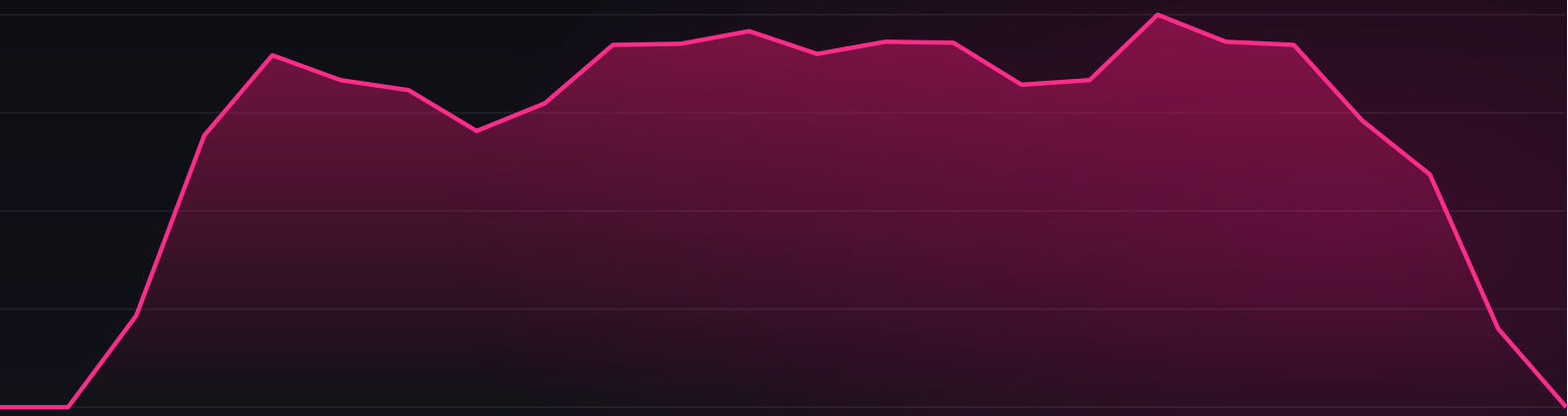
APICONTEXT RESEARCH

Anatomy of the Microsoft 365 outage

This study compares Microsoft's incident chronology with APIContext's scheduled Graph checks.



17:29:40
12 NOV · FIRST OBSERVED FAILURE

95.8%
13 NOV 02:00 · OVERNIGHT PEAK

13:33:14
13 NOV · LAST OBSERVED FAILURE

30.006 s
wait for the first response byte from Toronto, then HTTP 504

37 of 43
retained HTTP 502 responses waited between 21.0 and 21.1 seconds

6
monitored continents, four Exchange-backed operations

Silent failures break businesses.

13.25 ms

DNS resolution, TCP connection and TLS negotiation in total

30.006 s

wait for the first response byte, then HTTP 504

37 of 43

retained HTTP 502 responses with a post-upload wait between 21.0 and 21.1 seconds

IN THIS STUDY

01 Two clocks for the same incident 03

02 One request, six intervals 05

03 DNS, transport and the limits of an outside observation 07

04 Four operations with a common wait 09

05 Product dependencies and a regional recovery tail 11

06 Could an SRE have used this evidence to limit the blast radius? 14

On 12 November 2026 at 17:29:40 UTC, APIContext's Toronto probe requested a list of mail folders from Microsoft Graph. DNS resolution, TCP connection and TLS negotiation took 13.25 milliseconds in total. After sending the request, **the probe waited 30.006 seconds for the first response byte and received HTTP 504.**

Microsoft's connectivity monitoring detected an anomaly at 16:03 UTC, and its on-call engineers received an Exchange Online alert at 16:20. The external measurements add a transaction-level finding: **37 of 43 retained HTTP 502 responses recorded a post-upload wait between 21.0 and 21.1 seconds.** That band appeared across four Exchange-backed operations and six monitored continents.

This study compares Microsoft's incident chronology with APIContext's scheduled Graph checks. It examines which operations failed together, where elapsed time accumulated and which failures remained observable during recovery. It also evaluates what additional internal evidence a site reliability engineer (SRE) would need to use those observations in a routing decision.

