Engineer Software

Built fail-closed rate limiting and anti-abuse

Built fail-closed abuse controls — 22 Redis-backed rate limiters, Cloudflare Turnstile, request idempotency and an origin lock — so the platform sheds bots and floods instead of trusting its callers.

2025

22

rate limiters that fail closed

Situation. A public directory of companies and professionals is a target the day it goes live. Scrapers want the data, spam accounts want the reach, and an endpoint that costs the platform real money to serve — search, export, anything that touches an external API — is worth abusing simply because it is free to call. None of that is malice aimed at this platform in particular; it is background weather on the open internet.

Task. The expensive and abusable paths needed limits that hold under pressure, including the pressure of the limiter's own dependency being unavailable.

Action. There are 22 rate limiters, each constructed for the path it protects rather than one global cap, because a login attempt, a search and a bulk export are abusive at wildly different rates. State lives in Redis so a limit is shared across instances instead of being per-process and trivially escaped. The important decision is what happens when Redis is not there: the limiters fail closed. Traffic is refused rather than waved through, which is the less convenient answer and the only defensible one. Around them sit Cloudflare Turnstile on the paths worth challenging, 351 lines of idempotency middleware so a retried write does not become two, an origin lock that refuses requests not arriving through the front door, and a bespoke challenge on the directory itself.

Result. Abuse gets expensive for the abuser and cheap for the platform, and an outage in the limiter's own store degrades into refusal rather than into an open door. Failing closed does mean a Redis problem becomes a user-visible problem — accepted deliberately, because the alternative is a Redis problem becoming a billing one.

Bring this experience to your project

Use "Built fail-closed rate limiting and anti-abuse" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. The part users never see: where the data lives, what the rules are, and how other systems are allowed to talk to it.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Built fail-closed rate limiting and anti-abuse — 22 rate limiters that fail closed

Explore our services Get in touch →

Getting hammered by bots and scrapers?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Security Engineer. It draws on Backend Engineering, APIs & Integration, Reliability & Backups, Networking & VPN, Security and Performance Tuning. And while you are here, we are rather good at Backend & API Development, Site Reliability & Monitoring, Security & Access Management and AI Agent Security.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Getting hammered by bots and scrapers?

Ask about the decisions behind “Built fail-closed rate limiting and anti-abuse”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://software.engineer.company/portfolio/built-fail-closed-rate-limiting-and-anti-abuse/