The Silence Before the Update: Reconstructing a Decade of Cloud Incident Lifecycles Across Four Providers from the Public Record

Adepegba Akindayomi Akintade
School of Computer Science and Information Systems, Osiri University, Nebraska, USA
Corresponding author

Wasiu Abiodun Bello
Department of Decision and Information Sciences, Oakland University, Rochester, MI, USA

Harrison Oke Ekpobimi
School of Technology, Walsh College, Troy, MI, USA

ABSTRACT
Public cloud status pages are the only reliability record most customers ever see, yet providers publish them as rolling windows that quietly erase their own history. This paper shows that the erased record can be recovered at scale. Union-merging hundreds of Internet Archive captures of incident feeds, status APIs, and RSS histories, we reconstruct 4,181 unique incidents across Google Cloud, Cloudflare, DigitalOcean, and GitHub, spanning 2013 to 2026 and released with full provenance, and the Google Cloud reconstruction carries a verifiable completeness bound for 2019 to 2024. Two patterns hold at every provider. Incident durations follow a lognormal law, and onsets follow the working week, with weekday incident rates 2.5 to 6.2 times weekend rates, the public signature of change driven failure. Other patterns diverge instructively. Google Cloud's median resolution time roughly doubled after 2021 (Cox era hazard ratio 0.65, $p < 0.001$) while Cloudflare's fell and two providers held flat, so the slowdown is provider specific rather than an industry constant, and Google's severity triage is inverted in the tail, with the longest incidents graded least severe. Most operationally, the delay before Google's first status update predicts eventual duration within every severity stratum, and a gradient-boosting model using only first-update information identifies long-tail incidents on a strictly temporal holdout with AUC 0.707 at more than twice base-rate precision. The signal vanishes at GitHub, where the record shows no measurable acknowledgment delay, a pattern consistent with a shared-cause mechanism. Archived status feeds become longitudinal science, and their silences become signals.

General Terms
Measurement, Reliability, Performance, Experimentation.

Keywords
Cloud reliability, incident management, status pages, web archives, survival analysis, AIOps.

1. INTRODUCTION
Public cloud platforms now carry workloads whose failure is consequential far beyond the technology sector, from payment rails and airline operations to hospital systems and government services. When a hyperscale platform degrades, the primary channel through which millions of dependent organisations learn what is happening is a status page, a provider-operated feed of incident declarations, progress updates, and resolution notices. These feeds have become the de facto public record of cloud reliability. Yet they are a record with a peculiar property. They are written for the present, not the past. Providers publish incidents while they matter and quietly retire them afterwards, so the archive that would allow anyone to ask how reliability is changing over the years does not, in any official sense, exist.

This paper starts from the observation that the record is not actually lost. The Internet Archive has been capturing providers' machine-readable incident feeds, their status APIs, and their RSS history feeds since the mid 2010s, and each capture preserves the window of history the source exposed on that day. Because successive windows overlap, the captures can be stitched together. Applying this archive-union idea through three distinct channels, we reconstruct 4,181 unique incidents across four providers. These comprise 1,602 Google Cloud incidents spanning October 2013 to September 2026 with 7,174 timestamped status updates, 1,145 Cloudflare incidents from 2014 to 2026, 961 DigitalOcean incidents from 2017 to 2026, and 473 GitHub incidents from 2019 to 2026, all released with full capture-level provenance. To our knowledge this is the largest and longest longitudinal incident corpus ever assembled from public sources, and Google Cloud, whose reconstruction carries a verifiable completeness bound over 2019 to 2024, forms its deep spine.

The reconstruction is not an end in itself. It enables three questions that could not previously be answered with public data. First, how has the lifecycle of a cloud incident changed over a decade? We find that it has lengthened. The median time to resolution for the pre-2021 era, 1.82 hours, roughly doubled to 3.20 hours for 2021 to 2024, and the 90th percentile grew from about seven hours to a 2022 peak above fifty, a shift that survives winsorisation, exclusion of the pandemic year, and a Cox proportional-hazards specification in which the post-2021 era carries a resolution hazard ratio of 0.65 ($p < 0.001$) after severity and scope are controlled.

Second, does the provider's own severity triage behave the way service-level thinking assumes? It does not. High-severity incidents resolve fastest, with a median of 2.25 hours against 3.03 for low-severity events, and the heaviest resolution tail belongs to the incidents graded least severe. Prioritisation evidently works, but its cost is concentrated in a long tail of low-priority degradations that can linger for days, precisely the events that service-level agreements weigh least and that dependent operators find hardest to plan around.

Third, and most useful operationally, can the eventual character of an incident be anticipated from what the provider does in its opening minutes? Here the record yields a signal we did not expect to be so strong. The delay before the first public status update is the single most informative early feature of eventual incident duration. Incidents whose first update arrives within fifteen minutes go on to resolve in a median of 1.3 hours. Incidents preceded by four to ten hours of silence resolve in a median of 8.1 hours, a six-fold gradient that holds within every

severity stratum (Spearman ρ from 0.21 to 0.51, all $p < 0.001$). A gradient-boosting model using only features observable at the moment of the first update predicts membership in the long tail decile of durations with an area under the ROC curve of 0.707 on a strictly temporal 2023 to 2024 holdout, at more than twice the precision of the base rate. The silence before the update, it turns out, is itself a measurement.

The contribution of the paper is therefore fourfold. It introduces and validates an archive-union method, in three source-specific variants, that turns ephemeral status surfaces into longitudinal research data and releases the resulting four-provider corpus. It provides the first decade-scale characterisation of hyperscale incident lifecycles from public evidence, including survival modelling of resolution and a documented inversion between graded severity and resolution tail risk. It demonstrates a practical early-triage predictor that dependent operators can compute from information the provider has already published. And it tests every headline pattern across three further providers, finding that some replicate at every provider we test, the lognormal shape of durations and the working-week rhythm of onsets, while others, notably the post-2021 slowdown, are provider-specific, a divergence with direct consequences for how industry availability claims should be read. Throughout, we are explicit that the corpus measures the public reporting of incidents rather than their private engineering reality, and we treat changes in reporting practice, including the visible thinning of the public feed after 2024, as findings about transparency rather than noise.

2. RELATED WORK

Empirical failure studies of large systems have a long lineage. Gray's analysis of Tandem outage reports established the tradition of learning from operational failure records [1], and Oppenheimer, Ganapathi, and Patterson carried it to early Internet services, showing that operator error and front-end software dominated user-visible failures [2]. The taxonomy of Avizienis and colleagues gave the field its shared vocabulary of faults, errors, and failures [3]. For the cloud era, the touchstone is the study by Gunawi and colleagues of 597 unplanned outages reconstructed from headline news and public post-mortems across 32 services between 2009 and 2015 [4]. Our work differs from that lineage in its source and its span. Rather than sampling secondary press coverage across many providers over seven years, we reconstruct the primary machine-readable record of four providers over thirteen years, anchored by a hyperscale feed deep enough for a completeness analysis, which trades survey breadth for internal consistency, complete update timelines, and provider-assigned severity.

A second body of work studies incident management from inside the provider. Microsoft researchers have characterised incident triage on twenty large services [5], articulated the agenda of intelligent incident management [6], and analysed how production incidents are actually fought and mitigated [7]. These studies enjoy telemetry that no outsider can see, but for exactly that reason they cannot be reproduced, audited, or extended by the community, and they describe the provider's view rather than the dependent customers. Our corpus is the complement. It contains only what customers were told, which is the information set on which customers must actually act. The AIOps literature surveyed by Notaro, Cardoso, and Gerndt [8] pursues automated failure management from internal signals. Our early-triage model can be read as AIOps for the consumer side; built from the only telemetry the consumer has.

A third tradition asks where failures come from and how they behave inside production systems. Classical measurement studies established empirical baselines, from two decades of HPC failure records whose repair times already followed heavy-tailed lognormal laws [9], through hardware reliability at datacenter scale [10] and correlated unavailability in Google's global storage fleet [11], to log mining as an operational instrument [12]. A more recent line dissects failure genesis, covering gray failures that degrade services while evading detectors [13], metastable states that keep systems trapped in overload after the trigger has passed [14], network partitions with catastrophic downstream effects [15], and the bug populations behind severe Azure incidents [16], with Yuan and colleagues showing that most catastrophic failures trace to trivially testable error-handling lapses [17]. Provider-side prediction has been attempted from internal alert streams [18], and practitioner doctrine treats postmortem discipline as the engine of reliability learning [19, 20]. Our contribution stands apart from all of these in its vantage point. Each of those studies observes the system from within. We observe the only surface the customer ever sees, over a longer span than any of them, and show that even that thin public surface carries exploitable structure.