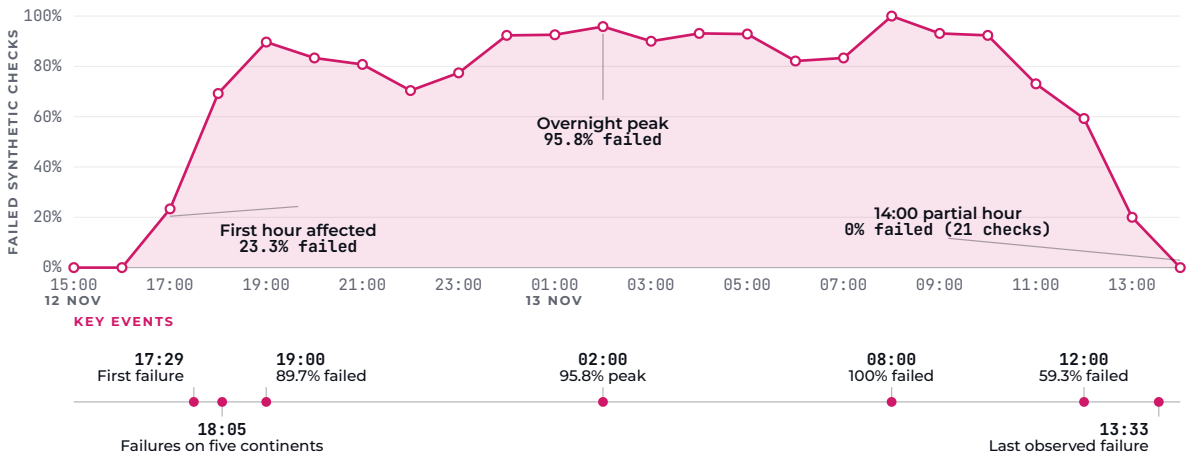
FIGURE 01

Microsoft Graph Exchange-backed API incident timeline

95.8%

failed at the overnight peak · 02:00

APICONTEXT SYNTHETIC MONITORING · 12–13 NOVEMBER 2026 · REFRESHED 14:41 GMT



SOURCE · APICONTEXT APIContext CLI, Enterprise IT / Microsoft Graph (Office). Seven Exchange-backed Graph monitors. Synthetic checks are not user counts. Final hour is partial to 14:41 GMT.

Figure 1. Incident timeline through the 14:41 update.

CHAPTER 01

Two clocks for the same incident

UTC	MICROSOFT · FLEET AND SERVICE	APICONTEXT · SCHEDULED CUSTOMER OPERATIONS
12 Nov DETECTION	16:03 connectivity anomaly; 16:20 Exchange alert	16:00 hour: 27 of 27 checks passed. First observed failure began 17:29:40.
12 Nov ESCALATION	17:45 access-path errors under investigation	17:00 hour: six HTTP 504s, one 502. Toronto recorded a 30-second post-upload wait.
13 Nov DIAGNOSIS	00:30 investigation narrowed to connection negotiation	00:00 hour: 24 of 26 affected-operation checks failed. The client leg of a connection and the internal gateway leg can behave differently.
13 Nov REPAIR	07:20 mitigation applied to part of the fleet	At 10:00, all 15 checks on four selected Exchange monitors failed; all 13 on three selected profile/drive controls passed.
13 Nov RECOVERY	11:35 Europe healthy in aggregate; 14:17 global webmail at expected availability	13:33 Frankfurt contacts read returned HTTP 502. All 21 affected-monitor follow-ups from 14:00 through 14:41 passed.

14%

nearly, of gateway capacity was unavailable

Microsoft's final report of 20 November identifies **four interacting conditions.** A defect introduced in July exposed lock contention on newer gateway machines under a particular mix of traffic and concurrency. Nearly 14% of gateway capacity was unavailable. Teams traffic had moved onto the gateway shared with Exchange earlier that day. Peak demand then brought the remaining fleet into the conditions that triggered the defect. Redirecting traffic spread the pressure. A working fallback under ordinary load became a route by which surviving machines encountered the same fault.

> **A working fallback under ordinary load became a route by which surviving machines encountered the same fault.**

36 min

roughly, from the first observation to failures from five continents

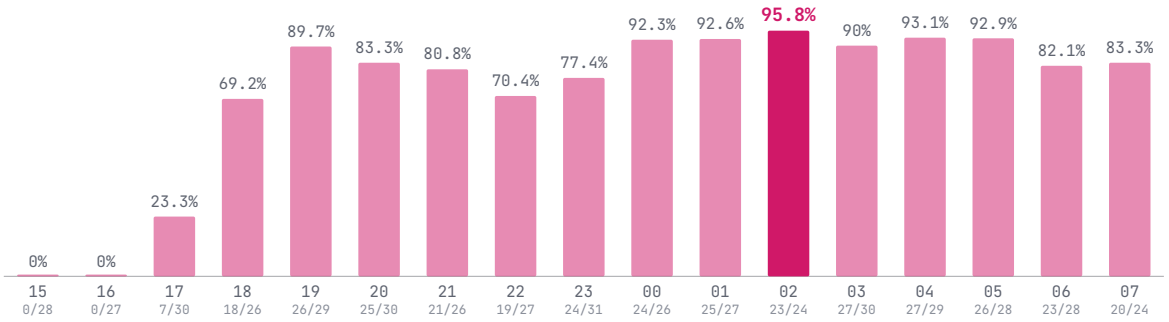
Our clock starts with a different event: the first scheduled customer-side check in our record to return a failure. At 17:29 UTC that was the Toronto call. **Within roughly 36 minutes of that first observation we had failures from five continents.** That interval depends on where checks were scheduled and when each ran. It measures how quickly the problem became observable across this network. It cannot establish a route from the first country to the fifth, or the minute at which the underlying defect reached each one.

FIGURE 02

How quickly did Exchange-backed checks deteriorate?

FAILED CHECKS / ALL CHECKS ON AFFECTED OPERATIONS, BY START HOUR · 12–13 NOVEMBER UTC

26 of 29
checks failed by the 19:00 hour



SOURCE · APICONTEXT Last bucket ends at 07:55 UTC. Synthetic check rate, independent of user population.

Figure 2. Failure rate over time.

