1 works the resulting economics for a reference fleet. Every rate in it is cited; the entries marked (A) are stated assumptions, given as ranges, and the conclusion is not sensitive to them: across the full assumption range, the expected loss avoided exceeds the cost of deployment. Against a single event the comparison is starker. One avoided loss of the 2023 battery-failure class ($348M [11]) funds NADE across a 500-vehicle fleet for more than a decade at the upper-bound cost assumption. The market is already pricing the status quo: the 2023 loss year drove several long-standing underwriters out of space insurance altogether [25], and detection that arrives before a limit is crossed is the only intervention that changes the expected-loss column rather than the premium column.

2.4 Attribution: Telling an Attack From a Fault

On the platform bus, a failing component and a deliberate attack can produce the same telemetry, so monitoring that bus alone can never separate them. A second, physically independent channel is required, and NADE obtains it from the optical inter-satellite link, the channel now becoming the backbone of new constellations yet still unmonitored at the physical layer. When undisturbed, a coherent laser link has a fixed signature: its photon statistics sit at unit Fano factor $F = 1$, zero Mandel parameter $Q = 0$, and flat second-order coherence $g^{(2)}(0) = 1$. The link's own data does not disturb this signature: legitimate traffic is phase-modulated, and photon-counting statistics are blind to phase. An injection cannot manage the same, because a second field co-propagating with the signal beats against it, and the interference disturbs several of the fourteen photon-statistical features NADE's optical detector tracks at once, which makes a clean imitation hard to sustain. The detector scores each window of the live link against a learned model of the nominal state and fires at a threshold set from an operator-chosen false-alarm budget; Appendix A gives the formal detection contract and the fusion of this evidence with platform health. A power monitor on the bus carries no comparable signature, and no other channel on the spacecraft offers one; the optical link is what turns a deviation into a diagnosis.

Table 1: Five-year fault economics for a reference fleet of 100 LEO vehicles at $10M per vehicle. Cited rates are drawn from the small-satellite failure record and the 2023 insurance year; entries marked (A) are assumptions, stated as ranges.

	Limit-based FDIR	**With NADE**
Expected total losses across the fleet (24.2% of vehicles [10]; $10M per vehicle (A))	$242M	$242M at risk
Share of losses in drift-preceded subsystem classes: batteries, reaction wheels, IMUs, transmit chains [12] (50% (A))	$121M	$121M addressable
Fraction of addressable losses averted given a pre-limit warning and a protective action (A)	0% (no warning issued)	33–50%
Expected loss averted	—	$40–61M
Cost of deployment, 5 years (over-the-air load; $10–50k per vehicle-year (A))	$0	$5–25M
Net expected benefit	—	**$15–56M**

3 Design Principles

Six principles govern NADE's architecture.

Detection by learned normal. The space of possible faults and attacks cannot be enumerated in advance; normal behavior, by contrast, is abundant and directly observable. Each detector learns a model of normal and measures the distance of live data from it, so a departure it has never seen still registers [26, 27].

Thresholds derived from a stated false-alarm rate. An alarm is useful only if the operator knows, and can adjust, how often it will fire when nothing is wrong. Every detection threshold in NADE is computed from an explicit false-alarm budget.

Physics-based prediction. A fault that cannot be safely induced on a real spacecraft can be induced freely on a digital twin, which also supplies labeled truth in quantity. Forecasting and evaluation are therefore built on a simulated vehicle whose dynamics match those of the real one.

Evaluation against boundary cases. The most informative scenarios are those that nearly cross the line between safe and failed: they are the hardest to classify and the ones an adversary would deliberately seek. NADE's test suites are built around such near-miss cases, in addition to unambiguous failures.

Local operation. Contact windows are short, communications can be denied, and sensor data is often sensitive, so every component runs locally on modest hardware, depends on no remote service, and continues at reduced confidence when an input is lost.

Inspectable reasoning. A recommendation that a person must authorize has to expose how it was reached, or that person cannot responsibly approve it. Every component therefore emits forensic evidence or a reasoning trace alongside its conclusion.

4 How NADE Works

NADE, the Neural Anomaly Detection Engine, is a software layer that runs *on the spacecraft*. Within the seconds a fault allows, it determines whether something has gone wrong, whether it was made to go wrong deliberately, and what to do about it. Onboard autonomy of this kind is not itself new; a remote agent flew on Deep Space 1 a quarter-century ago [28]. What is new is the operational picture that now requires it. Where conventional FDIR waits for a telemetry value to cross a fixed limit and then applies a pre-scripted remedy [29], NADE learns a model of normal and measures the distance from it, so it can flag a problem it has never been shown. It installs as firmware on hardware already in orbit and depends on no ground link that may be minutes away or denied.

Four stages on one decision path. NADE is organized as a closed loop (Figure 1); the full architecture and supporting mathematics are given in Appendix A.

- **SENTRY** senses on two physically independent planes: CLARION on the optical inter-satellite link and VITALS on the platform's own health telemetry. Each scores how far the live signal has drifted from a learned model of normal, $s_\theta = -\log \hat{p}_{\text{nom}}(\mathbf{x} \mid \mathbf{c})$, and fires when the score exceeds a threshold tuned to an operator-set false-alarm rate.
- **CRUCIBLE** grounds those judgments in physics. A digital twin of the vehicle generates the labeled, near-failure scenarios the detectors are calibrated against, fixing each detector's operating point at the chosen false-alarm budget.
- **ARBITER** fuses the two planes into a posterior over three hypotheses, $P(H \mid e_L, e_P)$ with $H \in \{\text{nominal, fault, attack}\}$, and reports the most probable cause with its supporting evidence, together with a candidate response for AEGIS to gate. A failing component and a hostile intrusion can read

identically on the platform bus; only the intrusion also disturbs the optical link.

- **AEGIS** acts within granted authority. It takes a bounded, reversible protective action on its own (safing a payload, quarantining a compromised crosslink) and escalates anything consequential to a human, with the full reasoning trace attached.

The sensing stack is small enough to run inside the control loop of a radiation-tolerant processor: the optical-plane pair carries fewer than 10,000 parameters and evaluates a window in roughly 90 microseconds, within the dynamics tick of current smallsat flight hardware (the demonstrator of Appendix A). Detection, fusion, and a bounded response policy therefore execute onboard, between ground contacts.

The loop is concrete enough to walk end to end on Figure 4's schematic scenario. A battery cell begins to fade: state-of-charge that should hold near 80% under the commanded duty cycle instead falls from 82% to 56% in thirty minutes, roughly 0.9 points per minute, while every individual reading stays far above the 30% low-charge trip. VITALS's residual score rises as the discharge departs from learned normal, and its forecast puts the probability of crossing the 30% limit above the actionability level β at a horizon of about thirty minutes: TTF ≈ 30 min. ARBITER sees a high platform score against a quiet optical link and attributes fault, not attack. The verifier finds a charge warning with TTF under the protective threshold and admits `disable_payload`; the selection step confirms that shedding the payload keeps the safety probability above $1 - \epsilon$ at the cost of one deferred imaging pass, and AEGIS executes it, logging the alert, the attribution, the admitting rule, and the reasoning trace. A limit-based system takes its first action thirty minutes later, at the trip line, with the battery thirty minutes deeper into failure and the cheap options gone.

