Engineer Software

Built email with failover and delivery logging

Built email as a platform capability — three providers with failover, delivery webhooks, send and delivery logging, templating and campaigns — behind a startup check that will not boot without one.

2025

mail that will not start without a way to send

Situation. Email carries a lot of weight on the platform — verification, notifications, digests, campaigns, the things a user actually waits for. And a mail provider is exactly the kind of dependency that fails quietly: the config looks fine, the app boots, and you only find out something's broken when a real person never gets the message they were promised. That's the worst way to learn about it. One provider makes it worse, because the failure is total and someone else's to fix.

Task. Email had to be treated as a capability the platform owns rather than a client library it calls — able to survive a provider outage, able to say what happened to a given message, and loud at startup in the environments where silence is dangerous.

Action. Three providers sit behind one interface — SendGrid as primary, with SMTP2GO and Azure Communication Services behind it — and failover between them is automatic rather than a configuration change made under pressure. Delivery is not assumed: inbound webhooks report what each provider did with a message, and both sides are recorded, in a send log and a delivery event table, so "did this person get their verification mail" is a query rather than a guess. Templating keeps the message bodies out of the code, and a separate broadcast schema — 5 tables and 24 functions — carries campaigns to segments of users, which is a different problem from transactional mail and was built as one. In front of all of it, a startup check sends a real message through the stack, behind a flag: in development it logs a warning and carries on, because nobody wants their laptop refusing to start over an expired sandbox key, and in staging

and production a failure is fatal and the process exits rather than deploy a build that cannot send mail. The send path itself goes through an SSRF-protected client with a 30-second timeout, and the async delivery path has retries and backoff so a momentary blip doesn't drop a message.

Result. A whole category of silent failure moved from "a user notices days later" to "the deploy stops", and a provider having a bad afternoon became a degraded path rather than an outage. The cost is three integrations to keep working instead of one, and delivery logs that grow and have to be pruned — both accepted, because email is the channel the platform cannot route around.

Bring this experience to your project

Use "Built email with failover and delivery logging" 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 email with failover and delivery logging — mail that will not start without a way to send

Explore our services Get in touch →

Do your emails actually reach people?

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: Backend Engineer. It draws on Backend Engineering, APIs & Integration, Cloud, Reliability & Backups and Security. And while you are here, we are rather good at Backend & API Development, Site Reliability & Monitoring and Security & Access Management.

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 →

Do your emails actually reach people?

Ask about the decisions behind “Built email with failover and delivery logging”. 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-email-with-failover-and-delivery-logging/