BURAK GALİBA
Thoughts on AI Transformation

ESSAY · ENTERPRISE AI

ChatGPT Went Down for Three Hours. What Would Your Process Have Done?

A 2023 ChatGPT outage went viral again this week as if it were news. The lesson still holds. One database behind login stopped most of ChatGPT for three and a half hours. What an enterprise needs is a planned degraded mode for every process that depends on AI.

AGENTIC AI | VENDORS | GOVERNANCE

Burak Galiba · November 22, 2026 · 7 min read
https://burakgaliba.com/blog/when-your-ai-vendor-goes-down

SCAN TO READ ONLINE

This weekend my feed carried news of a ChatGPT outage: three and a half hours, up to 75% of requests failing, and a database at the root of it. Several sites reported it as having happened on 21 November 2026.

It happened, but in 2023. OpenAI's own incident page dates it to 21–22 November 2023. The write-up says only "November 21st", with no year, and that was enough for it to circulate three years later as breaking news. OpenAI's status history shows no comparable incident this week.

I nearly wrote about it as current. I'm writing about it anyway, because the re-run makes a useful test. Read it as if it happened yesterday, then ask one question: what would your process have done?

What actually happened in 2023

According to OpenAI's write-up, between 1:48 and 5:21 PT a large portion of requests to OpenAI failed, peaking at a 75% failure rate on paid and free ChatGPT. The incident covered the API and ChatGPT together.

The cause was maintenance on a PostgreSQL database used by what OpenAI calls its Primary API service. That service handles user authentication and service-to-service authentication across the whole OpenAI stack. The database had two read replicas. The maintenance caused both to fail at the same time, and every request path that depended on them failed with them.

Recovery is the instructive part. Reviving the failed replicas didn't work. Bringing up replacements in other regions didn't work either. What worked was getting two new replicas running in the same region, with the help of OpenAI's service provider. OpenAI then paused similar maintenance and committed to pre-maintenance reviews, standby capacity during maintenance, and a review of the whole stack for redundancy.

That is a competent postmortem from a company that takes reliability seriously. It is also the point: this happens to competent companies.

Three things this outage tells a buyer

The weak point was nowhere you would look. No customer evaluating ChatGPT asks about the read replicas behind its login service. Yet that was the component that decided whether anyone could use the product that afternoon. Your AI vendor's architecture is part of your process now, and you can't inspect most of it.

Redundancy on paper is not redundancy. There were two replicas, which on any architecture diagram counts as resilient. Both failed together because one event caused the failure. When you read "redundant" in a vendor's security pack, the question is not how many copies there are. It's what would take all of them down at once.

Partial is harder than total. A 75% failure rate means one request in four still works. Users retry, integrations retry, and some work completes while most doesn't. A clean outage tells everyone to stop. A partial one leaves you guessing which outputs actually finished, and that is the version your

process has to handle.

In 2023 people waited. In 2026 processes stop.

In November 2023, most enterprise use of ChatGPT was a person typing into a browser. When it failed, that person waited, swore, and did the task by hand. The outage cost an afternoon of individual productivity.

That isn't where enterprise AI sits now. Models are wired into ticket triage, document intake, customer replies and internal agents that act on systems. When the model behind one of those stops answering, nobody is sitting there to notice and switch to doing it by hand. The queue builds up, and whatever the process feeds downstream waits too.

The same three and a half hours that used to cost some productivity can now stop a process. Nothing about the vendor changed. What changed is how much of your operation sits behind their login.

Every AI-dependent process needs a degraded mode

The fix isn't to avoid AI vendors, and it isn't to demand an uptime figure nobody can deliver. It's to decide in advance, for every process that depends on a model, what happens when the model isn't there.

I'd make every such process choose one of four degraded modes, and write the choice down at launch, next to the business case:

- **Wait.** The work queues and resumes when the model returns. Fine for anything that can be hours late without harm. Name the longest delay that is acceptable, so "wait" has a limit.
- **Fall back to people.** The work routes to a human queue. This only works if the people still know how to do it and have the capacity. A process that has run on AI for a year may have neither.
- **Fall back to another model.** A second provider or a self-hosted model takes over. It's the most resilient option and the most expensive one. The next section covers what it really involves.
- **Fail closed.** The process stops and says so. That's the right choice when a wrong or partial answer is worse than none, as with anything that acts on money, access or customer records.

The choice is not the hard part. The hard part is that most processes I'd expect to find have never had it made. Their degraded mode is whatever happens by accident, and they find out what that is during the outage.

A second model is a second product, not a switch

"We'll just fail over to another provider" is the answer I'd most expect in a planning meeting. It's also the one most likely to fail when it's needed.

A fallback model is a second product to run. Prompts tuned for one model behave differently on another. The evaluation set has to pass on both, which means maintaining both. And the second provider comes with its own contract, retention terms and data rules. If nobody negotiated those, then during an outage your data goes out on terms nobody reviewed. That is the shadow AI problem recreated by your own architecture.

A tested fallback is worth having for processes where stopping is expensive. An untested one is a line in a slide. The test is simple: has it carried real traffic in the last quarter?

Test the degraded mode before the vendor does

A degraded mode written on paper is a guess. The only way to know it works is to use it on purpose.

Once a quarter, on a scheduled day, cut each critical process off from its model for an hour. Watch what happens. Does the queue hold? Does the human fallback know it's on? Does the alternate model produce answers that pass the evaluation set? Does anything downstream quietly break?

The first time, things will go wrong. That's the point of doing it on a Tuesday morning you chose rather than one the vendor chose for you.

Three other things I'd put in place alongside it:

- **Your own monitoring.** A vendor's status page reports the incident after the vendor has confirmed it. Your process should notice before that, from its own error rate.
- **A named owner per process.** The same owner I argued every shipped agent needs decides when to switch to degraded mode and when to switch back. In an outage, "someone should decide" means nobody does.
- **A realistic reading of the SLA.** Service credits refund part of your subscription. They don't cover a queue that sat for an afternoon or replies that went out late. Price the risk yourself.

The other lesson in the re-run

This week also showed a second risk, and it's smaller but more common. An undated postmortem became a 2026 news story because nobody checked the date against the source. Content pipelines that summarize and republish without checking are the fluent-but-unverified failure I worry about in enterprise AI, run in public.

The discipline is the same in both cases. Find the primary source, check what it actually says, and design for the day it isn't there.

Where to start

List every process where a model call sits on the critical path. Next to each, write its degraded mode: wait, people, another model or fail closed. Then mark which of those choices has actually been tested.

If that column is empty, you've found your next quarter's work. The good news from 2023 is that you can rehearse this with nothing at stake. The 2023 outage hands you the scenario: three and a half hours, with three requests in four failing.

If you want to see how this sits against the rest of your setup, I built a free AI readiness assessment. It's twenty questions, and one of them asks exactly this: if your primary AI vendor walked away tomorrow, what would happen?

Read the essay online at https://burakgaliba.com/blog/when-your-ai-vendor-goes-down

I built a free AI Readiness Score — 20 questions across pilots, data, talent and governance, about ten minutes, no email required: https://burakgaliba.com/readiness

© 2026 Burak Galiba · https://burakgaliba.com · Views are my own.