Closest to our data source, Chu and colleagues characterised outages of eight public large-language-model services from their status pages over roughly two years [21]. Their study demonstrates the scientific value of status-feed data but inherits the feeds' short memory. The archive-union method removes that ceiling, extending coverage by an order of magnitude in time, and our analysis adds survival modelling, severity inversion evidence, and a validated predictive layer that characterisation studies have not attempted. Industry trackers such as the Uptime Institute's annual outage analyses [22] and the Parametrix outage risk reports [23] count and cost incidents across providers, but their methods and data are proprietary, their windows are short, and their claims are not peer reviewed. Where our decade-scale evidence bears on their headline assertions, we test them. Finally, our use of the Wayback Machine as a scientific instrument follows the methodological turn described by Ogden, Summers, and Walker [24], and our reliability framing draws on the meaningful-availability argument of Hauer and colleagues that metrics should capture what users experience [25]. Recent work has begun to mine incident reports with language models, extracting structured fields from unstructured post-mortems [26] and studying incidents in generative-AI services [27], while empirical studies of incident response practice [28], microservice failure diagnosis [29], provider behaviour under pandemic load [30], and dependency-aware linking of related incidents [31] round out a fast-growing literature that relies, almost without exception, on either private telemetry or short public windows. The present study extends prior work on operational readiness and dependency-aware recovery in cloud systems [32, 33] with the missing longitudinal, evidence-from-the-record layer.

3. RECONSTRUCTING THE PUBLIC INCIDENT RECORD

3.1 Source feed and its ephemerality

Google Cloud publishes service-health information at status.cloud.google.com, including a machine-readable feed, incidents.json, that lists incident objects with begin, end, created, and modified timestamps, a provider-assigned severity (low, medium, high), a status-impact label, the affected service or an explicit multiple-products marker, a list of affected products in the newer schema, and an ordered sequence of textual status updates each carrying its own timestamp. The feed is authoritative but ephemeral. It exposes a rolling window whose depth has varied over the years from several years of history in the mid 2010s to roughly ninety days in 2025, after

which incidents disappear from public view. The window depth is itself part of our findings, because it measures how much history the provider is willing to keep public.

3.2 Archive-union reconstruction

The Internet Archive's Wayback Machine has captured incidents.json repeatedly since 2015. We enumerated all captures through the archive's CDX index, obtaining 338 captures with HTTP status 200, of which 280 were distinct by content digest. We retrieved and successfully parsed 255 capture sources, namely 251 archived captures, three era anchor captures from the same distinct set handled separately for the largest windows of 2021, 2023, and 2025, and the live feed at reconstruction time. Twenty-six sparse-era captures remained unretrievable after repeated attempts, and the completeness analysis of Section 3.3 shows that the retrieved set already saturates the feed's retention window throughout the years on which our trend claims rest. Each capture contributes the incident objects visible in its window, and because consecutive windows overlap substantially their union is continuous. Incidents were merged on the provider's stable incident identifier, with a composite service-and-number key for the legacy schema used before identifiers were introduced. Where the same incident appeared in multiple captures we retained the version with the longest update sequence, which is by construction the most complete lifecycle record, and we recorded the contributing capture for every incident in a provenance file released with the corpus.

The merged corpus contains 1,602 unique incidents beginning between 7 October 2013 and 1 September 2026, each with a begin and end timestamp and a severity grade, and 989 with affected-product listings from the newer schema. Together they carry 7,174 timestamped status updates. Validation proceeded in three directions. Every incident present in the live feed at reconstruction time was also recovered from the archive union, confirming that the method loses nothing the provider still publishes. Spot comparison of overlapping captures showed byte-identical incident objects across captures, confirming that archived windows are faithful copies rather than re-renderings. And yearly incident counts track the capture density and window depth in the expected way, with two regime boundaries discussed next.

3.3 Completeness of the reconstruction

The obvious objection to any archive-union corpus is sampling. If captures were taken too rarely, incidents could have entered and left the feed unseen, and if visibility depended on incident properties, the corpus would be biased. Both concerns can be tested from the captures themselves. For every capture we measured the window depth, the distance from the capture time back to the earliest incident it lists, and for every year we compared the longest gap between consecutive retrieved captures against the shallowest window observed that year. An incident can escape the corpus only if a capture gap exceeds the retention window. For every year from 2019 through 2024, the era covered by the completeness bound, the maximum capture gap (between 14 and 226 days, depending on the year) never exceeded the minimum window depth (286 to 361 days), so the corpus provably contains every incident the provider published in those years. The only at-risk intervals fall in 2016 and 2018. We retain them, and excluding them leaves the era contrast intact, with medians of 2.00 and 3.20 hours and p below 0.001. The window is defined in calendar time rather than as a count of recent incidents, which matters for bias. A time-based window keeps an incident visible for a fixed period after its end regardless of how long the incident lasted, so even in sparse capture eras the visibility of an incident does not depend on its duration, and the duration distributions we analyse are unbiased by the capture process.

3.4 Reporting regimes and the thinning of the record

Two discontinuities in the yearly series reflect the provider's reporting practice rather than underlying reliability. Counts step upward from 2019 to 2020, when the feed's schema was enriched and per-product incident pages were consolidated, and they collapse after 2024, when the public feed narrowed to a short recent window and, on its face, to fewer and more consequential entries, with only 22 incidents visible for 2025 against 351 for 2023, a count shaped by capture cadence as well as provider retention policy. We therefore confine all trend analyses to the 2014 to 2024 window in which reporting practice was comparatively stable, treat the 2013 fragment and the post-2024 era as descriptive context, and read the post-2024 thinning itself as evidence. Over the very years in which our measured resolution times lengthened, the provider also reduced how much of its incident history remains publicly inspectable. A transparency claim of this kind can only be made because the archive preserves what the feed no longer shows.

3.5 Extending the reconstruction to three further providers

The same archive-union idea applies to any provider whose status surface has been captured, though the richest channel differs by provider. For GitHub, the Wayback Machine holds 37 content-distinct captures of the statuspage API endpoint, each listing the fifty most recent incidents as structured JSON with backdated start, creation, and resolution timestamps and an impact grade. Their union with the live feed yields 512 unique incidents from 2018 to 2026, of which 473 have positive, plausible durations. For Cloudflare and DigitalOcean the API endpoint is sparsely captured, but the RSS history feed is not. The two feeds yield 99 and 81 distinct captures respectively, spanning 2014 to 2026, each carrying recent incidents with their complete update timelines as timestamped text. Parsing both the legacy and the current RSS markup, filtering scheduled maintenance, and merging on incident identifiers yields 1,145 usable Cloudflare incidents and 961 DigitalOcean incidents. These three reconstructions do not enjoy Google Cloud's formal completeness bound, since RSS windows are count-based and capture cadence varies, so we use them for replication of distributional and structural findings rather than for volume trends, and we disclose their year coverage in the released provenance. Durations for the RSS reconstructed providers run from first to last update of each incident, a definition that differs from Google's backdated begin-to-end interval by the length of any unacknowledged silence, a difference that Section 5.8 turns from a nuisance into a finding.

4. METHODS

4.1 Measures

Incident duration is the elapsed time from the provider's begin to end timestamp, in hours. We excluded records with non-positive durations and durations above ninety days, which inspection showed to be bookkeeping artefacts. These filters removed nothing beyond entries already dropped at parse time, and all 1,602 merged incidents enter the reported analyses. First-update delay is the elapsed time from incident begin to the timestamp of the earliest status update, in minutes, available for 1,456 incidents after removing negative and extreme values beyond ten hours. The gradient bins used in the silence analysis are left-open, so delays of exactly zero minutes fall outside the

binned display. Scope is measured by the affected-product count and by a binary marker for incidents the provider labelled as spanning multiple products. Severity uses the provider's three-level grade, and impact uses the provider's status-impact label. Calendar features include start hour and a weekend indicator. Update cadence is summarised by the number of updates and the median gap in minutes between consecutive updates for incidents with at least three updates. Interarrival times between successive incident onsets are computed within the 2019 to 2024 complete-coverage era, and onset periodicity over days of the week is tested by $\chi 2$ goodness of fit against uniformity. Candidate duration distributions, lognormal, Weibull, and exponential, are compared by the Akaike information criterion. Concurrency is measured by sweeping incident intervals and counting, for each onset, how many earlier incidents remain open. Cause signatures are mined from the concatenated update texts of each incident by keyword families for networking, deployment and rollout, configuration change, capacity and quota, rollback, facility and hardware, and third-party dependencies, a deliberately transparent scheme whose limitations Section 6.2 discusses. Era contrasts are also summarised by restricted mean survival time at a horizon of 48 hours, and the proportional-hazards assumption of the Cox model was checked and not rejected. Headline inferences rest on contrasts significantly below 0.001, which survive any standard multiplicity adjustment across the roughly two dozen tests reported, and marginal coefficients are read descriptively.