NADE differs from classical FDIR and from generic anomaly detection in what it combines. The optical channel gives it attribution, an independent line of evidence that distinguishes sabotage from malfunction, where the two call for different responses. The pipeline is small enough to live on a radiation-tolerant flight processor, and it emits an inspectable reasoning trace with every decision, so an operator, and in time a certifying authority, can reconstruct why it acted [13]. And deployment is an over-the-air update to existing optical terminals and flight software, not a new hardware program.

Thus, NADE serves two buyers: defense and government operators, for whom it closes a documented monitoring gap on the constellations that carry national-security traffic; and commercial mega-constellation operators, for whom it turns fault management from a ground-bound, contact-limited task into an autonomous function that scales with the fleet.

5 Limitations

5.1 The Simulation-to-Flight Gap

Every advantage described here rests on a model of normal, which is only as good as the data it is learned from. A digital twin produces a physically consistent normal; a real spacecraft produces more: thermal cycling in and out of sunlight, slow component aging, the idiosyncrasies of individual sensors, and a long tail of benign irregularities that experienced operators learn to ignore [13]. A detector trained only on simulated normal would treat some of that variation as anomalous and raise its false-alarm rate. More than any new method, what would advance the field is access to real-time, fleet-scale telemetry from operating constellations, against which to learn true normal and measure the simulation-to-flight gap directly. That dependency is shared by everyone pursuing the approach, and it begins to resolve as soon as a fleet operator runs the system on its own data.

5.2 Adversarial Robustness

A model of normal can in principle be evaded by a fault or an injection shaped to look normal, or provoked into false alarms by an adversary who has learned its blind spots [26]. The optical channel constrains this more tightly than a power monitor does (an injected payload must carry modulation structure, which perturbs several photon-statistical features at once and cannot be zeroed out without abandoning the payload), but a determined adversary holding a model of the detector remains a real threat. Boundary-search calibration is the natural tool for probing and hardening against it; a rigorous characterization of the residual risk does not yet exist.

5.3 Confidence of Attribution

ARBITER treats the two sensing channels as conditionally independent given the cause. That is the standard naive-Bayes assumption, and it can fail: an attack engineered to disturb both channels in a correlated way could blur the fault-versus-attack boundary the system relies on. The posterior is also only as well-calibrated as its priors, which themselves depend on flight data. When the two channels disagree or confidence is low, the design routes the decision to a human, so miscalibration produces extra escalations to the operator rather than unauthorized actions.

5.4 The Flight-Hardware Budget

Running detection and fusion inside the dynamics tick of a radiation-tolerant processor, with the physics twin and a reasoning policy sharing the same board, is a real constraint. The base detectors fit with margin to spare, as noted in Section 4, but the onboard reasoning agent is the most compute-hungry element and the most likely

to require either a capable space-qualified processor or a reduced on-orbit configuration.

5.5 Generalization Across Platforms

A model of normal is learned per platform, and the bus, the payload, and the orbit each shape what normal means. Transferring a calibrated detector to a new vehicle class without re-learning is not yet demonstrated, and we expect per-class calibration to be necessary in the near term. The twin makes that calibration inexpensive to produce; what has not yet been driven down is the per-platform cost of onboarding a new vehicle type.

5.6 Certification and Data Access

Flight software is held to standards that assume deterministic, inspectable behavior [30], and a learned, partly probabilistic component does not map onto them without work. The reasoning trace and authority gating are designed to give a certifying authority something to audit, but the certification path for autonomy of this kind is still being written. Separately, defense primes may be unwilling to share classified telemetry with a third party, which argues for an operator-run or on-premises deployment model rather than a hosted one.

5.7 Retained Need for Human Oversight

Calibrated confidence, reasoning traces, and authority routing exist to keep a person able to understand and, where appropriate, veto what the system does. Which actions a spacecraft may take alone and which it must escalate is a doctrinal question as much as a technical one, and we do not attempt to settle it here. The system lets the operator draw that line and redraw it as doctrine changes.

6 Conclusion

This paper has argued that fault management belongs onboard. The pressures behind that claim are not going away: the fleet will keep outgrowing its operators, interference is now deliberate, and the optical links that carry national-security traffic remain unmonitored at the physical layer. Meanwhile flight compute has become equal to running a learned model of normal between ground contacts. NADE is one response, detection calibrated to a stated false-alarm budget, attribution drawn from a physically independent channel, and a graduated response a person can audit, on hardware already in orbit. The economics are not close: a single averted loss of the 2023 class pays for fleet-wide deployment many times over (Table 1). The open problems of Section 5 are questions of data and engineering; what remains is the work of carrying the system to flight. That work is incremental by design: the three-vehicle demonstrator already closes the sense–attribute–respond loop in simulation; the same stack next runs hardware-in-the-loop against a radiation-tolerant flight processor; a hosted pilot then carries the sensing planes on an operating vehicle; and fleet-wide deployment follows as an over-the-air load. Each step reuses the software of the one before it, and each produces the flight telemetry that closes the simulation-to-flight gap of §5.1.

References

[1] European Space Agency. *Space Environment Statistics*. Space Debris User Portal, ESA Space Debris Office. Approximately 14,200 functioning satellites; statistics updated January 2026. 2026. URL: https://sdup.esoc.esa.int/discosweb/statistics/ (visited on 07/01/2026).

[2] Union of Concerned Scientists. *UCS Satellite Database: A Record Number of Satellites in Orbit*. UCS Blog. Data as of 30 Nov 2018. 2018. URL: https://blog.ucs.org/lgrego/2018satellitedata/ (visited on 06/02/2026).

[3] B. Chance Saltzman. *Space Warfighting: A Framework for Planners*. Foreword dated 2 March 2025. United States Space Force, Mar. 2025. URL: https://www.spaceforce.mil/Portals/2/Documents/SAF_2025/Space_Warfighting_-_A_Framework_for_Planners_BLK2_(final_20250410).pdf (visited on 06/02/2026).

[4] Viasat Inc. *KA-SAT Network Cyber Attack Overview*. Viasat Inc., Mar. 2022. URL: https://www.viasat.com/about/newsroom/blog/ka-sat-network-cyber-attack-overview/ (visited on 06/25/2026).

[5] Nicolò Boschetti, Nathaniel G. Gordon, and Gregory Falco. "Space Cybersecurity Lessons from the ViaSat Cyberattack". In: *ASCEND 2022*. AIAA, 2022.

[6] Defense One. *Near-autonomous satellites could be coming in a decade, Space Force envisions*. Defense One. Remarks of Frank Calvelli, Assistant Secretary of the Air Force for Space Acquisition and Integration, at the NDIA Emerging Technologies for Defense conference. Aug. 2024. URL: https://www.defenseone.com/technology/2024/08/space-force-envisions-near-autonomous-satellites-decade-or-so/398694/ (visited on 07/02/2026).

[7] The Register. *Insurers make record-breaking loss as space gets cramped*. The Register. May 2024. URL: https://www.theregister.com/2024/05/01/space_insurer_record_loss/ (visited on 07/02/2026).

