Engineer Data

Know the licence of everything you ship

Audited 644 Rust crates for licence compatibility on every build, and proved the check fires by rewriting one crate's licence and by moving the pinned core revision without regenerating.

September 2026

From dependencies to a licence-policy check

Logical overview of the build check against declared licence metadata. Metadata can be wrong or incomplete; the check is not legal advice.

1.

Resolve dependencies

The audit resolves the complete transitive dependency graph for the pinned core revision.

2.

Evaluate declared licences

Each declared licence expression is evaluated against the project's explicit allow list, respecting AND and OR conditions.

3.

Fail on a mismatch

An unaccepted expression or a stale audit fails the build. Deliberately introduced failures verify that the check can detect both.

644

Rust crates checked for licence on every build

Situation. Linking a Rust core into a shipped application means shipping everything that core depends on. The dependency graph is 644 crates. Each carries a licence, some carry more than one, and a single copyleft crate arriving three levels down through a routine version bump changes what the application as a whole may be distributed under — silently, in a lock file nobody reads line by line.

Task. The licence position had to be verified on every build rather than reviewed once, with an explicit allow list, so that a change in the graph is a build failure and not a discovery made later by someone else.

Action. The audit resolves the full transitive graph and checks each crate's licence expression against a list of terms the project accepts, evaluating the boolean expressions properly — a crate offering a choice of two licences is acceptable if either is on the list, and a crate requiring both is only acceptable if both are. Anything unmatched fails the build rather than warning, and adding a term to the allow list is a deliberate edit with a reason. A check that has never failed is indistinguishable from one that cannot fail, so it was made to fail on purpose, twice. Once by rewriting a crate's licence expression to something the list does not accept, which is the shape of a crate changing its terms upstream between versions — the case no human review catches, because the crate itself is not new. Once by moving the pinned core revision without regenerating the audit, which is the shape of a core bump quietly bringing a new dependency with it. Both provocations failed the build as intended, and the check was left in place rather than widened.

Result. The distribution position is checked on every build, and the check is known to work because it was made to fail rather than because it has never spoken. Before it existed, the bundle, the licence file and the project's own README all made a claim about 644 crates that nothing had verified. The limit is precise and should not be overstated: the check reads the declared licence metadata, and metadata can be wrong or incomplete. It proves nothing about a crate that misdeclares itself, and it is not legal advice.

Bring this experience to your project

Use "Know the licence of everything you ship" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. Rules and checks that catch bad data on the way in, and a clear answer to who owns which figure and where it came from.

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. Know the licence of everything you ship — 644 Rust crates checked for licence on every build

Explore our services Get in touch →

Do you know the licence of everything you ship?

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: Build and Release Engineer. It draws on Data Governance, DevOps, Automation & CI/CD, Security and Documentation. And while you are here, we are rather good at Data Governance & Quality, DevOps & CI/CD Automation 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

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Do you know the licence of everything you ship?

Ask about the decisions behind “Know the licence of everything you ship”. 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://data.engineer.company/portfolio/know-the-licence-of-everything-you-ship/