4.2 Statistical analysis

Duration distributions are summarised by medians and 90th percentiles with bootstrap confidence intervals, since means are dominated by the extreme tail. Era contrasts use the Mann-Whitney test [34] on durations winsorised at the 99th percentile, with rank-biserial effect sizes. Resolution is modelled as a survival process. Kaplan-Meier estimators [35] describe the probability that an incident remains unresolved as a function of elapsed time, stratified by severity and era, and a Cox proportional-hazards model [36] estimates the association of severity, multi-product scope, weekend onset, and era with the hazard of resolution. All incidents in the corpus reach resolution, so censoring is not required. The survival machinery is used for its distributional honesty about heavy tails rather than for censoring adjustment. Analyses were implemented in Python with lifelines [37] and scikit-learn [38].

4.3 Early-triage prediction protocol

The predictive task asks whether an incident will fall in the long tail of durations, defined as exceeding the 90th percentile of the training era, 15.7 hours, using only information available at the moment the first status update is published. These inputs are severity, impact label, multi-product scope, affected-product count, start hour, weekend indicator, first-update delay, and the word count of the first update text. Records without an affected products listing carry a product count of zero and missing first update delays are imputed at the sample median. The split is strictly temporal, training in 2014 to 2022 (1,051 incidents) and testing on 2023 to 2024 (515 incidents), so that evaluation reflects genuine prospective use under a shifted base rate, which rose from 0.10 in training to 0.18 in the test years. We compare a class-weighted logistic regression with a gradient boosting classifier [39] and report area under the ROC curve, average precision against the shifted base rate, and the Brier score for calibration quality. Feature importance is taken from the boosting model. As robustness checks we re-estimated the silence association nonparametrically within each severity stratum, re-ran era contrasts excluding 2020, and verified that results are unchanged under winsorisation. Cross-provider replication applies a fixed battery to each provider, consisting of distribution fitting by AIC, the pre-2021 against post-2021 era contrast by Mann-Whitney, weekday against weekend onset rates, and the silence gradient wherever recorded start times make acknowledgment delay measurable.

5. RESULTS

Findings are presented as fourteen numbered observations, O1 to O14, each tied to the analyses of Section 4.

5.1 A decade of incident lifecycles

Figure 1 shows the reconstructed corpus against the archive captures that produced it, and Table 1 summarises durations by year. Two movements dominate the stable 2014 to 2024 window. Incident volume grows with the platform and with reporting enrichment, from 55 published incidents in 2014 to a peak of 351 in 2023. Resolution, meanwhile, slows. The yearly median sits below 1.7 hours in two of the three years from 2018 to 2020, then holds between 2.9 and 3.9 hours in every year from 2021 to 2024, and the 90th percentile moves from roughly 7 hours in 2018 to between 37 and 57 hours in 2022 through 2024 (Figure 2). The era contrast is decisive. Winsorised durations for 2014 to 2020 against 2021 to 2024 differ with $p < 0.001$ by the Mann-Whitney test, medians 1.82 against 3.20 hours, rank-biserial 0.22, and the contrast is materially unchanged when 2020 is excluded to remove any pandemic effect. Restricted mean survival time at a 48-hour horizon doubles from 4.60 to 9.45 hours across the same eras, and the share of published incidents lasting beyond a full day quadruples from 3.2 to 13.3 percent. Observation O1: the publicly visible time from declared begin to declare end of an incident has roughly doubled since 2021, and the long tail has widened several-fold. Observation O2: measured against a conventional 99.9 percent monthly availability budget of 43.8 minutes, 86.1 percent of published incidents would, on their own, exhaust the month's budget for a fully affected service.

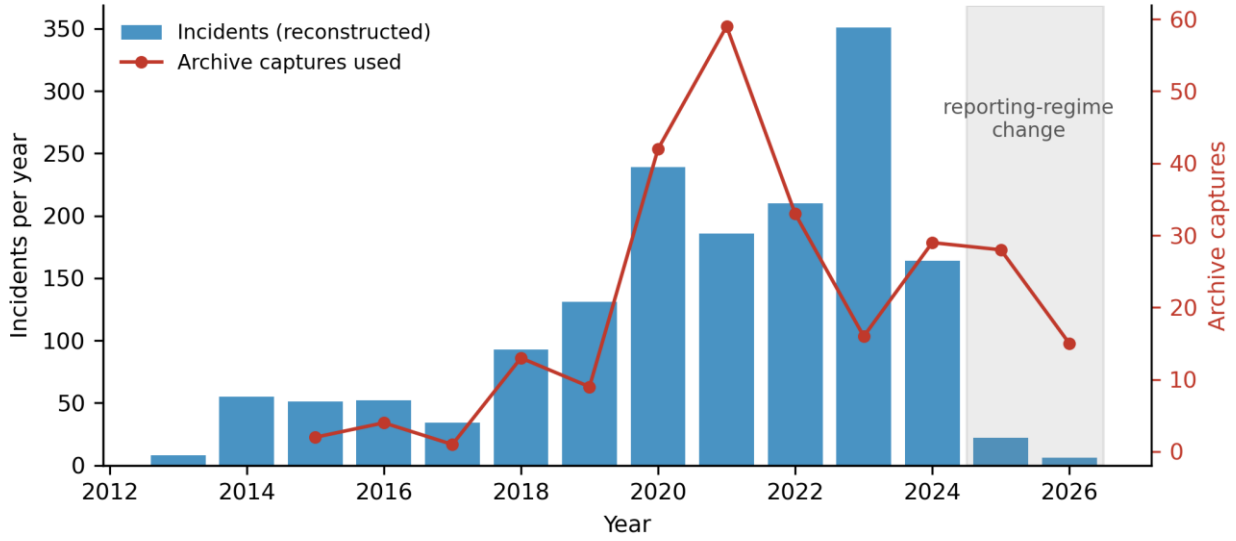

Figure 1. The reconstructed corpus. Bars: unique incidents per year recovered by the archive union. Line: distinct Internet Archive captures contributing to each year. The shaded region marks the post-2024 reporting regime in which the public feed retains only a short window.

Table 1. Published incident volume and resolution times by year, 2014 to 2025. Medians and 90th percentiles in hours. Post-2024 rows reflect the thinned reporting regime and are excluded from trend tests.

Year	Incidents	Median (h)	P90 (h)
2014	55	3.30	12.08
2015	51	1.70	7.63
2016	52	1.82	16.97
2017	34	1.80	18.24
2018	93	1.16	6.85
2019	131	2.79	14.99
2020	239	1.64	10.37
2021	186	2.95	25.62
2022	210	3.08	56.53
2023	351	2.94	38.42
2024	164	3.91	36.71
2025	22	5.18	22.05

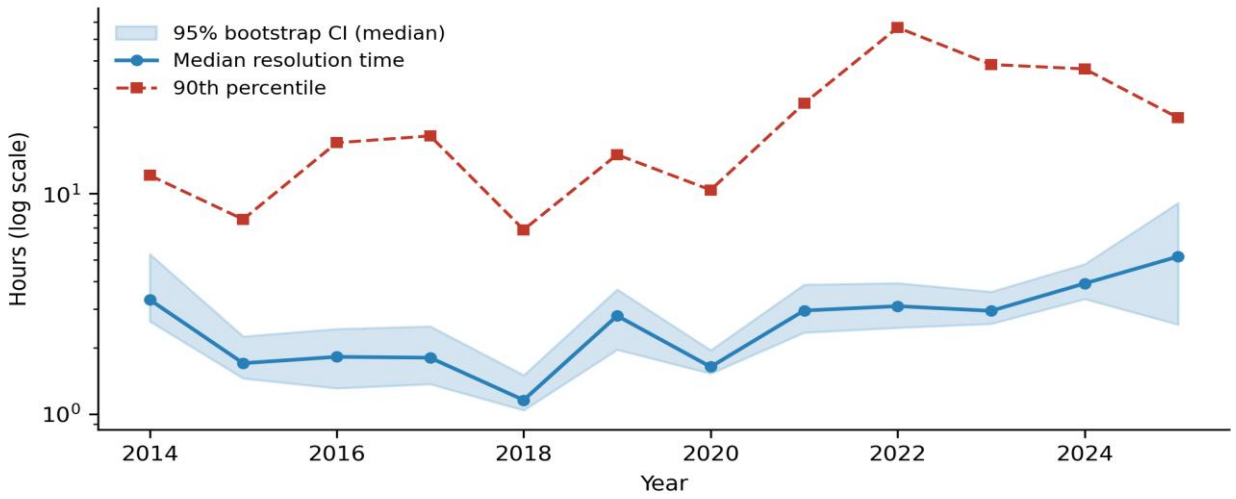

Figure 2. Yearly median resolution time with 95% bootstrap confidence band and 90th percentile, 2014 to 2025, log scale. The median roughly doubles after 2021 and the tail widens by a factor of five or more.