[8] Air and Space Forces Magazine. *'Whatever it Takes': Saltzman Says 'Space Superiority' Is USSF's Mission*. Air & Space Forces Magazine. Keynote of Gen. B. Chance Saltzman, AFA Warfare Symposium, 3 March 2025. Mar. 2025. URL: https://www.airandspaceforces.com/saltzman-space-force-destruction-offensive-space/ (visited on 07/02/2026).

[9] "Practical Horizon Plane and Communication Duration for LEO Satellite Ground Stations". In: *WSEAS Transactions on Communications* (2009). LEO contact-minutes-per-day derived from standard orbital geometry (∼550 km circular orbit). URL: https://www.wseas.us/e-library/transactions/communications/2009/29-213.pdf (visited on 06/02/2026).

[10] Stephen A. Jacklin. *Small-Satellite Mission Failure Rates*. Technical Memorandum NASA/TM-2018-220034. NASA Ames Research Center, Mar. 2019. URL: https://ntrs.nasa.gov/api/citations/20190002705/downloads/20190002705.pdf (visited on 06/02/2026).

[11] Slingshot Aerospace. *State of Satellite Deployments & Orbital Operations 2023*. Slingshot Aerospace, Apr. 2024. URL: https://www.satellitetoday.com/sustainability/2024/05/01/slingshot-aerospace-reveals-record-insurance-losses-in-2023-in-new-satellite-deployments-report/ (visited on 06/02/2026).

[12] Jean-François Castet and Joseph H. Saleh. "Satellite and satellite subsystems reliability: Statistical data analysis and modeling". In: *Reliability Engineering & System Safety* (2009). See also: "Epidemiology of satellite anomalies and failures," *Acta Astronautica* (2011). DOI: 10.1016/j.ress.2009.05.004.

[13] R. Capelli, J. de Souza Rosa, G. De Tommasi, et al. "On-board Telemetry Monitoring in Autonomous Satellites: Challenges and Opportunities". In: *arXiv preprint* (Apr. 2026). arXiv: 2604.08424. URL: https://arxiv.org/html/2604.08424v1 (visited on 06/02/2026).

[14] B. Chance Saltzman. *Opening Keynote, Mitchell Institute 3rd Annual Spacepower Security Forum*. Official transcript, Mitchell Institute for Aerospace Studies. Mar. 2024. URL: https://www.mitchellaerospacepower.org/app/uploads/2024/04/03.27.24-Gen-Saltzman-Transcript-1.pdf (visited on 06/02/2026).

[15] Clayton Swope et al. *Space Threat Assessment 2025*. CSIS Aerospace Security Project, Apr. 2025, p. 17. URL: https://csis-website-prod.s3.amazonaws.com/s3fs-public/2025-04/250425_Swope_Space_Threat.pdf (visited on 06/02/2026).

[16] International Air Transport Association. *IATA Releases 2024 Safety Report*. Press release. Feb. 2025. URL: https://www.iata.org/en/pressroom/2025-releases/2025-02-26-01/ (visited on 06/02/2026).

[17] *China's Shijian-21 spacecraft docked with and towed a dead satellite to a graveyard orbit*. SpaceNews. Jan. 2022. URL: https://spacenews.com/chinas-shijian-21-spacecraft-docked-with-and-towed-a-dead-satellite/ (visited on 06/02/2026).

[18] Kari A. Bingen, Clayton Swope, et al. *Space Threat Assessment 2024*. CSIS Aerospace Security Project, Apr. 2024. URL: https://aerospace.csis.org/wp-content/uploads/2024/04/240417_Swope_SpaceThreatAssessment_2024.pdf (visited on 06/02/2026).

[19] *Iran's internet shutdown crippled Starlink, and why the world should care*. Rest of World. Jan. 2026. URL: https://restofworld.org/2026/iran-starlink-internet-shutdown/ (visited on 06/02/2026).

[20] Jonathan McDowell. *Starlink Statistics*. planet4589.org. As of 25 May 2026. 2026. URL: https://planet4589.org/space/con/star/stats.html (visited on 06/02/2026).

[21] *China files ITU paperwork for megaconstellations totaling nearly 200,000 satellites*. SpaceNews / IEEE ComSoc Technology Blog. Jan. 2026. URL: https://techblog.comsoc.org/2026/01/13/china-itu-filing-to-put-200k-satellites-in-orbit-fcc-authorizes-7-5k-additional-starlink-leo-satellites/ (visited on 06/02/2026).

[22] *Geostationary satellite latency and time delay*. satsig.net. n.d. URL: https://www.satsig.net/latency.htm (visited on 06/02/2026).

[23] *How fast does light go to the Moon and Earth?* IERE. Cislunar one-way ∼1.28 s. n.d. URL: https://iere.org/how-fast-does-light-go-to-the-moon-and-earth/ (visited on 06/02/2026).

[24] Tereza Pultarova. *SpaceX Starlink satellites made 50,000 collision-avoidance maneuvers in the past 6 months*. Space.com. Analysis by Prof. Hugh Lewis. July 2024. URL: https://www.space.com/spacex-starlink-50000-collision-avoidance-maneuvers-space-safety (visited on 06/02/2026).

[25] Data Center Dynamics. *Paying the premium: Why 2023 was a bad year for space insurance*. Data Center Dynamics. 2024. URL: https://www.datacenterdynamics.com/en/analysis/paying-the-premium-why-2023-was-a-bad-year-for-space-insurance/ (visited on 07/02/2026).

[26] Guansong Pang et al. "Deep Learning for Anomaly Detection: A Review". In: *ACM Computing Surveys* 54.2 (2021). DOI: 10.1145/3439950.

[27] Kyle Hundman et al. "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding". In: *Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 2018. DOI: 10.1145/3219819.3219845.

[28] Nicola Muscettola et al. "Remote Agent: To Boldly Go Where No AI System Has Gone Before". In: *Artificial Intelligence* 103.1–2 (1998). See also JPL, "Remote Agent Experiment Meets All Objectives" (1999), pp. 5–47. DOI: 10.1016/S0004-3702(98)00068-X.

[29] European Space Agency. *Fault Detection, Isolation and Recovery*. TEC Software Engineering and Standardisation. n.d. URL: https://www.esa.int/TEC/Software_engineering_and_standardisation/TEC4WBUXBQE_0.html (visited on 06/02/2026).

[30] RTCA. *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. RTCA, Inc. Companion standard: DO-254, airborne electronic hardware. 2011.

[31] Patrick W. Kenneally, Scott Piggott, and Hanspeter Schaub. "Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework". In: *Journal of Aerospace Information Systems* 17.9 (2020). AVS Lab, University of Colorado Boulder. https://avslab.github.io/basilisk/, pp. 496–507. DOI: 10.2514/1.I010762.

[32] Autonomous Vehicle Systems (AVS) Laboratory. *Basilisk: An Astrodynamics Simulation Framework*. https://avslab.github.io/basilisk/. Accessed 2026-06-02. 2026.

[33] Roy J. Glauber. "Coherent and Incoherent States of the Radiation Field". In: *Physical Review* 131 (1963), pp. 2766–2788.

[34] Leonard Mandel and Emil Wolf. *Optical Coherence and Quantum Optics*. Cambridge University Press, 1995.