335 of 470
checks failed in the hourly series, 15:00 Thursday to the 07:55 UTC Friday cutoff

In the 16:00 UTC hour, all 27 checks in our affected-operation series passed even as Microsoft was detecting trouble. In the 17:00 hour seven of 30 failed. By 19:00, 26 of 29 failed. At 02:00 on Friday, 23 of 24 failed. The 335 failed checks cited in our initial coverage belong to an export ending at 07:55 UTC on Friday. The hourly series contains 470 checks from 15:00 Thursday to that cutoff. Further failures on Friday afternoon appear in a later collection. **Adding those to the numerator while keeping the original denominator would inflate the failure rate.**

The two monitoring systems sampled different populations. At 16:20, Microsoft had evidence of Exchange impact while the selected external monitors were still recording successes. By the 19:00 hour, external failures were frequent enough to provide reproducible examples with known start times, operations and durations for comparison with gateway records.

HTTP 504
A responder reporting a timeout.

HTTP 502
A responder reporting a problem obtaining a usable upstream response.

Six of the seven failures in the first affected hour were HTTP 504s. In the 06:00 hour on Friday, 21 of the 23 failures were 502s. Both are gateway-class responses, but **collapsing them into one red availability line loses the change in behaviour.** A 504 records a responder reporting a timeout. A 502 records a responder reporting a problem obtaining a usable upstream response. Neither status identifies the component that generated it. Their changing mix gives investigators a reason to compare examples from different hours before assuming one timeout policy explains the entire incident.

FIGURE 03

Gateway errors changed overnight

FAILURE COUNTS PER HOUR · 12–13 NOVEMBER UTC

21 of 23
failures were 502s in the 06:00 hour

HTTP	15	16	17	18	19	20	21	22	23	00	01	02	03	04	05	06	07
500	0	0	0	0	3	2	4	0	1	2	1	0	2	4	3	0	0
502	0	0	1	9	8	14	11	8	18	10	19	17	15	16	18	21	18
503	0	0	0	2	5	4	2	6	3	5	2	1	2	1	0	0	2
504	0	0	6	7	9	5	4	5	2	7	3	5	8	6	5	2	0
other	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

SOURCE · APICONTEXT Shading scales within each hour. The final hour is partial.

Figure 3. HTTP codes per hour.

CHAPTER 02

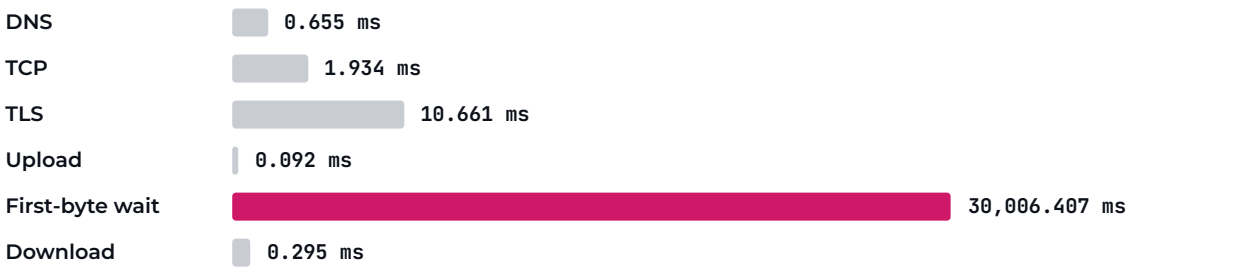
One request, six intervals

FIGURE 04

Inside the first observed mail-folder timeout

TORONTO · 12 NOVEMBER 17:29:40 UTC · HTTP 504

30,006.407 ms
first-byte wait



SOURCE · APICONTEXT Bar lengths use log10(1 + milliseconds) so sub-millisecond intervals remain labelled.

Figure 4. Toronto request phases.

The Toronto record below is a field-limited extract of a real result. Times are milliseconds. We retained the operation name and monitoring location and removed account identifiers, headers, content and authentication context. Reading the request from left to right in time is more informative than its total duration. Lookup, TCP and TLS together consumed 13.250 ms. The upload took 0.092 ms. **The phase the agent calls processing took 30,006.407 ms.** Once the first byte arrived, the rest of the response took 0.295 ms to download.

30,006.407 ms
the phase the agent calls processing

TORONTO · MAILFOLDERS · HTTP 504

```
{
  "time": "2026-11-12T17:29:40.889704Z",
  "location": "public-linode-ca-central",
  "region": "NA",
  "city": "Toronto",
  "operation": "mailFolders",
  "status": 504,
  "ms": {
    "dns": 0.655,
    "download": 0.295,
    "handshake": 10.661,
    "processing": 30006.407,
    "tcp": 1.934,
    "total": 30020.044,
    "upload": 0.092
  }
}
```

A client can observe when its request leaves and when a response begins. Between those events there may be a proxy, retries, admission control, a queue, another secure connection or an application doing work. **The processing field spans that wait from the client's vantage point. It does not isolate CPU time at the origin.** Likewise, "time to edge" below means the observed time to establish a usable secure connection, including lookup. The identity and physical location of the machine that eventually performed mailbox work cannot be recovered from those three timers.