Two readings of the slowdown deserve statement, because the record cannot fully separate them. Incidents may genuinely take longer to resolve, plausibly because the platform's internal dependency depth has grown, consistent with the general argument that recovered components do not imply recovered services. Alternatively, the provider may have raised the bar for declaring an incident at all, so that the published population shifted toward harder events, a mechanism consistent with the post-2024 thinning of the feed. Both readings matter to dependent operators. The first says the platform heals more slowly, and the second says the public record is becoming a curated sample of it. Either way, planning assumptions calibrated on pre-2021 behaviour is stale, and the completeness bound of Section 3.3 rules out the third explanation for the 2019 to 2024 arm, and the time-based retention argument of Section 3.4 extends that protection to earlier years under a stationarity assumption.

5.2 The rhythm of failure

Within the complete-coverage era the platform published a new incident every 9.6 hours at the median, with a heavy-tailed interarrival distribution whose mean of 40.8 hours reflects quiet stretches between bursts. Onsets are anything but uniform in time (Figure 3, left). Thursday alone accounts for 287 onsets against 27 on Sunday, weekdays outnumber weekends by roughly six to one per day, and uniformity is rejected overwhelmingly ($\chi 2$, $p < 0.001$). Observation O3: published incident onset follows the working week, the signature of change-induced failure, since the dominant proximate trigger of cloud incidents is human activity, deployments, configuration pushes, and rollouts, which happen when engineers are at their desks. This echoes, from a fully public record, what internal studies have long reported about change as the leading incident trigger [4, 7]. Durations themselves are well approximated by a lognormal law (Figure 3, right). The lognormal fit dominates Weibull and exponential alternatives by more than 490 AIC points, with a log-scale sigma of 1.57, though with 1,566 observations the Kolmogorov-Smirnov test still formally rejects an exact fit, as happens whenever an idealised parametric family meets real operational data in this quantity. Observation O4: incident durations are approximately lognormal with heavy right skew, matching the repair-time laws long observed in HPC failure records [9], so means are misleading and median-and-tail reporting of the kind used throughout this paper is the appropriate summary.

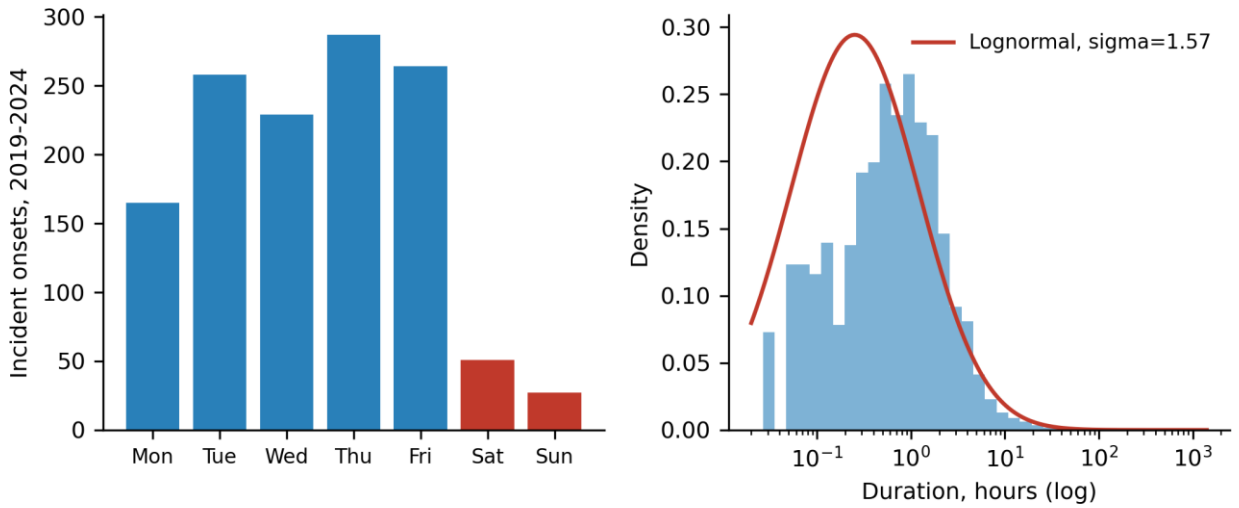

Figure 3. The rhythm of failure. Left: incident onsets by day of week over the 2019 to 2024 window; weekend days in red. Right: duration distribution over the 2014 to 2024 trend window with fitted lognormal density, log scale.

5.3 The severity inversion

Provider-graded severity orders incidents by impact, and one might expect it also to order them by resolution difficulty. The data show the opposite ordering of tails. High-severity incidents, 291 in all, resolve with a median of 2.25 hours and a 90th percentile of 10.75. Medium-severity incidents sit at 2.36 and 16.79, and low-severity incidents, 516 of them, at 3.03 and 44.65 (Table 2, Figure 4 left). In the Cox model (Table 3), whose proportional-hazards assumption was checked and not rejected, high severity carries a resolution hazard ratio of 1.18 relative to low ($p = 0.053$), while the post-2021 era carries 0.65 ($p < 0.001$), and scope and weekend onset show no reliable association once severity and era are held. Observation O5: severity triage is inverted in the tail, read as a description of the tail rather than a mean effect, since the Cox high-severity coefficient is marginal at $p = 0.053$. Severe incidents attract engineering attention and resolve comparatively fast, while the longest-lived incidents are those graded least severe, which service-credit regimes keyed to severity grades [40] are structurally blind to.

Table 2. Resolution time by provider-assigned severity, full corpus.

Severity	n	Median (h)	P90 (h)	Max (h)
High	291	2.25	10.75	56.5
Medium	795	2.36	16.79	1410.9
Low	516	3.03	44.65	743.1

Table 3. Cox proportional-hazards model of resolution, 2014 to 2024 (n = 1,566). Hazard ratios above 1 indicate faster resolution. The proportional-hazards assumption was checked and not rejected.

Covariate	Hazard ratio	95% CI	p
High severity (vs low)	1.18	1.00-1.40	0.053
Medium severity (vs low)	1.03	0.91-1.16	0.685
Multi-product scope	1.07	0.93-1.24	0.344
Weekend onset	1.00	0.82-1.23	0.970
Era 2021-2024 (vs 2014-2020)	0.65	0.57-0.74	< 0.001

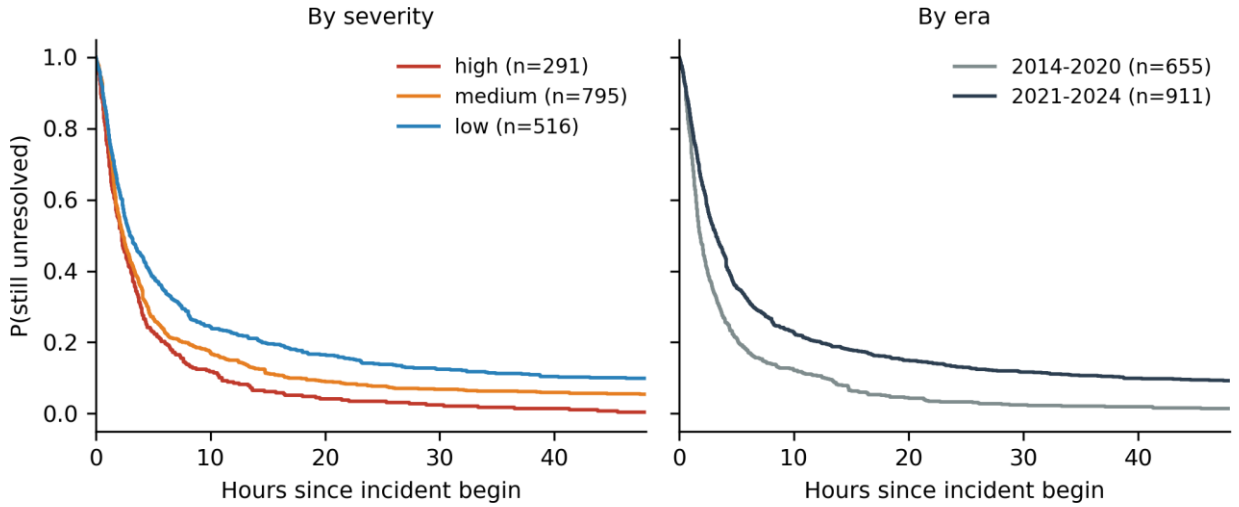

Figure 4. Kaplan-Meier curves for the probability that an incident remains unresolved. Left: by provider severity; the low-severity stratum owns the heaviest tail. Right: by era; the post-2021 curve lies above the earlier curve at every horizon.