[35] Hemani Kaushal and Georges Kaddoum. "Optical Communication in Space: Challenges and Mitigation Techniques". In: *IEEE Communications Surveys & Tutorials* 19.1 (2017), pp. 57–96.

[36] Larry C. Andrews and Ronald L. Phillips. *Laser Beam Propagation through Random Media*. 2nd ed. SPIE Press, 2005.

[37] Peter J. Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4 (1995), pp. 711–732. DOI: 10.1093/biomet/82.4.711.

[38] David A. Vallado et al. "Revisiting Spacetrack Report #3". In: *AIAA/AAS Astrodynamics Specialist Conference*. AIAA 2006-6753. 2006. URL: https://celestrak.org/publications/AIAA/2006-6753/ (visited on 06/02/2026).

[39] Yaakov Bar-Shalom, X.-Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Wiley-Interscience, 2001. DOI: 10.1002/0471221279.

A Architecture

The product is built from four cooperating modules, each owning one stage of a single decision path: SENTRY observes, CRUCIBLE grounds those observations in physics and calibrates the detectors, ARBITER assigns a cause, and AEGIS chooses a response within its authority. They talk over a shared, time-synchronized bus and pass each other typed results, so any one can be retrained or replaced without disturbing the rest. The whole stack is sized for the radiation-tolerant single-board computers that current and near-term smallsat buses carry, and every module writes an inspectable account of each decision it makes. Together they form the closed cycle of Figure 2, taking a vehicle from a raw observation to an authorized action.

The quantitative results below (Figure 3) come from a demonstrator of three low-Earth-orbit vehicles at 500 km and 51.6° inclination, built in Basilisk [31, 32] on a vehicle model of a TechEdSat-class 3U CubeSat. All three run inside one shared physical model that couples eclipse and sun geometry, an exponential atmosphere whose density follows the F10.7 solar index, a WMM geomagnetic field, and line-of-sight to six ground stations. Each orbit is initialized from a current two-line element set advanced to the present epoch, so the geometry the system reasons over corresponds to spacecraft actually flying today.

A.1 SENTRY: Two Calibrated Sensing Planes

On each vehicle, SENTRY is the sensing layer. It runs two detectors over channels that share no failure physics: CLARION on the optical inter-satellite link and VITALS on the platform's own health telemetry. CLARION reads the receive path of the optical terminal's signal processing, and VITALS reads the per-tick telemetry stream the onboard flight software already emits. Both emit the single output format described next, which is what lets ARBITER weigh them on equal footing, and every alert they raise is stamped with the false-alarm rate it was tuned to.

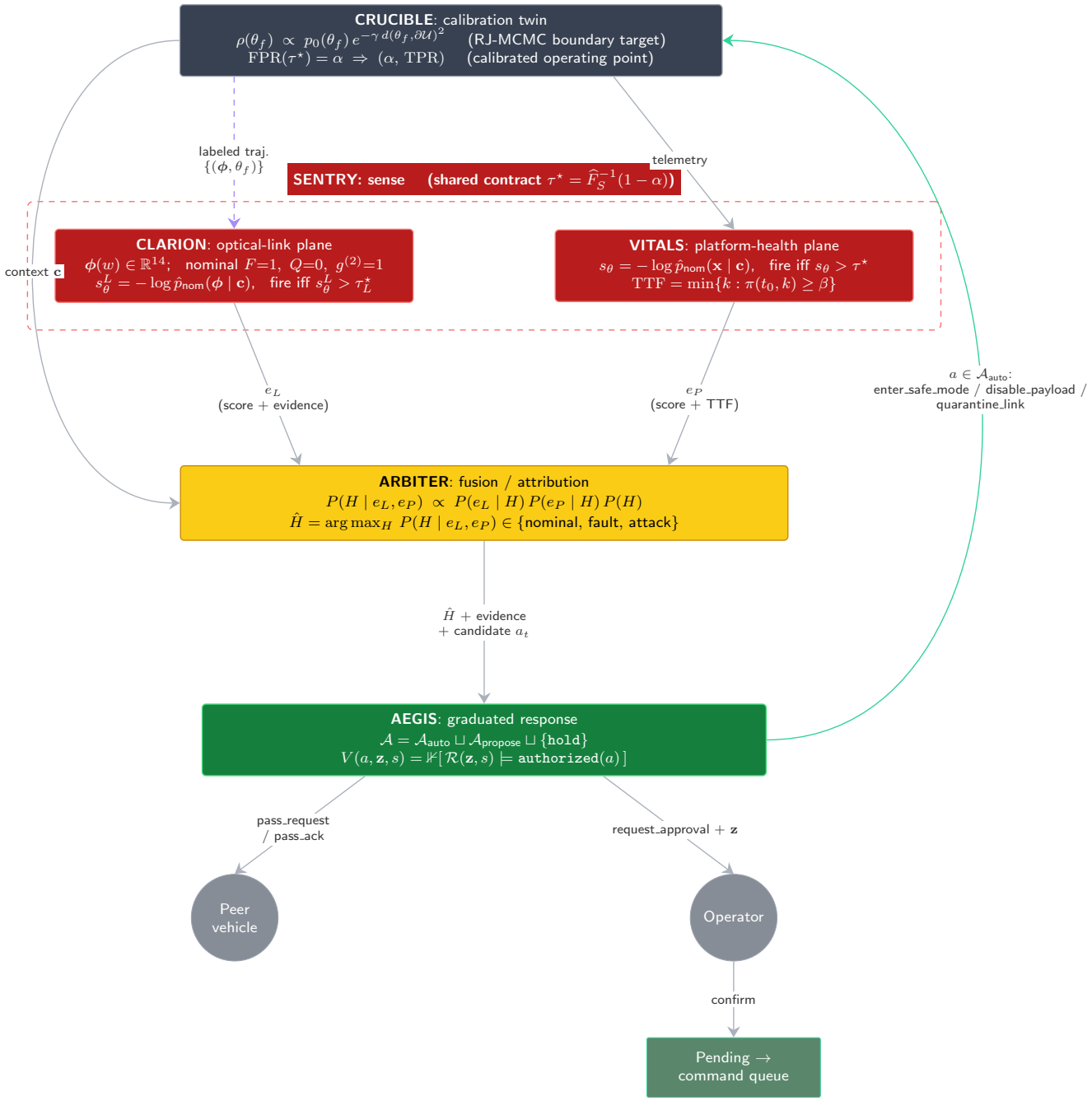

Figure 2: NADE's defensive architecture, annotated with the governing relation of each module and the quantity carried on each flow. CRUCIBLE calibrates both planes against a physics twin whose RJ-MCMC explorer concentrates on near-boundary cases and fixes each detector's operating point at the operator's false-alarm budget α. SENTRY senses on two co-equal planes that expose the same score-and-threshold contract: CLARION on the optical inter-satellite link (photon-statistical fingerprint ϕ) and VITALS on platform health (with a predictive time-to-failure). ARBITER fuses the independent evidence channels e_L and e_P into a posterior over {nominal, fault, attack}, and AEGIS maps the attribution $\hat{H}$ to a graduated, authority-gated response, coordinating peers over a per-constellation bus.

