Engineer Software

Moved slow work onto a River job queue

Moved slow work off the request path onto a River job queue — 15 worker modules, 8 scheduled tasks and 20 pg_cron jobs — so a request returns while the work behind it carries on.

2025

15

worker modules keeping the request path quick

Situation. Some work has no business happening while a user waits. Sending mail, rebuilding a search index, generating a document, recalculating standings — do any of it inside the request and the user watches a spinner for something they never asked to see. Do it in a goroutine instead and it disappears the moment the process restarts, which it will, mid-deploy, with no record that it was ever supposed to happen.

Task. Background work needed somewhere durable to live: a queue that survives a restart, retries a failure, and can be looked at when something has not happened.

Action. River was the choice, largely because it keeps its queue in PostgreSQL — the database is already the system of record, so a job and the rows it touches commit or roll back together, and there is no second piece of infrastructure to run and reason about. There are 15 worker modules behind it and 8 scheduled tasks. Underneath, 20 pg_cron jobs handle the maintenance the database is better placed to do itself: pruning partitions, rotating salts, refreshing aggregates. Anything slow enough to notice was moved off the request path and onto one of the two.

Result. Requests return quickly and the slow work still finishes, with retries and a visible history when it does not. Keeping the queue in Postgres rather than a dedicated broker is a deliberate limit: it will not scale forever, and at some

volume it becomes the wrong answer. For a platform whose bottleneck is the database anyway, one less moving part was worth more than headroom that was not going to be used.

Bring this experience to your project

Use “Moved slow work onto a River job queue” 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. Moved slow work onto a River job queue — 15 worker modules keeping the request path quick

Explore our services Get in touch →

Is slow work holding up your users?

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, Databases, PostgreSQL, Automation & CI/CD, Reliability & Backups and Performance Tuning. And while you are here, we are rather good at Backend & API Development, AI Agent Tools & MCP Servers, Database Design & Modeling and Site Reliability & Monitoring.

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 →

Is slow work holding up your users?

Ask about the decisions behind “Moved slow work onto a River job queue”. 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/moved-slow-work-onto-a-river-job-queue/