5.4 Concurrency and where the burden falls

The platform's incidents are not isolated episodes. Within the complete-coverage era, 52.5 percent of incidents began while at least one earlier-opened incident was still open, and the maximum observed concurrency reached 34 simultaneously open incidents, an environment in which gray and metastable failure modes that evade clean detection [13, 14] become especially plausible. Observation O6: for a dependent operator, the modal experience of a Google Cloud incident is not a lone event but an arrival into an already degraded environment, which strengthens the case for customer-side recovery orchestration that reasons about multiple simultaneous faults. Load is also concentrated. Compute Engine, Cloud Networking, App Engine, BigQuery, and Kubernetes Engine together account for 41.6 percent of incidents attributed to a single named product, and explicit multi-product incidents, 280 in all or 17.5 percent of the corpus, are harder events, with a median duration of 3.31 hours against 2.35 for single-product incidents (Mann-Whitney $p < 0.001$) and a heavier 90th percentile.

5.5 What the updates say: cause signatures

The 7,174 update texts allow a transparent, keyword-based reading of what providers say incidents involve (Figure 5, Table 4). Among the 1,560 trend-era incidents with update text, 904 matches at least one of seven cause-signature families. Networking language dominates, appearing in 659 incidents, followed by deployment and rollout language in 317 and configuration change in 233, a public-record echo of the change-driven onset rhythm of O3. Duration separates the families. Incidents whose updates mention rolling back run a median of 4.38 hours, and the small third-party and upstream family runs longest at 4.50, against 2.35 for networking incidents generally. Observation O7: networking is the modal vocabulary of Google Cloud incidents, but the expensive vocabulary is remediation by rollback, which is consistent with rollback being the fix of last resort reached only after faster mitigations fail. Observation O8: cause language predicts duration coarsely but usefully, and a customer parsing updates in real time gains signal beyond the metadata fields.

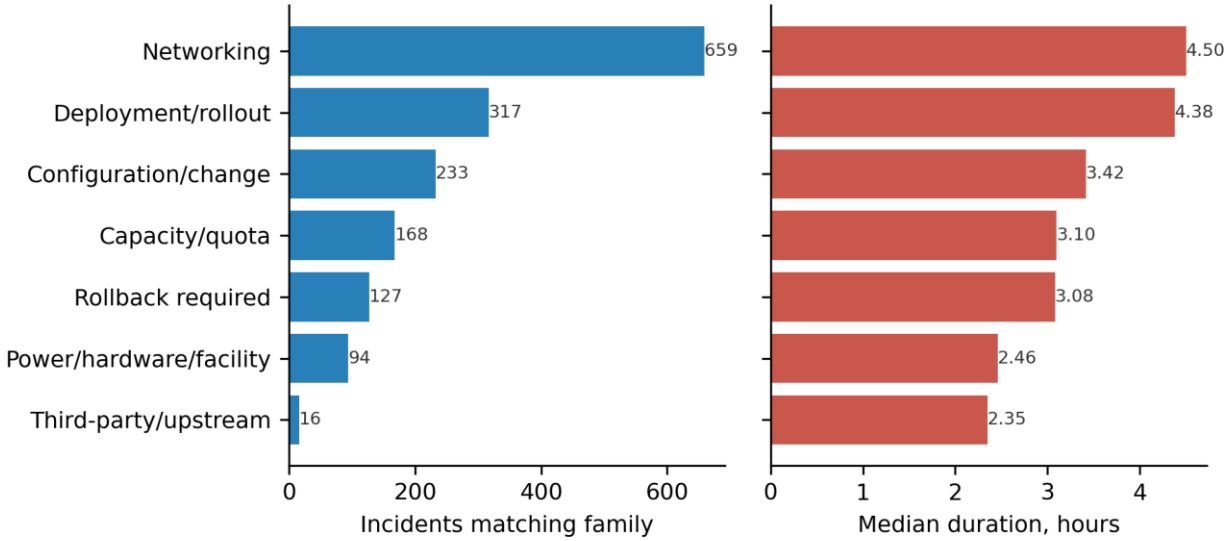

Figure 5. Cause-signature families mined from update texts. Left: incidents matching each family. Right: median duration per family; rollback and third-party signatures mark the longest incidents.

Table 4. Cause-signature families mined from update texts, 2014 to 2024. An incident may match several families.

Signature family	Incidents	Median duration (h)
Networking	659	2.35
Deployment/rollout	317	3.42
Configuration/change	233	3.08
Capacity/quota	168	3.10
Rollback required	127	4.38
Power/hardware/facility	94	2.46
Third-party/upstream	16	4.50

5.6 The silence before the update

The first thing a provider does in an incident is decide when to say something. That decision turns out to forecast the incident's future. Across 1,456 incidents with a measurable first-update delay, the correlation between the delay and the eventual duration is $\rho = 0.30$ ($p < 0.001$), and the binned gradient is monotone and steep. Incidents first acknowledged within fifteen minutes resolve in a median of 1.33 hours, those acknowledged between one and two hours in about 2.1, those after two to four hours in 4.1, and those preceded by four to ten hours of silence in 8.07 hours, with the 90th percentile in the longest-silence bin exceeding 22 hours (Figure 6). Crucially, the association is not an artefact of severity mix. Within high severity incidents alone the correlation strengthens to $\rho = 0.51$, within medium to 0.43, and within low it remains positive at 0.21, all $p < 0.001$. Observation O9: the delay before the first public status update is a graded, severity-independent leading indicator of incident duration. Long silence is not merely a property of unimportant events. Among the most severe incidents it is the strongest marker of a hard road ahead, and the plausible mechanism is shared cause rather than communication policy. An incident that is hard to diagnose is slow both to explain and to fix, so the delay in explanation leaks information about the difficulty of the fix. The cadence within an incident carries the same signal. Among 1,134 incidents with at least three updates, the median gap between consecutive updates correlates with eventual duration at $\rho = 0.49$ ($p < 0.001$), and the median updates per incident rose from three to four across the eras, so the provider now talks more per incident even as incidents last longer.

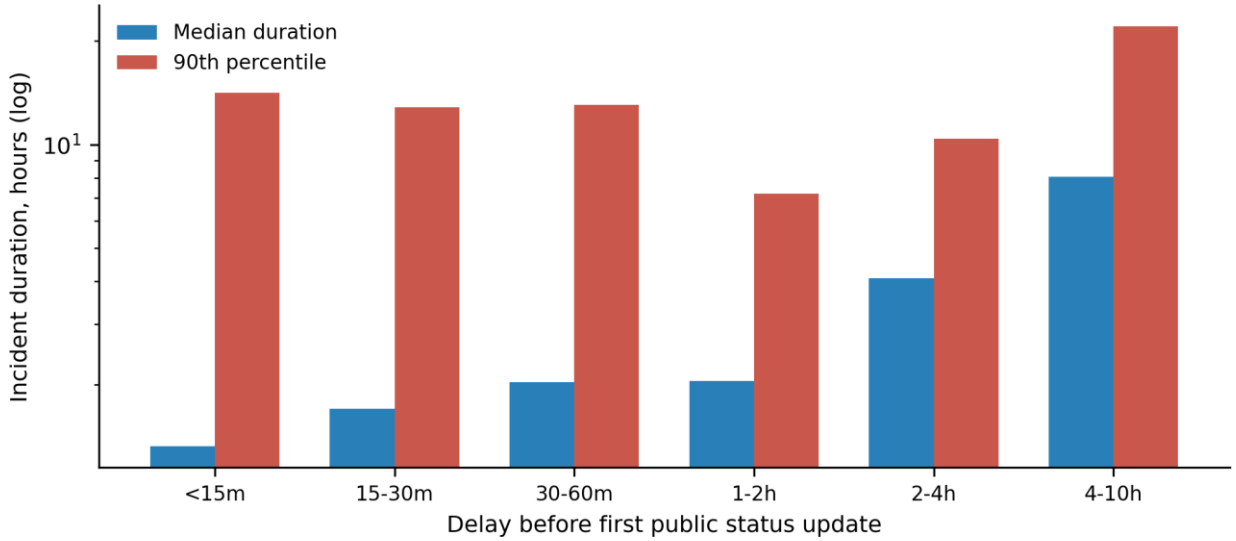

Figure 6. The silence gradient. Median and 90th-percentile incident duration by the delay before the first public status update. Medians rise monotonically from 1.33 hours under fifteen-minute acknowledgement to 8.07 hours after four to ten hours of silence.