A.1.1 The Unified Detection Contract

ARBITER can only weigh the two planes against each other if both report their findings in one format. SENTRY therefore fixes a single detection interface, and each plane delivers the same three quantities: an anomaly score s_θ, a firing threshold $\tau^\star$, and the false-alarm budget α that the threshold encodes. The rest of this subsection defines those quantities.

At each tick a plane assembles two inputs. The first is

an observation vector $\mathbf{x}_t$ (a telemetry vector for VITALS, a photon-statistics feature vector for CLARION). The second is a context vector $\mathbf{c}_t$ recording the operating regime: orbit phase, eclipse state, commanded mode, and time since the last ground contact. A network trained on nominal operation models the conditional density $p_{\text{nom}}(\mathbf{x} \mid \mathbf{c})$, and the anomaly score is the negative log-likelihood of the current observation under that model:

$$s_\theta : \mathcal{X} \times \mathcal{C} \to \mathbb{R}_{\geq 0}, \quad s_\theta(\mathbf{x}, \mathbf{c}) \;\approx\; -\log \hat{p}_{\text{nom}}(\mathbf{x} \mid \mathbf{c};\, \theta). \tag{1}$$

The score stays near zero while the observation matches nominal behavior and grows as it departs from it. Nothing in the interface fixes how the score is computed. A sliding-window predictor judged on residual variance, a normalizing flow trained one-class, a masked autoencoder, or a contrastively trained density-ratio model all yield a score of this kind, and the two planes do not choose the same one.

The threshold turns that score into a decision with a known error rate. The operator selects α, the fraction of nominal ticks permitted to raise an alert. On a held-out nominal set $\{(\mathbf{x}^{(i)}, \mathbf{c}^{(i)})\}_{i=1}^{N}$, the threshold $\tau^\star$ is set to the $(1-\alpha)$-quantile of the empirical score distribution:

$$\begin{aligned} \tau^\star \;&=\; \widehat{F}_S^{-1}(1-\alpha), \\ \widehat{F}_S(\tau) \;&=\; \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\Big[s_\theta(\mathbf{x}^{(i)}, \mathbf{c}^{(i)}) \leq \tau\Big]. \end{aligned} \tag{2}$$

A plane fires whenever $s_\theta(\mathbf{x}_t, \mathbf{c}_t) > \tau^\star$. Because $\tau^\star$ is computed from α and stored with the rest of the detector configuration, the false-alarm rate behind any alert can be read back from that configuration after the fact. Both planes deliver the triple $(s_\theta, \tau^\star, \alpha)$ to ARBITER.

A.1.2 CLARION: The Optical-Link Plane

CLARION instantiates the detection contract on the optical inter-satellite link, where it serves as the system's independent evidence channel for deliberate interference. Optical crosslinks are replacing RF as the backbone of military and commercial constellations, yet no operational link carries physical-layer intrusion monitoring. The most dangerous active attack, *beam-riding injection*, co-propagates adversary photons along the legitimate beam to insert forged commands, spoofed telemetry, or falsified routing updates without interrupting the signal. Because injection only *adds* power, a monitor that thresholds total power misses it at the modest strengths a capable adversary would actually choose. CLARION instead watches the photon-statistical fingerprint of the link.

What nominal light obeys. Laser light from a well-behaved transmitter is a coherent state: over a $100\,\mu$s window the photon count n at mean μ is Poisson,

$$p_{\text{nom}}(n \mid \mu) = \frac{\mu^n e^{-\mu}}{n!}, \qquad \mathbb{E}[n] = \mathrm{Var}(n) = \mu. \tag{3}$$

This fixes a set of coherence laws (unit Fano factor, zero Mandel parameter, flat second-order coherence, and exponentially distributed photon arrivals) [33, 34]:

$$F \equiv \frac{\mathrm{Var}(n)}{\mathbb{E}[n]} = 1, \quad Q \equiv F - 1 = 0, \quad g^{(2)}(\tau) = 1 \;\; \forall \tau, \tag{4}$$

$$p_{\text{nom}}(\Delta t) = \lambda\, e^{-\lambda \Delta t}, \qquad \lambda = \mu / T_{\text{win}}. \tag{5}$$

These laws hold on an operating link, not only an idle one. The payload of a coherent optical link rides in the optical phase (binary or quadrature phase-shift keying), and phase modulation leaves the intensity envelope, and with it the photon-count statistics, unchanged: a phase-modulated coherent state obeys the same Poisson law as the bare carrier. Legitimate traffic therefore imposes no structure on the fingerprint that follows.

From each window CLARION extracts a 14-feature fingerprint

$$\begin{aligned} \boldsymbol{\phi}(w) = \big[\, & n,\, F,\, g^{(2)}(\tau_1), \ldots, g^{(2)}(\tau_6),\, \mathrm{IAT}_\mu, \\ & \mathrm{IAT}_{\sigma^2}, \mathrm{IAT}_\gamma, \mathrm{IAT}_\kappa,\, D_{\text{KS}},\, \mathsf{H} \big] \in \mathbb{R}^{14}, \end{aligned} \tag{6}$$

collecting the photon count, Fano factor, second-order coherence at six lags, the first four inter-arrival-time moments, a Kolmogorov–Smirnov statistic against the theoretical exponential, and the Hurst exponent H (written sans-serif to keep it distinct from the cause hypothesis H of the fusion stage).

What injection does. An adversary injecting at power fraction

$$s = \frac{P_{\text{inj}}}{P_{\text{sig}} + P_{\text{inj}}} \in [0, 1], \qquad \Delta P_{\text{inj}} = s\, P_{\text{sig}}, \tag{7}$$

must carry a payload it cannot hide in the phase of a carrier it does not control: its field beats against the legitimate signal at the detector, and its own modulation adds intensity structure on top. Both effects perturb the coherence laws in correlated ways: the stream becomes super-Poissonian, bunched, and non-exponential:

$$\begin{aligned} & F(s) > 1, \quad Q(s) > 0, \quad g^{(2)}(0; s) > 1, \\ & D_{\text{KS}}\big(\hat{p}(\Delta t),\, \lambda e^{-\lambda \Delta t}\big) > 0 \quad (s > 0). \end{aligned} \tag{8}$$

CLARION scores the fingerprint under the same unified contract as the platform plane and selects its threshold from the same false-alarm budget:

$$s_\theta^L(w) \approx -\log \hat{p}_{\text{nom}}\big(\boldsymbol{\phi}(w) \mid \mathbf{c}\big), \qquad \tau_L^\star = \widehat{F}_{S^L}^{-1}(1-\alpha). \tag{9}$$

Why the fingerprint beats a power meter. A capable adversary keeps the injected power low, since saturating the beam is both conspicuous and unnecessary. A realistic attack therefore sits at modest strength ($s \leq 0.3$), where the added photons are hidden by the ± 5–10% intensity drift that pointing jitter and shifting link geometry already produce [35, 36]. A detector limited to

total power cannot react until the injected component rises above that drift,

$$\begin{aligned} &\text{power-only detectable} \\ &\iff s\,P_{\text{sig}} \gtrsim z_{1-\alpha}\,\sigma_{\text{benign}}, \\ &\text{with } \sigma_{\text{benign}} \approx \kappa\,P_{\text{sig}}, \quad \kappa \in [0.05,\, 0.10], \end{aligned} \tag{10}$$

which, with $z_{1-\alpha}$ the standard-normal quantile of the alarm budget α, ties its smallest detectable injection to the size of that drift. CLARION reads the whole distribution instead of its mean, so the structure the injected modulation leaves behind is visible even when the total count looks ordinary, and its detection floor sits below the power-only one:

$$s^{n}_{\min}(\alpha) \approx z_{1-\alpha}\,\kappa, \qquad s^{\phi}_{\min}(\alpha) < s^{n}_{\min}(\alpha). \tag{11}$$