CHAPTER 03

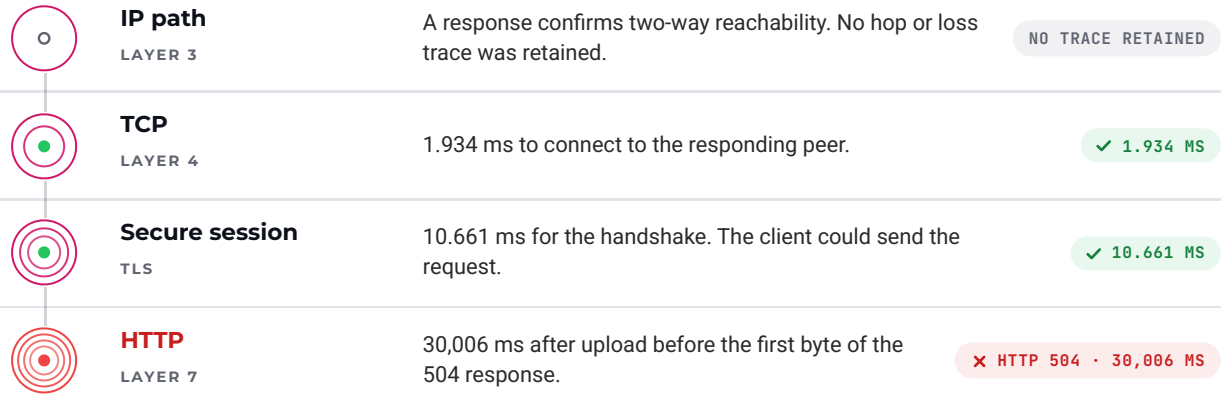
DNS, transport and the limits of an outside observation

FIGURE 05

Where did the customer transaction stop making progress?

TORONTO MAIL-FOLDER READ · 17:29:40 UTC ON 12 NOVEMBER

13.25 ms
lookup, TCP and TLS together



SOURCE · APICONTEXT TLS provides one measured interval between TCP and HTTP. Upstream hops are outside the client trace.

Figure 5. The measured boundaries of one request.

8.89 ms
mean lookup time across 335 failed-result samples

APIContext's project-level DNS query returned 335 failed-result samples between 17:20 UTC on Thursday and 07:55 UTC on Friday, with a mean lookup time of 8.89 ms. The response contained no per-monitor breakdown. In the Toronto trace, resolution took 0.655 ms. **DNS latency therefore accounted for little of that request's delay.** These measurements do not establish whether a resolver selected an optimal destination or returned a cached answer.

63
distinct Graph requests with detailed phase timings: fifty errors, thirteen HTTP 200

We also recovered 63 distinct Graph requests with detailed phase timings. Fifty produced errors and thirteen returned HTTP 200. The sample reflects earlier targeted investigations, especially lookups of failed requests during recovery. Its all-request mean DNS, TCP and TLS times were 3.21, 10.09 and 24.32 ms respectively. The smaller DNS mean here and the 8.89 ms from the larger failed-result query describe different populations. Neither is a pre-incident baseline. **The comparisons below deliberately keep counts next to measurements so a two-request cell cannot masquerade as a regional trend.**

Consider South America. The three retained requests averaged about 35 ms for TCP and 77 ms for TLS. Europe's twelve averaged about 5 ms and 16 ms. Those differences deserve attention when comparing normal interactive performance.

FIGURE 06

Where elapsed time accumulated, by HTTP outcome

21,140.9 ms
mean first-byte wait · 43 HTTP 502s

MEAN MILLISECONDS PER PHASE · 63 DISTINCT RETAINED REQUESTS · EACH COLUMN HAS ITS OWN COLOUR SCALE

HTTP	DNS	TCP	TLS	Upload	First-byte wait	Download
200 n=13	2.7	5.7	16.4	0.1	563.2	33.6
500 n=2	4.1	14.1	25.1	0.2	2,756.3	0.0
502 n=43	3.4	10.6	25.2	0.1	21,140.9	0.2
503 n=2	2.9	26.1	59.4	0.1	5,238.2	0.1
504 n=3	1.7	8.5	21.8	0.1	30,294.2	0.3

SOURCE · APICONTEXT Sample includes targeted failure lookups. Counts describe coverage, not prevalence.

Figure 6. Phases by HTTP status.

FIGURE 07

The regional view of DNS, TCP, TLS and first-byte wait

76.8 ms
mean TLS in South America · 15.9 ms in Europe

MEAN MILLISECONDS PER PHASE · 63 DISTINCT RETAINED REQUESTS · EACH COLUMN HAS ITS OWN COLOUR SCALE

REGION	DNS	TCP	TLS	Upload	First-byte wait	Download
AF n=2	2.1	11.6	27.0	0.1	25,939.4	0.5
AS n=22	3.0	12.0	28.3	0.1	15,970.2	18.2
EU n=12	2.7	5.4	15.9	0.1	20,541.5	0.1
NA n=13	2.4	9.7	22.8	0.1	10,541.7	3.1
OC n=11	5.1	4.8	12.5	0.1	15,760.7	0.2
SA n=3	4.4	35.1	76.8	0.1	21,041.3	0.2

SOURCE · APICONTEXT Sample includes targeted failure lookups. Counts describe coverage, not prevalence.