5.7 Early triage from the customer's seat

If silence carries signal, a dependent operator should be able to act on it. The prediction protocol of Section 4.3 tests exactly this. On the 2023 to 2024 temporal holdout the gradient boosting model attains an area under the ROC curve of 0.707 and an average precision of 0.424 against a base rate of 0.181, with a Brier score of 0.130. The class-weighted logistic baseline reaches 0.682 and 0.422 but with much poorer calibration (Brier 0.298) (Figure 7). First-update delay dominates the boosting model's importance at 0.55, followed by the first update's word count at 0.17 and the high-severity flag at 0.13. Table 5 gives operating points. At a decision threshold of 0.5 the model flags 47 of 515 test incidents and 62 percent of the flags are true long-tail events, recalling 31 percent of them, while a looser threshold of 0.2 trades precision 0.41 for recall 0.42. Observation O10: at the moment the provider publishes its first update, a customer running this model can concentrate long-tail incidents at two to three times their natural frequency among its alerts, a consumer-side counterpart to provider-side outage prediction from internal alert streams [18], and can trigger failover, workload deferral, or communication plans hours before the status page itself concedes that resolution is distant. During the incident, the cadence result of O9 extends the same logic, since widening gaps between updates justify escalating the customer-side response. The model uses nothing beyond the public feed, so the capability is available to any tenant at zero marginal cost.

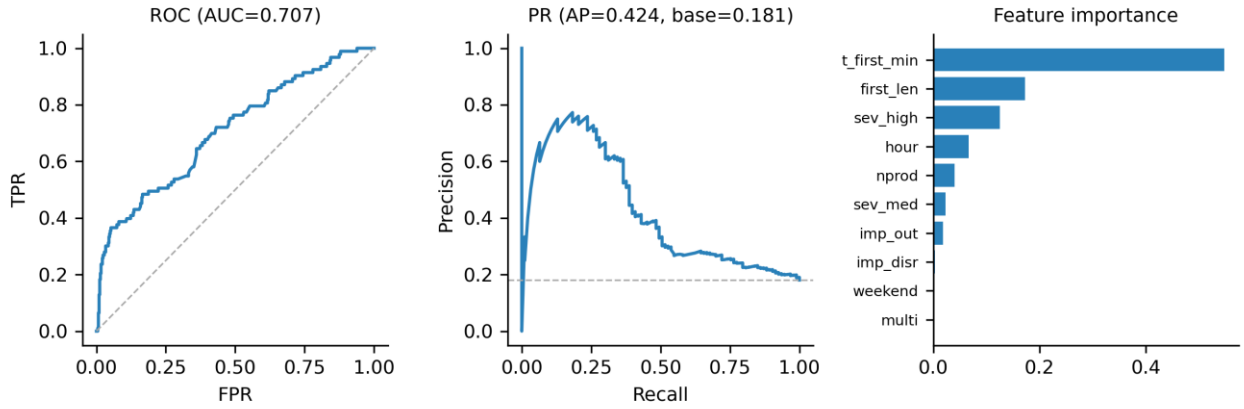

Figure 7. Early-triage model on the 2023 to 2024 temporal holdout. ROC (AUC = 0.707), precision-recall against the 0.181 base rate (AP = 0.424), and gradient-boosting feature importances dominated by first-update delay. Feature codes: t_first_min, minutes to first update; first_len, first-update word count; sev_high and sev_med, severity grades; imp_out and imp_dsr, impact labels; nprod, affected-product count.

Table 5. Operating points of the early-triage model on the 2023 to 2024 holdout (515 incidents, 93 long-tail).

Threshold	Flagged (of 515)	Precision	Recall
0.5	47	0.62	0.31
0.3	72	0.50	0.39
0.2	95	0.41	0.42

5.8 Replication across providers: two regularities, one divergence, and a silent control group

The three additional reconstructions allow every headline pattern to be tested outside Google Cloud (Figure 8, Table 6). Two patterns replicate as regularities of the public incident record. Durations are best fit by a lognormal among the candidate families at every provider, with AIC margins of 61 to 524 over the nearest alternative and log-scale sigmas between 1.01 and 1.57, so median-and-tail reporting is the right summary everywhere, not just at Google (Observation O11). The working-week rhythm is equally universal. Weekday days carry 2.4 times the incident rate of weekend days at DigitalOcean, 3.4 at Cloudflare, 4.5 at GitHub, and 6.2 at Google Cloud, each rejecting uniformity at $p < 0.001$, so change-driven failure is a property of the industry, not of one operator (Observation O12).

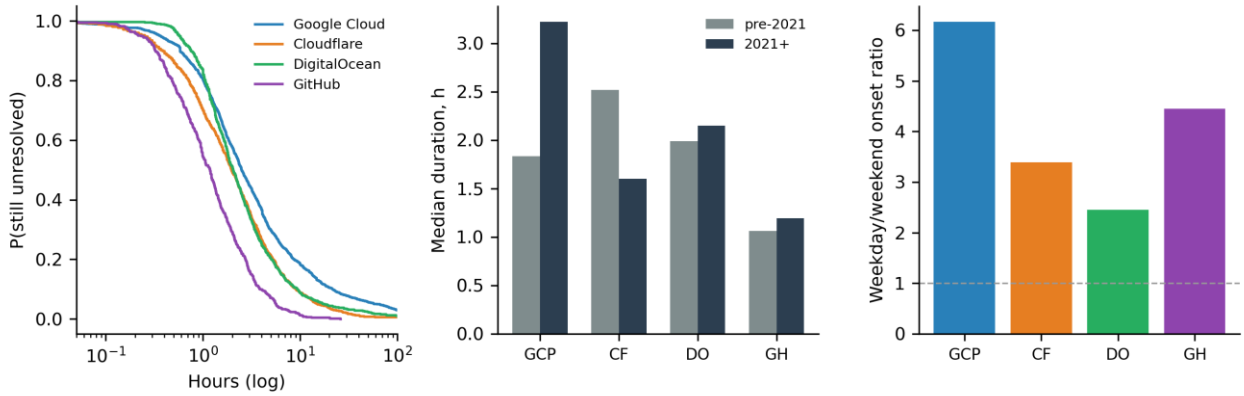

Figure 8. Cross-provider replication. Left: duration survival curves for all four providers, log scale. Centre: era medians before and after 2021; the slowdown is specific to Google Cloud while Cloudflare accelerates. Right: weekday-to-weekend onset rate ratios; the working-week rhythm holds everywhere.

Table 6. Cross-provider replication battery. Era medians in hours; weekday ratio is the weekday-to-weekend daily onset rate, for Google Cloud over its 2019 to 2024 window; silence is the median acknowledgment delay where measurable.

Provider	Incidents	Span	Median pre-2021 (h)	Median 2021+ (h)	Weekday ratio	Median silence
Google Cloud	1,602	2013-2026	1.83	3.22	6.2	33 min
Cloudflare	1,145	2014-2026	2.52	1.60	3.4	n/a
DigitalOcean	961	2017-2026	1.99	2.15	2.5	n/a
GitHub	473	2019-2026	1.06	1.19	4.5	0 min

The post-2021 slowdown, by contrast, does not replicate, and the divergence is the finding. While Google Cloud's median moved from 1.83 to 3.22 hours ($p < 0.001$), Cloudflare moved the opposite way, from 2.52 to 1.60 hours ($p < 0.001$), and DigitalOcean and GitHub were statistically flat, under the first-to-last-update definition and order-inferred years described in Section 3.5. The small offsets from the 1.82 and 3.20 medians of Section 5.1 reflect the full-span window used for the cross-provider battery. Observation O13: the lengthening of incident lifecycles is provider-specific, not an industry constant, which cuts both ways. It undermines aggregating industry narratives that average across providers, and it strengthens the dependency depth reading of Google's own trend, since an industry-wide reporting fashion would have moved all four records together. For procurement the implication is immediate. Reliability trajectories differ enough across providers that the reconstructed public record is worth consulting before contracts are signed, as one input read with the definitional caveats of Section 3.5.