The first relation follows from the drift bound above; the second is an empirical finding of the demonstrator rather than a derived guarantee. When injection is strong ($s \geq 0.5$) the extra count is obvious and a single power threshold is already near-optimal, so CLARION adds little. Its advantage opens up in the moderate band, where the separating signal lives in the shape statistics (Fano factor, inter-arrival moments, the KS distance) that expose the damage injection does to the link's extinction ratio, the contrast between bright and dark symbols, which the mean power hides. Figure 3 quantifies this: the area-under-curve gap, widest in that moderate band, shows the fingerprint leading the power monitor across the entire ROC curve. The optical-plane data behind these results comes from the twin's link model, photon arrivals sampled for the nominal coherent state and for injections at each strength s, while the platform-plane data comes from the Basilisk vehicle model. In the demonstrator the plane is a small classifier ($14 \to 64 \to 32 \to 1$) paired with a fail-closed autoencoder ($14 \to 64 \to 32 \to 8 \to 32 \to 64 \to 14$), about 9,700 parameters in total at roughly 90 μs per window, well within an edge budget.

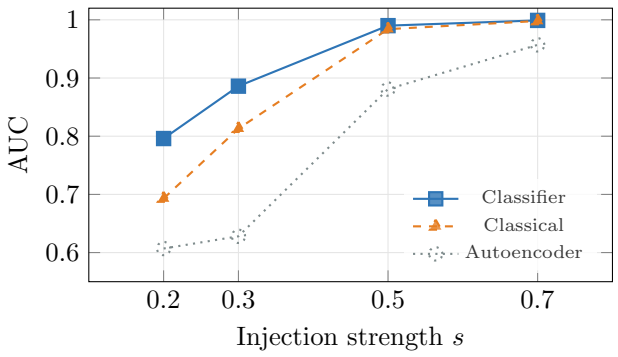

Figure 3: AUC vs. injection strength on the optical-link plane. CLARION's classifier advantage is widest at moderate strengths (+0.103 at $s = 0.2$, +0.073 at $s = 0.3$).

A.1.3 VITALS: The Platform-Health Plane

VITALS instantiates the same contract on the platform-telemetry bus. Its observation vector aggregates the subsystem channels a developing fault perturbs,

$$\mathbf{x}_t = \left[\,\text{SOC}_t,\ V_{\text{bus},t},\ \boldsymbol{\omega}_{\text{rw},t},\ \boldsymbol{\tau}_{\text{rw},t},\ \mathbf{b}_{\text{imu},t},\ \mathbf{T}_t\,\right], \tag{12}$$

with SOC the battery state-of-charge, V_{bus} the bus voltage, $\boldsymbol{\omega}_{\text{rw}}$ and $\boldsymbol{\tau}_{\text{rw}}$ the reaction-wheel speeds and commanded torques, $\mathbf{b}_{\text{imu}}$ the gyro bias estimates, and $\mathbf{T}$ the component temperatures, sampled on the dynamics tick with context $\mathbf{c}_t$. The plane scores a window by one-step prediction error: a windowed predictor f_ζ estimates the next sample from the recent window, and the residual

$$\mathbf{r}_t = \mathbf{x}_t - f_\zeta(\mathbf{x}_{t-\ell:t-1}, \mathbf{c}_t) \tag{13}$$

is scored by its squared Mahalanobis norm under a covariance Σ estimated on nominal residuals,

$$s_\theta(\mathbf{x}_t, \mathbf{c}_t) = \mathbf{r}_t^\top\, \Sigma^{-1} \mathbf{r}_t. \tag{14}$$

For $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ this equals $-\log \hat{p}_{\text{nom}}$ up to an additive constant, so VITALS takes its threshold $\tau^\star$ from the same false-alarm budget α as every other detector.

VITALS does not stop at the present sample; it also predicts forward, which is what lets it warn before a limit is reached rather than after. The plane learns a k-step-ahead forecast

$$\hat{p}_\zeta\left(\mathbf{x}_{t_0+k} \,\middle|\, \mathbf{x}_{t_0-\ell:t_0},\, \mathbf{c}_{t_0-\ell:t_0+k}\right), \tag{15}$$

over a horizon $k = 1, \ldots, K$ from a window of ℓ recent samples. Let $\mathcal{B} \subset \mathcal{X}$ be the set of unsafe states (charge under its floor, a reaction wheel at saturation, gyro bias past its bound). Integrating the forecast over $\mathcal{B}$ gives the probability of entering that set at horizon k,

$$\pi(t_0, k) \;=\; \int_{\mathcal{B}} \hat{p}_\zeta\left(\mathbf{x} \,\middle|\, \mathbf{x}_{t_0-\ell:t_0},\, \mathbf{c}_\cdot\right)\, d\mathbf{x}, \tag{16}$$

and the time-to-failure is the earliest horizon whose probability clears an actionability level β:

$$\text{TTF}(t_0) = \min\left\{k \in [1, K] : \pi(t_0, k) \geq \beta\right\}. \tag{17}$$

The lead time VITALS announces is therefore the point at which it becomes β-sure the state will leave the safe region. Figure 4 draws this for battery charge: the predicted distribution spreads as the horizon grows, and the warning fires once the mass beneath the low-charge limit first reaches β. Plain linear regression is the limiting case in which the forecast collapses to a single line: $\hat{p}_\zeta$ becomes a point mass at $y_0 + mk$ for current level y_0 and fitted trend m, $\beta = 1$, and the time-to-failure reduces to the crossing time $(y^\star - y_0)/m$ of the limit $y^\star$.

The validation harness exercises VITALS with these faults tuned to stay below their limits: a cell short that halves capacity while remaining above the low-voltage trip, a friction increase that never quite drives ω_{rw} across $\omega_{\max}$, and an IMU bias that creeps but never crosses the limit-checker's threshold, and confirms that the predictive detector flags each with lead time while a rule-based limit check stays silent.

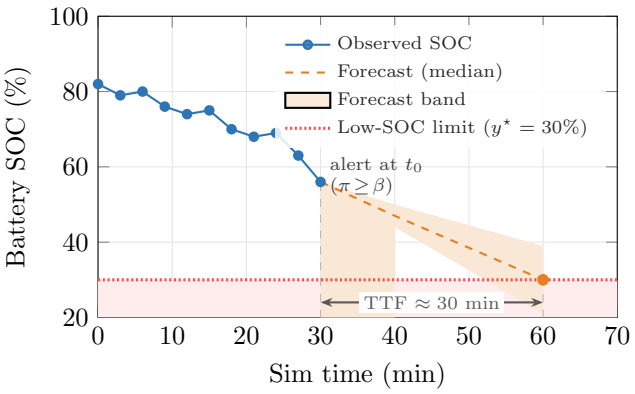

Figure 4: Predictive detection on the platform plane (schematic; values illustrate the mechanism rather than a recorded run). A learned forecast of battery state-of-charge (median, with its widening uncertainty band) projects the discharge trend ahead of the data. The detector fires at t_0, the lead time at which the forecast probability of dropping below the low-SOC limit first reaches β, giving a time-to-failure of ≈ 30 min before the limit is actually crossed.