Figure 7. Phases by region.

In these samples the mean post-upload wait in both groups exceeded 20 seconds. An engineer could reasonably keep investigating regional transport variation while prioritising the much larger wait shared across locations. Declaring a continent's network degraded would require matched healthy measurements from the same agents, destinations and hours. **Two samples from Africa cannot support a continent-wide percentile.**

Where does Layer 3 fit? A completed TCP connection demonstrates that packets travelled in both directions between the probe and a reachable peer. An HTTP error over TLS demonstrates further progress. Neither measurement supplies the route those packets followed, per-hop loss, retransmission counts, asymmetric paths or upstream connectivity from the gateway. TCP connect timing is a Layer 4 observation influenced by Layer 3 conditions. DNS and TLS are separate protocol stages. Labelling the DNS timer “Layer 3” would claim visibility we did not collect. Route traces or packet-level observations on the affected paths would be needed to adjudicate a routing hypothesis directly.

> **Labelling the DNS timer “Layer 3” would claim visibility we did not collect.**

CHAPTER 04

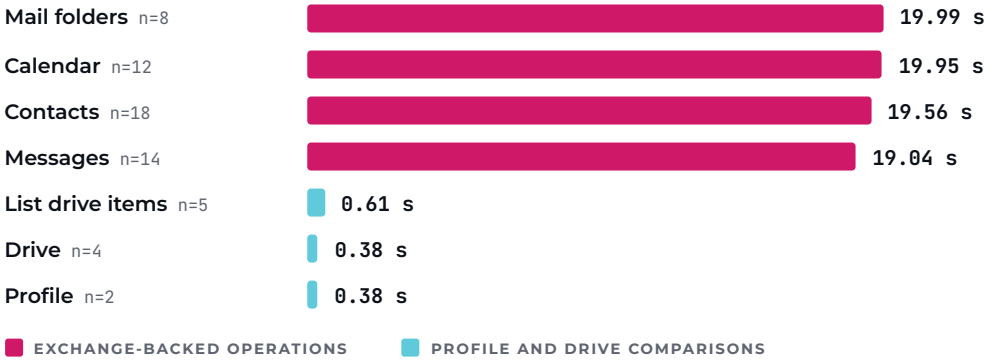
Four operations with a common wait

FIGURE 08

Mailbox operations waited about 20 seconds

MEAN WAIT AFTER REQUEST UPLOAD, IN SECONDS · COUNTS ARE RETAINED REQUESTS

19–20 s
mean post-upload wait · four mailbox operations



SOURCE · APICONTEXT 63 selected requests. Failure-focused sample; no population estimate.

Figure 8. Mean post-upload wait by operation in 63 retained requests.

Requests for calendars, contacts, messages and mail folders used different Graph resources. **In our retained traces each of the four averaged between 19 and 20 seconds in the post-upload phase.** The small comparison samples for profile, drive metadata and drive children averaged well below one second in that phase. These are descriptive values for the selected transactions. They are particularly useful as a guide to which individual records to compare. A ratio of pooled means would mix the incident’s effect with differences in location, response body, observation time and how we selected the records.

FIGURE 09

From lookup to a usable connection to first byte

MEAN MILLISECONDS · SETUP INCLUDES DNS + TCP + TLS · GROUPS HAVE DIFFERENT MIXES OF OPERATIONS

39.2 ms
mean connection setup before a 21-second wait · HTTP 502

GROUP	Connection setup	Post-upload wait	Body download
All retained (63)	37.6	16,242.1	7.1
HTTP 200 (13)	24.9	563.2	33.6
HTTP 502 (43)	39.2	21,140.9	0.2
HTTP 504 (3)	31.9	30,294.2	0.3

SOURCE · APICONTEXT No inference about individual server processing time or physical edge location.

Figure 9. Connection setup and first-byte wait.

563 ms
mean post-upload wait among the thirteen HTTP 200s

21,141 ms
among the 43 HTTP 502s

Among the thirteen HTTP 200s, the mean post-upload wait was 563 ms. Among the 43 HTTP 502s it was 21,141 ms. The latter group averaged about 39 ms for DNS, TCP and TLS combined and 0.15 ms for download. A small error response is cheap to transfer once it exists. The user still waited around twenty seconds to receive it. That asymmetry explains why measuring throughput, response size or the successful completion of TLS alone could give a reassuring view of a transaction that was functionally unusable.

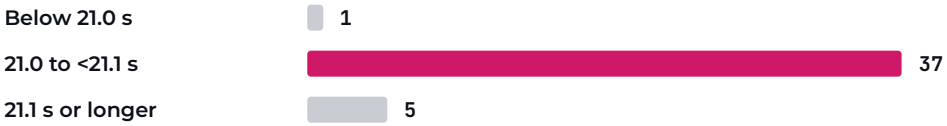
> **A small error response is cheap to transfer once it exists. The user still waited around twenty seconds to receive it.**

FIGURE 10

37 of 43 sampled 502s fell in one narrow band

TIME BETWEEN REQUEST UPLOAD AND FIRST RESPONSE BYTE, GROUPED BY DURATION

37 of 43
sampled 502s between 21.0 and 21.1 s



Comparison: all three sampled 504s waited at least 30 seconds.

