Engineer

Conocer la licencia de todo lo que se publica

Audité la compatibilidad de licencias de 644 crates Rust en cada compilación y probé que el control se activa al cambiar la licencia de una crate y al modificar la revisión fijada del núcleo sin regenerar los datos.

septiembre 2026

De las dependencias a la comprobación de la política de licencias

Descripción lógica de la comprobación de compilación basada en los metadatos de licencia declarados. Estos metadatos pueden ser erróneos o incompletos; la comprobación no constituye asesoramiento jurídico.

1.

Resolver las dependencias

La auditoría resuelve el grafo completo de dependencias transitivas para la revisión fijada del núcleo.

2.

Evaluar las licencias declaradas

Cada expresión de licencia declarada se evalúa frente a la lista explícita de licencias permitidas del proyecto, respetando las condiciones AND y OR.

3.

Detener la compilación ante una discrepancia

Una expresión no aceptada o una auditoría desactualizada hace fallar la compilación. Los fallos introducidos deliberadamente comprueban que la verificación puede detectar ambos casos.

644

crates Rust con licencia verificada en cada compilación

Situación. Enlazar un núcleo Rust dentro de una aplicación distribuida significa distribuir todo aquello de lo que depende ese núcleo. El grafo de dependencias tiene 644 crates. Cada uno lleva una licencia, algunos llevan más de una, y un único crate copyleft que llegue tres niveles más abajo mediante una actualización rutinaria de versión cambia bajo qué condiciones puede distribuirse la aplicación completa: silenciosamente, en un archivo de bloqueo que nadie lee línea por línea.

Tarea. Había que verificar la situación de las licencias en cada compilación y no revisarla una vez, con una lista explícita de licencias permitidas, para que un cambio en el grafo sea un fallo de compilación y no un descubrimiento posterior de otra persona.

Acción. La auditoría resuelve el grafo transitivo completo y comprueba la expresión de licencia de cada crate frente a una lista de términos que acepta el proyecto, evaluando correctamente las expresiones booleanas: un crate que permite elegir entre dos licencias es aceptable si cualquiera está en la lista, y uno que exige ambas solo lo es si están las dos. Todo lo que no coincide hace fallar la compilación en lugar de producir una advertencia, y añadir un término a la lista permitida es una modificación deliberada con un motivo. Una comprobación que nunca ha fallado es indistinguible de otra que no puede fallar, así que se la hizo fallar a propósito dos veces. Una, reescribiendo la expresión de licencia de un crate con algo que la lista no acepta, lo que reproduce un crate que cambia sus condiciones en origen entre versiones: el caso que ninguna revisión humana detecta, porque el propio crate no es nuevo. Otra, cambiando la revisión fijada del núcleo sin regenerar la auditoría, lo que reproduce una actualización del núcleo que incorpora silenciosamente una nueva dependencia. Ambas provocaciones hicieron fallar la compilación como se pretendía, y la comprobación se mantuvo en lugar de ampliarse para aceptarlas.

Resultado. La situación de distribución se comprueba en cada compilación, y se sabe que el control funciona porque se le hizo fallar, no porque nunca haya dicho nada. Antes de que existiera, el paquete, el archivo de licencias y el propio README del proyecto hacían una afirmación sobre 644 crates que nada había verificado. El límite es preciso y no debe exagerarse: la comprobación lee los metadatos de licencia declarados, y los metadatos pueden ser erróneos o incompletos. No demuestra nada sobre un crate que se declara incorrectamente, y no es asesoramiento jurídico.

Aplique esta experiencia a su proyecto

Utilice «Conocer la licencia de todo lo que se publica» como punto de partida para un reto similar en su organización. Los servicios que respaldan este trabajo están disponibles para su proyecto. Reglas y comprobaciones que detectan datos incorrectos al entrar y aclaran quién responde de cada cifra y de dónde procede.

Explore nuestros servicios Póngase en contacto →

Mire más allá del resultado

Lea la situación, las decisiones y el resultado de este ejemplo. Podemos hablar de qué partes se adaptan a sus restricciones y cuáles necesitan otro enfoque. Conocer la licencia de todo lo que se publica — 644 crates Rust con licencia verificada en cada compilación

Explore nuestros servicios Póngase en contacto →

¿Conoce la licencia de todo lo que publica?

Converse sobre su proyecto con la ingeniera responsable de la entrega. Acordamos por escrito el alcance, la base del precio y el traspaso antes de empezar a trabajar.

Contexto de la función: Ingeniera de compilación y versiones. Se apoya en Gobernanza de datos, DevOps, Automatización y CI/CD, Seguridad y Documentación. Y ya que está aquí, también se nos da bien Gobernanza y calidad de datos, DevOps y automatización CI/CD y Seguridad y gestión de accesos.

Póngase en contacto Explore nuestros servicios

Una parte de **una trayectoria más amplia** — y toda ella está en este sitio.

Cada entrada sigue la misma estructura: la situación, la tarea, lo que hicimos y lo que cambió.

Vea la trayectoria completa

Defina la mejora que necesita

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

¿Conoce la licencia de todo lo que publica?

Pregunte por las decisiones detrás de «Conocer la licencia de todo lo que se publica». Hablará con las personas que pueden explicar el trabajo y valorar su propio proyecto.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/portfolio/know-the-licence-of-everything-you-ship/