A.2 CRUCIBLE: The Adversary-Aware Calibration Twin

The thresholds defined above require data that an operating spacecraft cannot supply. Setting $\tau^\star$ to a target false-alarm rate needs a distribution of nominal scores, and reporting a detector's sensitivity needs trajectories whose fault labels are known. Faults cannot be induced on a flight vehicle on demand, and waiting for them to occur yields too few labeled cases to calibrate against. CRUCIBLE produces both from a physics simulation carried on the vehicle itself.

The simulation is a full-dynamics model of the vehicle [31]. It integrates attitude as a rigid body driven by reaction wheels and gravity-gradient torque; translational motion under J_2 with atmospheric drag scaled by $F_{10.7}$; eclipse and sun-vector geometry; and visibility to the ground network. Faults enter through a typed registry (battery cell short, stuck reaction wheel, IMU bias, atmospheric-density anomaly), and each fault type is parameterized to match a documented component failure mode, so that a simulated fault corresponds to a physical one.

Sampling fault parameters uniformly would spend most of the budget on cases that are easy to classify. The scores under nominal and fault operation overlap only near the boundary between safe and failed trajectories, so that boundary is where calibration data is informative. CRUCIBLE concentrates its sampling there with a reversible-jump MCMC explorer [37]. Let θ_f collect the fault parameters (capacity lost to a shorted cell, reaction-wheel friction, gyro bias rate, atmospheric-density offset) and let $\mathcal{U}$ be the region of fault-parameter space whose runs are unsafe, with the safe/unsafe line set by a configurable trajectory test (charge under threshold inside a window, attitude error past limit during a pass, or conjunction range under margin). The explorer samples from a target that decays with distance to the region's boundary:

$$\rho(\theta_f) \;\propto\; p_0(\theta_f) \cdot \exp\Big(-\gamma \left(d(\theta_f, \partial\mathcal{U})\right)^2\Big). \qquad (18)$$

Here p_0 encodes prior belief about the fault parameters, taken from datasheets, fielded failure-rate data, and on-orbit incident reports; $d(\theta_f, \partial\mathcal{U})$ is the distance to that boundary, estimated from the simulated trajectory's margin against the safety test, so locating the boundary requires only running the twin, not knowing it in closed form; and γ sets how strongly the chain is pulled toward it. Because the kernel can add or remove faults, not only move their values, it also builds multi-fault scenarios that a one-fault-at-a-time sweep never visits.

Calibration uses the resulting trajectories directly. The false-alarm rate is the probability that a nominal observation exceeds the threshold,

$$\mathrm{FPR}(\tau) \;=\; \mathbb{P}_{\mathbf{x}\sim p_{\mathrm{nom}}}[s_\theta(\mathbf{x}, \mathbf{c}) > \tau]\,, \qquad (19)$$

and the true-positive rate is the same probability averaged over the labeled fault population,

$$\mathrm{TPR}(\tau) \;=\; \mathbb{E}_{\theta_f\sim\rho}[\mathbb{P}[s_\theta(\mathbf{x}, \mathbf{c}) > \tau \mid \theta_f]]\,. \qquad (20)$$

The operating point $\tau^\star$ is the threshold for which $\mathrm{FPR}(\tau^\star) = \alpha$ at the operator's budget, and the corresponding $\mathrm{TPR}(\tau^\star)$ is recorded as the detector's calibrated sensitivity. Each plane reports the pair (α, TPR) to ARBITER.

The same twin produces the forward forecasts ARBITER consumes during operation. One forecaster advances the present orbit under live $F_{10.7}$ drag to predict re-entry timing; a second works the orbital geometry to list the next acquisition and loss-of-signal windows for each ground station; and a third screens the Celestrak catalog with SGP4 [38], reporting any approach that closes within a set miss distance. These forecasters share the flight processor with the detectors and the reasoning agent.

A.3 ARBITER: Fusion and Attribution

ARBITER produces two outputs each tick. The first is an estimate of cause. It combines CLARION's optical-link evidence e_L, VITALS's platform-health evidence e_P, and CRUCIBLE's physical context into a distribution over the three causes the system distinguishes, $\{\mathrm{nominal}, \mathrm{fault}, \mathrm{attack}\}$. The second is a candidate response, described at the end of this subsection.

The fusion is valid because the two planes observe physically separate channels. Conditioned on a cause H, the optical evidence and the platform evidence are

independent, so ARBITER combines them as a product and normalizes over the three causes [39]:

$$P(H \mid e_L, e_P) = \frac{P(e_L \mid H)\, P(e_P \mid H)\, P(H)}{\sum_{H' \in \{\text{nom,fault,atk}\}} P(e_L \mid H')\, P(e_P \mid H')\, P(H')}, \tag{21}$$

The reported cause is the maximum-a-posteriori estimate, which reduces to three regimes:

$$\hat{H} = \arg\max_H P(H \mid e_L, e_P) = \begin{cases} \textbf{fault} & e_P \text{ high, } e_L \text{ low,} \\ \textbf{attack} & e_L \text{ high,} \\ \textbf{nominal} & e_L, e_P \text{ both low.} \end{cases} \tag{22}$$

A high platform score with a quiet optical link reads as a fault. A high optical score reads as an attack, because injection on the crosslink is not a byproduct of platform degradation. Low scores on both channels read as nominal. The fault and attack regimes are the pair that platform telemetry alone cannot separate; the optical channel is the term that distinguishes them, and any further evidence channel enters the same product as it is added. In the demonstrator the likelihoods come from CRUCIBLE's calibration: the labeled trajectory population fixes each plane's score distribution under nominal operation and under fault, so $P(e \mid H)$ is read off those calibrated distributions rather than assumed, and the priors $P(H)$ encode fleet incident rates and the operator's threat posture. Physical context needs no separate term in the posterior because the score models are conditioned on $\mathbf{c}$ throughout.

The second output converts the attribution into a proposed action. The policy is a small language-model agent running on the vehicle, on the order of 10^8–10^9 parameters after quantization. It is invoked when an alert fires or an attribution changes, not on every dynamics tick, and its seconds-scale latency sits outside the control loop: detection and fusion stay inside the tick, and the agent reasons on the compute margin discussed in §5.4. The agent is confined to a fixed, named tool set: query its physics twin, pull a neighbor's published status, command safe mode, take the payload offline, isolate a crosslink, and post or address a message on the peer bus. Write $s_t = (\mathbf{x}_t, \mathbf{c}_t, \mathbf{a}_t, \mathbf{m}_t)$ for the vehicle's telemetry $\mathbf{x}_t$, its context $\mathbf{c}_t$, the evidence $\mathbf{a}_t$ carried by recent alerts, and the contents $\mathbf{m}_t$ of the peer-message bus. The policy emits an action together with the reasoning trace that produced it:

$$\pi_\psi(a, \mathbf{z} \mid s) = \pi_\psi^{(z)}(\mathbf{z} \mid s) \cdot \pi_\psi^{(a)}(a \mid s, \mathbf{z}), \tag{23}$$

where $\mathbf{z} = (z_1, \ldots, z_n)$ is the logged chain the agent produced before settling on a, each entry z_k recording one evidence lookup or tool call and its returned result. Because the action is conditioned on this trace, whether the same action would have been taken without a given piece of evidence can be checked from the recorded trace.