SOURCE · APICONTEXT One count per request. The timing pattern does not identify the timer owner.

Figure 10. Post-upload waits for the 43 sampled 502 responses and three sampled 504s.

Inspect the individual values behind that mean. **Thirty-seven of 43 HTTP 502s fall between 21,000 and 21,100 ms after upload.** The band includes calendars, mail folders, contacts and messages, with observations from all six continents.

One 502 arrived after about 9.6 seconds and another after about 31.6. All three sampled 504s waited at least 30 seconds. The 21-second concentration is consistent with a shared timer or repeated sequence of waits somewhere behind the responding interface. It is also possible that several layers used similar budgets. The external record alone cannot assign the timer to Graph, Exchange or a particular gateway process.

The 21-second band provides a search criterion for an internal trace comparison. Matching these requests to downstream deadlines, retry counts and queue age could identify where the budget expired. A discrepancy between an internal span and the client-observed wait would identify an instrumentation gap. Establishing why the work exceeded its deadline requires evidence from inside the service.

CHAPTER 05

Product dependencies and a regional recovery tail

The geography changed as the incident developed. In the 21:00 hour, 14 of the 21 failed checks came from North America. By 06:00 Friday, Europe accounted for 12 of 23. The overnight export contains regional failure counts, but lacks the matching hourly regional totals required to estimate the chance of failure in each market. We can use those cells to find a request or choose where to run a follow-up. **We cannot rank customers' experiences across continents by treating the count as a percentage.** The location of an agent also tells us little about which Microsoft region ultimately served it.

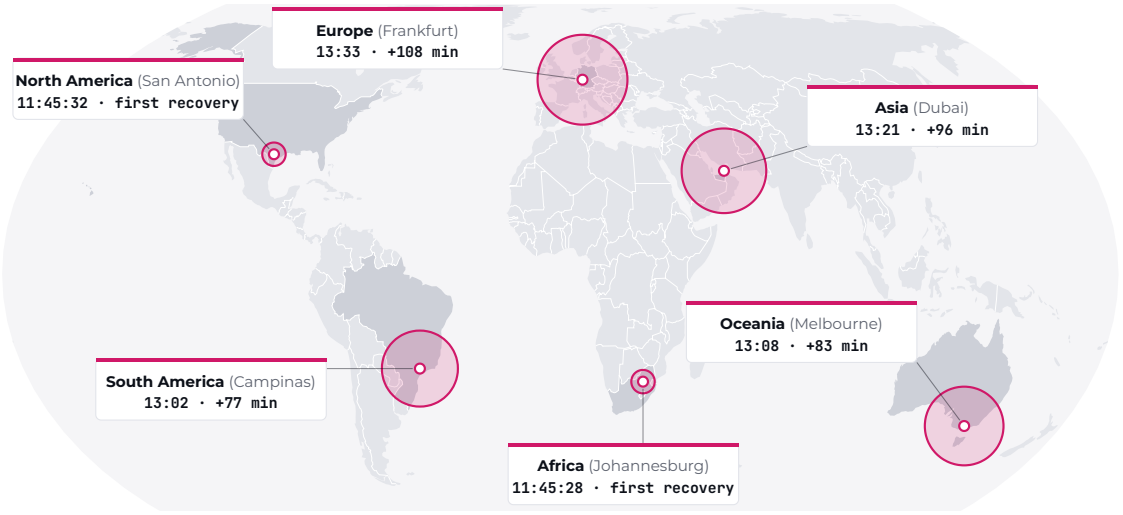
At 10:00 UTC Friday the contrast between operations was especially clear. The calendar, contacts and messages monitors shown here each failed four times out of four. Mail folders failed three times out of three. During that hour, the profile monitor passed four times, the drive monitor passed five and listing drive children passed four. These calls shared the public Graph service name. **This narrows the scope of the observed failure to particular operations behind that interface.** It also gives an on-call engineer working examples from the same hour to compare with the failing set.

FIGURE 11

Microsoft Graph Exchange-backed API recovery by region

+108 min
Frankfurt after the first observed recovery

APICONTEXT SYNTHETIC MONITORING · 13 NOVEMBER 2026 · LAST OBSERVED FAILURE BY PROBE REGION, GMT



RECOVERY-TAIL DELAY



Bubble area shows how many minutes failures continued in a region after recovery was first observed (11:45 GMT). Each label gives the time of the last failure seen from that probe location.

SOURCE · APICONTEXT Observed recovery = last failure seen from APIContext probes; it is not Microsoft's internal regional restoration timestamp. Synthetic checks are not user counts.

Figure 11. Last observed failure by probe geography.

FIGURE 12

Where failures appeared during the overnight escalation

14 of 21
failed checks came from North America · 21:00

FAILED CHECKS BY RECORDED GEOGRAPHY · 12–13 NOVEMBER UTC

REGION	15	16	17	18	19	20	21	22	23	00	01	02	03	04	05	06	07
Africa	0	0	0	0	0	0	0	0	0	1	0	0	1	2	0	1	0
Asia	0	0	0	4	9	8	6	3	11	8	8	6	7	9	11	6	8
Europe	0	0	1	4	6	4	0	2	0	2	4	7	5	3	2	12	8
North America	0	0	4	6	7	11	14	12	12	11	9	7	11	7	7	3	3
Oceania	0	0	1	2	2	2	0	1	1	2	2	2	1	5	4	0	0
South America	0	0	1	2	2	0	1	1	0	0	2	1	2	1	2	1	1

SOURCE · APICONTEXT Counts only. Scheduling and coverage vary across continents. Final hour is partial.