The silence signal receives the most instructive test. GitHub's feed carries a separate onset field alongside the record-creation time, and the measured gap between the two is essentially zero. The median gap between started and created is 0.0 minutes, every measured incident is acknowledged within fifteen minutes, and none waits beyond an hour, against a median of 33 minutes and a 35 percent share beyond one hour at Google Cloud. With no variance in silence there is no silence gradient to detect at GitHub ($\rho = 0.04$, not significant), and that is exactly what the shared-cause mechanism predicts. The signal lives where acknowledgment discipline varies. Observation O14: the silence before the update is not a universal law but a symptom, present in records whose operators acknowledge slowly and variably, absent where the record shows no measurable delay, which makes acknowledgment latency a legible, comparable indicator of communication discipline wherever the record actually measures it.

6. DISCUSSION

6.1 Implications for providers, customers, and the study of reliability

For providers, the silence result reframes status-page communication as an engineering telemetry channel rather than a public-relations obligation. Because first-update delay predicts eventual duration, disciplined early acknowledgement, already doctrine in the postmortem culture of site reliability engineering [20], is not merely courteous. Its absence is informative, and its systematic measurement across incidents would let providers audit their own diagnostic latency. The severity inversion suggests a complementary reform, tail latency objectives for low-severity incidents, since the current

triage economy quietly finances fast resolution of severe events with multi-day persistence of minor ones. The weekday onset rhythm adds a third lever, since failure that follows the deployment calendar is failure that change-management policy can move, spread, or gate. For customers, the findings supply concrete negotiating and engineering material. Service-level discussions can cite a doubled public median, a quadrupled share of day-long incidents, and the fact that most published incidents singly exceed a month's 99.9 percent budget. Procurement can ask what fraction of incident history a provider keeps publicly inspectable. Operations teams can implement the early-triage model directly, since it requires only the feed they already poll, and can treat concurrent-incident arrival as the planning default rather than the exception. The readiness perspective gains an empirical instrument here. Verified recovery is only as trustworthy as the record in which recovery is declared.

For the research community, the cross-provider battery of Section 5.8 shows the archive-union method generalising in practice, through three different source channels, and already sharpening claims. What looked like a possible industry trend is a provider divergence, and what looked like a candidate law of incident communication is a diagnostic of communication discipline. We release corpus, provenance, and code to make such replication mechanical. There is also a preservation lesson in the post-2024 thinning. The public memory of cloud reliability currently depends on a nonprofit archive's crawl schedule, which is a fragile foundation for a record this consequential, and a case for regulated retention of incident histories analogous to financial disclosure archives.

6.2 Threats to validity

Construct validity is bounded by the nature of the source. We measure the published lifecycle of incidents, and begin, end, and severity are the provider's declarations rather than independent observations. The slowdown finding in particular admits the dual reading developed in Section 5.1, and we present it as a property of the public record, not a measurement of internal engineering speed. The cause signatures of Section 5.5 are keyword matches over provider-authored text, transparent and reproducible but coarse. They measure what providers say, not verified root cause, and a language-model extraction layer [26] is the natural refinement. Internal validity of the silence analysis is protected by stratification, which shows the association within every severity level, but unobserved confounders such as incident complexity necessarily remain. We claim prediction, not causation. Collection bias is addressed head-on by the completeness analysis of Section 3.3, which shows that no incident published between 2019 and 2024 can be missing from the corpus and that the time-based retention window makes visibility independent of duration in earlier eras. Incidents still open at the final capture lack an end timestamp and are excluded, a right-truncation that could slightly understate the most recent durations, and all frequencies remain lower bounds on unpublished reality, a limitation intrinsic to any public-record study and, in part, the paper's subject. External validity now rests on four providers of different scales and business models, though depth remains unequal by design. Only Google Cloud carries a completeness bound, the RSS reconstructions have count-based windows with capture-dependent coverage that we disclose per year, their durations run from first to last update rather than backdated begin to end, RSS timestamps lack explicit years and are inferred from feed order, and maintenance events are excluded by status-pattern filtering that may be imperfect at the margin. A further threat concerns feature timing in the triage model. The corpus preserves the final state of each record, so severity, impact label, and product fields could reflect revisions made after the first update, and the first-update delay is measured from a begin timestamp the provider may assign retrospectively. The model is therefore a prioritisation heuristic over the published record rather than a strictly causal early-warning instrument, and re-estimating its features from per-capture snapshots is a natural extension. GitHub's zero silence admits two readings, instantaneous acknowledgment or unbackdated start times, and we draw no conclusion that depends on distinguishing them. Statistical conclusions rest on nonparametric contrasts, bootstrap intervals, a checked proportional-hazards specification, and a strictly temporal prediction split, and every headline result survived winsorisation and era-exclusion checks.

7. CONCLUSION

More than four thousand cloud incidents across four providers were recovered from archives of feeds designed to forget, and the recovered record speaks clearly. Durations follow a lognormal law, and failure follows the working week at every provider studied. At Google Cloud, the deep spine of the corpus, published lifecycles have lengthened while Cloudflare's shortened, the resolution tail has migrated to the incidents graded least severe, most incidents arrive while another is already open, and the provider's opening silence forecasts how an incident will end well enough to power a practical early-warning model that any customer can run, a signal that vanishes precisely at the provider whose acknowledgment is instantaneous. Beyond its specific findings, the paper demonstrates that the ephemeral transparency of status pages can be converted, through web archives, into durable scientific evidence with a verifiable completeness bound over the 2019 to 2024 window, and it argues that the reliability of the record itself now belongs on the cloud-dependability research agenda. Future work deepens the three replication corpora toward completeness bounds of their own, replaces keyword cause signatures with language-model extraction of mitigation actions, and closes the loop by integrating the early-triage signal into dependency-aware recovery orchestration on the customer side.

STATEMENTS AND DECLARATIONS

Acknowledgements: The authors gratefully acknowledge the Internet Archive, whose Wayback Machine and CDX index preserved the captures from which the incident corpus was reconstructed, and the four providers whose public status feeds constitute the primary record.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing interests: The authors declare no competing interests.

Ethics approval: Not applicable. The study analyses publicly available operational records of computing infrastructure and involves no human participants or personal data.

Data availability: Source captures are independently retrievable from the Internet Archive [41] and the provider's live feed [42]. The reconstructed incident corpus, the capture level provenance file, and all analysis code are openly deposited on Zenodo under a persistent identifier [43] and are sufficient to reproduce every number, table, and figure in this article.

Author contributions: A. A. Akintade designed the study, performed the data collection and analysis, and wrote the manuscript. W. A. Bello contributed to the design of the analysis and to critical revision of the manuscript. H. O.

Ekpobimi contributed to the validation of the reconstructed corpus and to critical revision of the manuscript. All authors read and approved the final manuscript.

Use of generative artificial intelligence: Artificial intelligence tools were used solely to edit the authors' own text for language. All other work, including the reconstruction of the incident corpus, the statistical analysis, the predictive modelling, interpretation and the drafting of all substantive content, was carried out by the authors alone, who take full responsibility for the content of this article.

8. REFERENCES

[1] Gray, J. 1985. Why do computers stop and what can be done about it? Tandem Technical Report TR-85.7. Tandem Computers, Cupertino.

[2] Oppenheimer, D., Ganapathi, A., and Patterson, D. A. 2003. Why do Internet services fail, and what can be done about it? In Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS). USENIX, Seattle.