A.4 AEGIS: Graduated Response and Containment

AEGIS maps an attribution to an action and enforces the limits on what the vehicle may do without operator approval. It selects one of three responses: execute a protective action immediately, refer a consequential action to the operator for approval with the supporting reasoning attached, or hold, taking no action. One consistency rule applies throughout. A protective action is recorded only when its tool call has executed, so if the agent's text claims an autonomous action such as `enter_safe_mode` with no matching successful `enter_safe_mode` call on record, the framework discards the claim before it reaches the operator.

Authority gating. The action space is partitioned by the authority each action requires:

$$\mathcal{A} = \mathcal{A}_{\text{auto}} \sqcup \mathcal{A}_{\text{propose}} \sqcup \{\texttt{hold}\}. \tag{24}$$

$\mathcal{A}_{\text{auto}}$ holds protective, reversible actions (`enter_safe_mode`, `disable_payload`, `quarantine_link`, `request_pass`); $\mathcal{A}_{\text{propose}}$ holds operationally consequential actions (an attitude-mode swap, a propulsive maneuver, a flight-software load); and `hold` is the choice to take no action. `quarantine_link` is the one auto action a telemetry-only system could not justify: it isolates a crosslink the optical plane has flagged, under the precondition given below.

The verifier. A protective action runs only if a rule system $\mathcal{R}$ entails it from the current reasoning trace $\mathbf{z}$ and state s:

$$V(a, \mathbf{z}, s) = \mathbb{1}[\, \mathcal{R}(\mathbf{z}, s) \models \texttt{authorized}(a) \,]. \tag{25}$$

Each auto action in the demonstrator has its own entry in $\mathcal{R}$. `enter_safe_mode` fires only when $\mathbf{z}$ holds a predictive alert that names this vehicle and ARBITER's current verdict is not **attack**; under an attack verdict the request routes to the operator instead, because a forced retreat into safe mode can be the very outcome an adversary is engineering. `request_pass` needs both an upcoming ground pass inside the horizon and a resource the vehicle is short on. `disable_payload` is admitted by a thermal warning, or by a charge warning whose TTF has dropped under the protective threshold. `quarantine_link` demands an attack verdict from ARBITER backed by a CLARION score e_L over threshold on the link in question. Whichever rule admits an action is recorded next to it, and that recorded rule is what explainability amounts to in practice: an operator can reconstruct, after the fact, exactly why the action was allowed.

Protective-action selection. Among the authorized auto actions, AEGIS executes the one that minimizes expected post-horizon cost while keeping the vehicle safe. With $\hat{p}_\zeta$ the forecast model of §A.1.3, rolled out by the onboard twin under each candidate action so that the expectation runs over action-conditioned trajectories, and

safe(s) $\in \{0, 1\}$ the per-vehicle safety predicate (charge in range, wheels under saturation, gyro bias within bound, and nothing in conjunction inside the miss-distance margin), the selected action solves

$$\begin{aligned} a_t^\star = \arg \min_{a \in \mathcal{A}_{\text{auto}}} \; & \mathbb{E}_{\hat{p}_\zeta}[\,\text{cost}(s_{t+K}) \,|\, s_t,\, a\,] \\ \text{s.t.} \quad & V(a, \mathbf{z}_t, s_t) = 1, \\ & \mathbb{P}_{\hat{p}_\zeta}[\,\text{safe}(s_{t+k}) = 1 \;\forall k \in [1, K] \,|\, s_t,\, a\,] \\ & \qquad \geq 1 - \epsilon, \end{aligned} \tag{26}$$

over a horizon K short enough to fit inside the next contact window. The first constraint restricts the search to actions the verifier authorizes; the second forbids trading mission utility for an unacceptable probability of a safety violation, with the tolerance ϵ set by the operator per fleet. The cost term measures utility lost (data not collected, battery discharged beyond plan, attitude margin spent). The feasible set is the authorized auto actions that also satisfy the safety bound; when it is empty, no protective action is admissible.

Dispatch. The agent's proposed action a_t selects the response tier, and the framework applies

$$\tilde{a}_t = \begin{cases} a_t^\star & a_t \in \mathcal{A}_{\text{auto}} \text{ and feasible}, \\ \texttt{request\_approval} & a_t \in \mathcal{A}_{\text{propose}}, \\ \texttt{hold} & \text{otherwise}. \end{cases} \tag{27}$$

An auto-tier proposal triggers the verified, safety-constrained action $a_t^\star$ from the selection above. A consequential proposal is packaged for operator approval rather than executed: the same optimization runs over it, and its output, the recommended action with its expected cost and safety probability, becomes the body of the request. Anything the verifier cannot authorize resolves to `hold`.

Outputs and coordination. Every executed or proposed action $\tilde{a}_t$ carries its reasoning trace, the verifier rule that cleared it, its cost, and its safety probability, all written when the action is decided rather than reconstructed later. That record is what an operator, an independent reviewer, or eventually a certifying authority uses to confirm an action was taken for stated reasons rather than as a model artifact. Across vehicles, AEGIS coordinates over a per-constellation message bus. When a vehicle cannot finish a scheduled task by itself, because it lacks the charge for an imaging pass, has lost its payload to safe mode, or has no downlink, it issues `request_pass`, broadcasting a `pass_request` that any able neighbor answers with a `pass_ack`. The bus enforces timeouts and per-topic cooldowns, and should no able peer reply within the tick, a deterministic fallback emits the `pass_ack` itself, so the handoff never depends on model behavior alone. On the ground, every $\mathcal{A}_{\text{propose}}$ action passes through an authorization gate: it enters a pending-command registry, the operator accepts or rejects it, and only an accepted action reaches the command queue.

A.5 Deployment

Every module has a definite place on the vehicle. CLARION runs in the receive-side signal-processing chain of each optical terminal, downstream of the coherent receiver (or an optional SPAD monitoring tap) and upstream of link management, so it sees the photon fingerprint before the link layer acts; the base capability needs no hardware change, mapping the 14 features into the terminal's quadrature domain in software. VITALS attaches to the telemetry stream the flight software already produces each tick. ARBITER, AEGIS, and the CRUCIBLE twin run on the flight processor (a LEON5-class radiation-tolerant single-board computer or a space-qualified FPGA); the detectors stay within the sub-10,000-parameter, $\sim$90 μs-per-window budget quoted above, while the reasoning agent claims most of the remaining compute margin and sets the floor on processor choice, the trade discussed in §5.4. The operator console and its authorization gate live on the ground, and the peer-coordination bus is per-constellation, both as described in the preceding subsection. The stack ships as an over-the-air module (CLARION as an optical-terminal firmware update, VITALS/ARBITER/AEGIS as a flight-software load), with per-link scores aggregating at a ground operations center for constellation-wide awareness.