Figure 12. Hourly failure counts by region.

FIGURE 13

Which customer operations were still failing?

MONITOR-LEVEL FAILED SHARE · 13 NOVEMBER UTC · INDIVIDUAL HOURLY SAMPLES ARE SMALL

15 of 15

Exchange-backed checks failed at 10:00; all 13 controls passed

MONITOR	08:00	09:00	10:00	11:00	12:00	13:00	14:00
Calendar	100%	100%	100%	75%	50%	0%	0%
Contacts	100%	100%	100%	75%	67%	25%	0%
Messages	100%	100%	100%	100%	67%	20%	0%
Mail folders	100%	75%	100%	75%	60%	33%	0%
Profile	0%	0%	0%	–	0%	0%	–
Drive	0%	0%	0%	–	0%	0%	–
List drive items	0%	0%	0%	0%	0%	0%	–
Token refresh	–	–	–	–	–	–	–

SOURCE · APICONTEXT Dash = no retained bucket. Latest hour observed through 14:41. See dataset for check counts.

Figure 13. Service overlay.

Microsoft describes Teams traffic moving to a shared gateway and Exchange dependencies affecting features in other products. Our Graph checks independently establish the split between Exchange-backed and file or profile operations. They do not measure Teams calls, Copilot answers or end-to-end mail delivery. For diagnosis, the relevant overlay is therefore twofold: Microsoft explains which products shared infrastructure, and the monitor series shows which directly tested operations succeeded in the same hour. Treating every affected product as a separately measured outage would obscure the shared dependency that made the comparison useful.

13:33:14

a Frankfurt contacts read still returned a 502, nearly two hours after the European milestone

On Friday Microsoft reported healthy aggregate availability in Europe at 11:35 UTC. **At 13:33:14 a contacts read from the location recorded as Frankfurt still returned a 502.** Its TCP phase took 21.079 ms, TLS took 49.420 ms, and the post-upload wait was 21,027.746 ms. That is nearly two hours after the European milestone, with the same narrow timing signature visible elsewhere. A regional average can improve substantially while a particular operation still fails. Different populations, protocols and definitions of health make these observations compatible.

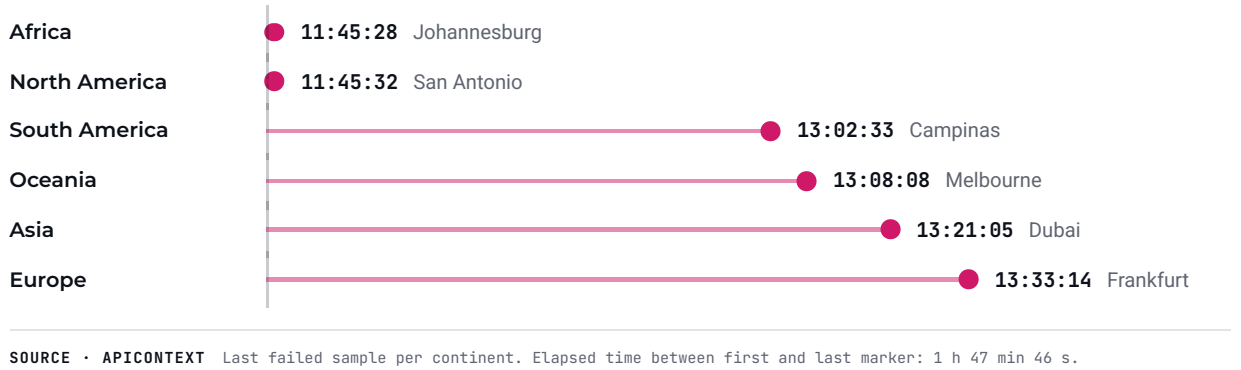
> **A regional average can improve substantially while a particular operation still fails.**

FIGURE 14

The long tail of observed Graph failures

FINAL RECORDED FAILURE PER CONTINENT · 13 NOVEMBER UTC · ELAPSED FROM 11:45:28

1 h 47 min 46 s
between the first and last marker



SOURCE · APICONTEXT Last failed sample per continent. Elapsed time between first and last marker: 1 h 47 min 46 s.

Figure 14. Regional last failures.

The last failed samples from Africa and North America occurred four seconds apart at about 11:45. The final observations from South America, Oceania and Asia followed at 13:02, 13:08 and 13:21. The span to Frankfurt was 1 hour 47 minutes 46 seconds using whole-second timestamps. It is a spread between observed failures, subject to schedule cadence. All 21 follow-up checks of affected monitors from 14:00 to the 14:41 cutoff passed. Microsoft separately placed global webmail recovery at 14:17 and the end of active impact at 16:10. Mail backlog recovery continued afterward. **A successful Graph read cannot confirm delivery of a message that was queued earlier.**

CHAPTER 06

Could an SRE have used this evidence to limit the blast radius?