[3] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. 2004. Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing 1(1), 11-33. https://doi.org/10.1109/TDSC.2004.2

[4] Gunawi, H. S., Hao, M., Suminto, R. O., Laksono, A., Satria, A. D., Adityatama, J., and Eliazar, K. J. 2016. Why does the cloud stop computing? Lessons from hundreds of service outages. In Proceedings of the 7th ACM Symposium on Cloud Computing (SoCC), pp. 1-16. ACM, Santa Clara. https://doi.org/10.1145/2987550.2987583

[5] Chen, J., He, X., Lin, Q., Xu, Y., Zhang, H., Hao, D., Gao, F., Xu, Z., Dang, Y., and Zhang, D. 2019. An empirical investigation of incident triage for online service systems. In Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp. 111-120. IEEE, Montreal. https://doi.org/10.1109/ICSE-SEIP.2019.00020

[6] Chen, Z., Kang, Y., Li, L., Zhang, X., Zhang, H., Xu, H., et al. 2020. Towards intelligent incident management: why we need it and how we make it. In Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), pp. 1487-1497. ACM. https://doi.org/10.1145/3368089.3417055

[7] Ghosh, S., Shetty, M., Bansal, C., and Nath, S. 2022. How to fight production incidents? An empirical study on a large-scale cloud service. In Proceedings of the 13th ACM Symposium on Cloud Computing (SoCC), pp. 126-141. ACM, San Francisco. https://doi.org/10.1145/3542929.3563482

[8] Notaro, P., Cardoso, J., and Gerndt, M. 2021. A survey of AIOps methods for failure management. ACM Transactions on Intelligent Systems and Technology 12(6), Article 81. https://doi.org/10.1145/3483424

[9] Schroeder, B. and Gibson, G. A. 2006. A large-scale study of failures in high-performance computing systems. In Proceedings of the International Conference on Dependable Systems and Networks (DSN), pp. 249-258. IEEE. https://doi.org/10.1109/DSN.2006.5

[10] Vishwanath, K. V. and Nagappan, N. 2010. Characterizing cloud computing hardware reliability. In Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC), pp. 193-204. ACM. https://doi.org/10.1145/1807128.1807161

[11] Ford, D., Labelle, F., Popovici, F. I., Stokely, M., Truong, V.-A., Barroso, L., Grimes, C., and Quinlan, S. 2010. Availability in globally distributed storage systems. In Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 61-74. USENIX.

[12] Xu, W., Huang, L., Fox, A., Patterson, D., and Jordan, M. I. 2009. Detecting large-scale system problems by mining console logs. In Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP), pp. 117-132. ACM. https://doi.org/10.1145/1629575.1629587

[13] Huang, P., Guo, C., Zhou, L., Lorch, J. R., Dang, Y., Chintalapati, M., and Yao, R. 2017. Gray failure: the Achilles' heel of cloud-scale systems. In Proceedings of the 16th Workshop on Hot Topics in Operating Systems (HotOS), pp. 150-155. ACM. https://doi.org/10.1145/3102980.3103005

[14] Bronson, N., Aghayev, A., Charapko, A., and Zhu, T. 2021. Metastable failures in distributed systems. In Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS), pp. 221-227. ACM. https://doi.org/10.1145/3458336.3465286

[15] Alquraan, A., Takruri, H., Alfatafta, M., and Al-Kiswany, S. 2018. An analysis of network-partitioning failures in cloud systems. In Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 51-68. USENIX.

[16] Liu, H., Lu, S., Musuvathi, M., and Nath, S. 2019. What bugs cause production cloud incidents? In Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS), pp. 155-162. ACM. https://doi.org/10.1145/3317550.3321438

[17] Yuan, D., Luo, Y., Zhuang, X., Rodrigues, G., Zhao, X., Zhang, Y., Jain, P. U., and Stumm, M. 2014. Simple testing can prevent most critical failures: an analysis of production failures in distributed data-intensive systems. In Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 249-265. USENIX.

[18] Chen, Y., Yang, X., Lin, Q., Zhang, H., Gao, F., Xu, Z., Dang, Y., Zhang, D., Dong, H., Xu, Y., Li, H., and Chen, Y. 2019. Outage prediction and diagnosis for cloud service systems. In Proceedings of the World Wide Web Conference (WWW), pp. 2659-2665. ACM. https://doi.org/10.1145/3308558.3313501

[19] Dang, Y., Lin, Q., and Huang, P. 2019. AIOps: real-world challenges and research innovations. In Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings (ICSE Companion), pp. 4-5. IEEE. https://doi.org/10.1109/ICSE-Companion.2019.00023

[20] Beyer, B., Jones, C., Petoff, J., and Murphy, N. R. (eds.) 2016. Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media, Sebastopol.

[21] Chu, X., Talluri, S., Lu, Q., and Iosup, A. 2025. An empirical characterization of outages and incidents in public services for large language models. In Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering (ICPE), Toronto. https://doi.org/10.1145/3676151.3719372

[22] Uptime Institute. 2024. Annual Outage Analysis 2024. Uptime Institute Intelligence Report, New York.

[23] Parametrix. 2025. Cloud Outage Risk Report 2024. Parametrix Insurance, New York.

[24] Ogden, J., Summers, E., and Walker, S. 2024. Know(ing) infrastructure: the Wayback Machine as object and instrument of digital research. Convergence 30(1), 391-410. https://doi.org/10.1177/13548565231164759

[25] Hauer, T., Hoffmann, P., Lunney, J., Ardelean, D., and Diwan, A. 2020. Meaningful availability. In Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 545-557. USENIX, Santa Clara.

[26] Goel, D., Husain, F., Singh, A., Ghosh, S., Parayil, A., Bansal, C., Zhang, X., and Rajmohan, S. 2024. X-lifecycle learning for cloud incident management using LLMs. In: Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering (FSE Companion), pp. 417-428. ACM. https://doi.org/10.1145/3663529.3663861

[27] Yan, H., Chen, Y., Ma, M., Wen, M., Lu, S., Zhang, S., Xu, T., Wang, R., Bansal, C., Rajmohan, S., Lin, Q., Zhang, C., and Zhang, D. 2025. An empirical study of production incidents in generative AI cloud services. In Proceedings of the 36th IEEE International Symposium on Software Reliability Engineering (ISSRE), pp. 359-370. IEEE. https://doi.org/10.1109/ISSRE66568.2025.00044

[28] Sillito, J. and Kutomi, E. 2020. Failures and fixes: a study of software system incident response. In Proceedings of the IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 185-195. IEEE. https://doi.org/10.1109/ICSME46990.2020.00027

[29] Zhang, S., Xia, S., Fan, W., Shi, B., Xiong, X., Zhong, Z., Ma, M., Sun, Y., and Pei, D. 2025. Failure diagnosis in microservice systems: a comprehensive survey and analysis. ACM Transactions on Software Engineering and Methodology. https://doi.org/10.1145/3715005

[30] Ahsan, M., Talluri, S., and Iosup, A. 2022. Failure analysis of big cloud service providers prior to and during Covid-19 period. arXiv preprint arXiv:2210.08006.

[31] Ghosh, S., Grover, K., Wong, J., Bansal, C., Namineni, R., Verma, M., and Rajmohan, S. 2024. Dependency aware incident linking in large cloud systems. In: Companion Proceedings of the ACM Web Conference (WWW Companion). ACM. https://doi.org/10.1145/3589335.3648311

[32] Akintade, A. A. 2026. From detection to verified action: operational readiness for AI-enabled cloud failure management. International Journal for Research in Applied Science and Engineering Technology 14(VIII), 812-828. https://doi.org/10.22214/ijraset.2026.84530

[33] Akintade, A. A. 2026. Restoring minimum viable service under cascading failures: dependency-aware multi-objective recovery orchestration for multi-site hybrid clouds. International Journal for Research in Applied Science and Engineering Technology 14(VIII), 829-845. https://doi.org/10.22214/ijraset.2026.84531

[34] Mann, H. B. and Whitney, D. R. 1947. On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics 18(1), 50-60. https://doi.org/10.1214/aoms/1177730491

[35] Kaplan, E. L. and Meier, P. 1958. Nonparametric estimation from incomplete observations. Journal of the American Statistical Association 53(282), 457-481. https://doi.org/10.1080/01621459.1958.10501452

[36] Cox, D. R. 1972. Regression models and life-tables. Journal of the Royal Statistical Society, Series B 34(2), 187-202. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x

[37] Davidson-Pilon, C. 2019. Lifelines: survival analysis in Python. Journal of Open-Source Software 4(40), 1317. https://doi.org/10.21105/joss.01317

[38] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. 2011. Scikit-learn machine learning in Python. Journal of Machine Learning Research 12, 2825-2830.

[39] Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. Annals of Statistics 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

[40] Baset, S. A. 2012. Cloud SLAs: present and future. ACM SIGOPS Operating Systems Review 46(2), 57-66. https://doi.org/10.1145/2331576.2331586

[41] Internet Archive. 2026. Wayback Machine. https://web.archive.org. Accessed 7 September 2026.

[42] Google Cloud. 2026. Google Cloud Service Health incident feed. https://status.cloud.google.com/incidents.json. Accessed 6 September 2026.

[43] Akintade, A. A., Bello, W. A., and Ekpobimi, H. O. 2026. Reproducibility package for "The silence before the update: reconstructing a decade of cloud incident lifecycles across four providers from the public record" (Version 1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.22766249