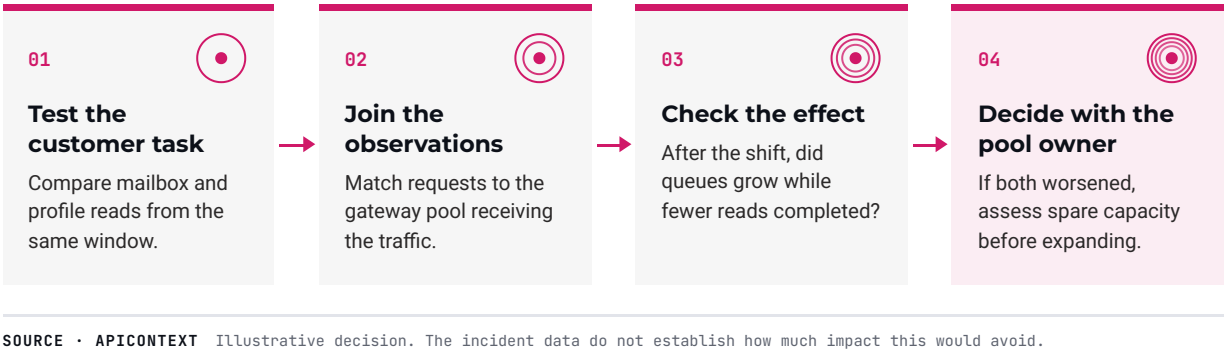
At 17:32 UTC on Thursday, Microsoft recorded that automated traffic redirection was continuing and conditions were deteriorating across multiple regions. Minutes earlier, our Toronto probe had completed a secure connection and received a 504 after a thirty-second wait. A responder deciding where to send the next batch of traffic had a concrete question: **when a pool accepts redirected traffic, do mailbox reads through that pool actually complete?** A server can accept a connection and still be a poor destination for the next thousand requests.

> **A server can accept a connection and still be a poor destination for the next thousand requests.**

FIGURE 15

Should the next traffic shift proceed?

PROPOSED SRE DECISION DURING GATEWAY RECOVERY



SOURCE · APICONTEXT Illustrative decision. The incident data do not establish how much impact this would avoid.

Figure 15. How an SRE could evaluate the next traffic shift.

To answer that question, the SRE would need the routing decision, the receiving pool, its spare capacity and a way to associate test requests with it. An external probe's city alone cannot identify that pool. With a request correlation ID or a bounded routing trial, a team could compare the results of Exchange-backed reads before and after a small traffic shift, alongside queue growth on the receiving servers. If more connections succeeded but fewer mailbox reads completed and queues grew, that would be evidence to pause the next shift. Whether pausing would protect other users depends on the ability to isolate the affected pool and on what happens to traffic that still needs service.

16:17

redirection began, more than an hour before our first observed failure

Timing matters to this counterfactual. Redirection began at 16:17, more than an hour before our first observed failure. Our measurements could have informed subsequent routing and readmission decisions. The cluster of 21-second waits emerged in the retained sample, including during recovery; an SRE acting on Thursday afternoon could use only the examples that had already arrived. Microsoft alone could determine which capacity was safe to isolate. **A reduction in the number of affected users or regions remains a hypothesis.**

A read-only Model Context Protocol (MCP) interface could expose request timestamps, operations, outcomes and phase timings to an SRE or an incident-response agent. For the first Toronto failure, the available evidence would have placed nearly all elapsed time after upload. The next diagnostic step would have been to match that request to the receiving pool and inspect its downstream connection attempts.

During recovery, the same interface could return recent successful and failed checks by operation and location, including the observation time and sample count. At 13:33 UTC the Frankfurt contacts read would have supplied a specific exception to investigate against improving regional availability. The age of each result matters because a gap in the schedule provides no evidence of recovery.

> **The age of each result matters because a gap in the schedule provides no evidence of recovery.**

A chronological replay could test the effect on diagnosis time. One team would receive the internal timeline. A second would also receive external results as each check completed. Measures would include time to identify the failing phase, time to isolate the affected operation family, and incorrect hypotheses pursued. **This study did not run that experiment.**

Critical services need a way to follow one transaction across the boundaries between teams. At Layer 3, a network engineer can investigate the IP path and packet loss if path evidence is available. At Layer 4, the Toronto probe timed the TCP connection. Between transport and HTTP, the TLS handshake shows when a secure channel became usable. The OSI model calls Layers 5 and 6 session and presentation; this HTTPS trace exposes a handshake interval, without a separate timer for each conceptual layer. At Layer 7, we know which operation ran, how long it waited and what HTTP response came back.

Across the retained traces, TCP and TLS could complete while a mailbox operation stalled. Linking those phases to the HTTP operation and outcome narrows the interval that requires investigation. Assigning that interval to a specific queue, retry or lock requires a correlation identifier or comparable internal trace. **The study establishes a reproducible external signature and a way to investigate it. The effect on diagnosis time and incident scope remains unmeasured.**



APIContext

Silent failures break businesses.

SOURCES

Provider chronology and root cause: Microsoft's final post-incident report, 20 November (public copy). External measurements: APIContext, Microsoft Graph (Office) project. All times are UTC.

apicontext.com · info@apicontext.com

© APIContext 2026