EBOOK

Your Roadmap to Cyber Resilience Act Readiness

The EU CRA Clock Is Already Running



Copyright 2026 JFrog Ltd.

Table of **Contents**

Introduction 1

Chapter 1: The CRA Countdown & Market Reality 2

- Secure-by-Design Is Now Backed by Law 3

Chapter 2: Why Legacy Compliance Fails the CRA Test 4

- The Snapshot Fallacy 4
- Outpaced Before It Starts 4
- No Single Chain of Evidence 5

Chapter 3: Enter DevGovOps 6

- Governance Across the Full Lifecycle 6
- Segregation of Duties as Code 7

Chapter 4: Architecting Trust with JFrog AppTrust 8

- Evidence Bound to Every Artifact 8
- Blast Radius Visibility 9

Chapter 5: The Execution Roadmap 10

- Phase 1: Discovery & Inventory 10
- Phase 2: Policy Translation 10
- Phase 3: Automated Gating & Attestation 11
- Phase 4: Audit-Ready by Default 11

Conclusion 12

- Start Your CRA Readiness Journey 12

2026 JFrog Ltd. All rights reserved.

Introduction



A critical zero-day hits a library buried deep in your stack. Your organization now has 24 hours to advise EU authorities. Can you name every affected product, version, and owner before the clock runs out?

Since early September 2026, that question is no longer hypothetical. The EU Cyber Resilience Act's reporting obligations are live. They apply to products already on the market, and they arrive more than a year before the Act's full requirements take effect in December of 2027. After that date, products placed on the EU market must demonstrate CRA conformity and carry the CE marking. Products already on the market before that date come into scope when they undergo a substantial modification.

"Secure-by-Design" is now law. The organizations that keep their EU market access will be the ones that replace manual audits with continuous, automated governance. This playbook shows how to do it with policy-as-code, automated pipeline gating, and cryptographic traceability on the JFrog Platform, so every release is audit-ready by default.

2026 JFrog Ltd. All rights reserved.

CHAPTER 01

The CRA Countdown & **Market Reality**



As of September 11, 2026, manufacturers must report actively exploited vulnerabilities and severe incidents through the CRA single reporting platform maintained by ENISA. An early warning is due within 24 hours of becoming aware of the issue, and a full notification within 72 hours. A final report follows within 14 days of a fix becoming available, or within one month for severe incidents. These obligations cover products already on the EU market, not just new releases, so if your detection and reporting workflows aren't operational today, you are already at risk of missing compliance obligations.

The second deadline, December 11, 2027, determines market access. From that date, every product with digital elements placed on the EU market must meet the CRA's essential requirements and carry the CE marking to prove it. While the deadline might seem distant, the clock is ticking - conformity depends on processes and evidence that take time to build. If those are not in place, market surveillance authorities can prohibit non-compliant products from entering the EU market, order them withdrawn, or demand a full recall.

2026 JFrog Ltd. All rights reserved.

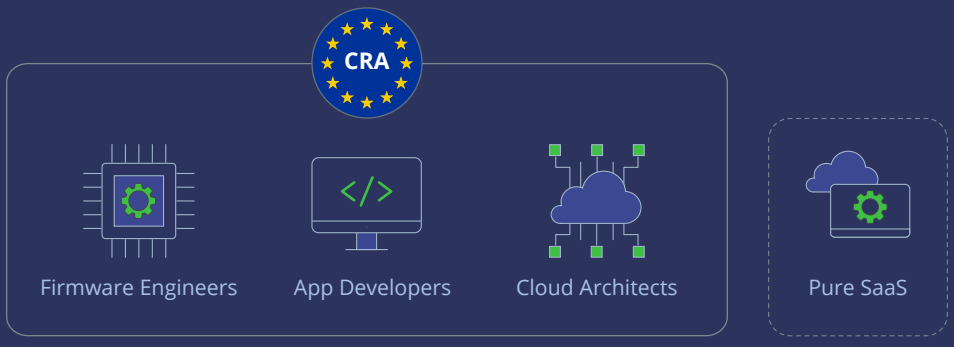
Secure-by-Design Is Now Backed by Law

The CRA turns "Secure-by-Design" from a marketing phrase into an obligation that spans the entire lifecycle, from design, development, and testing through release and years of post-release support. Ad hoc measures won't satisfy regulators. Security must be demonstrable at every stage, backed by evidence you can produce on request.

That obligation reaches almost every company's systems. The CRA covers a broad range of hardware and software products that connect to a device or network, including operating systems, IoT devices, B2B software, and mobile apps. Fines for the most serious breaches, including missed vulnerability reports, reach €15 million or 2.5% of total worldwide annual turnover for the preceding financial year, whichever is higher. At that scale, CRA readiness belongs to the board, not just the engineering floor.

Architect's View: The Scope of "Digital Elements"

If your product is intended to connect directly or indirectly to a device or network, it is likely in scope, and worth confirming with your own advisers. That puts firmware engineers, application developers, and cloud architects under the same regulation. Pure SaaS generally falls outside the CRA, but remote data processing solutions that the manufacturer designs or develops and that the product needs to perform one of its functions are covered.



2026 JFrog Ltd. All rights reserved.

CHAPTER 02

Why Legacy Compliance **Fails the CRA Test**



Most compliance programs were built for a slower era of regulation, when annual audits and manual sign-offs could keep up with the pace of change. At modern release velocity, those same practices create gaps that regulators will find. They also leave teams unable to produce evidence when it matters most. In 2026, the JFrog 2026 Software Supply Chain Security State of the Union found that 59% of organizations report full provenance visibility in production, while 48% say they need a week or more to generate compliance audit proof per application, with 10% taking a month or more.

The Snapshot Fallacy

Traditional audits rely on point-in-time assessments that prove a system was secure on Tuesday. By Wednesday, a new commit or a newly disclosed CVE can make that proof obsolete. The CRA expects conformity throughout a product's support period, and a static PDF report can't keep pace with software that changes daily.

Manual processes also slow delivery. When compliance depends on human approvals, security teams become a tollbooth on the path to production, and releases increasingly queue while tickets wait for sign-off. Over time, developers learn to route around the process, which undermines the very controls it was meant to enforce.

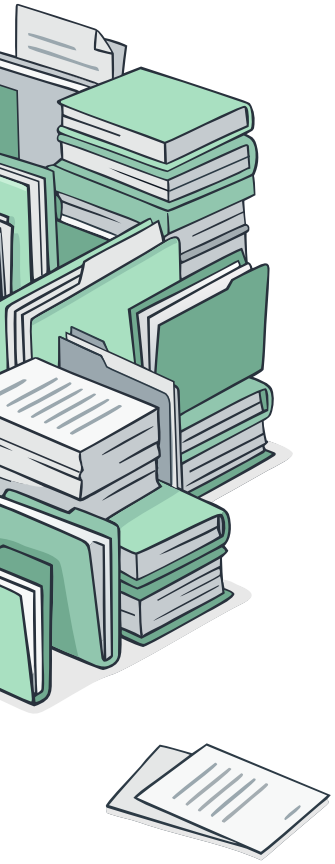
2026 JFrog Ltd. All rights reserved.

Outpaced Before It Starts

AI now generates an increasing share of the software entering production, and review processes built for human commit volume are already falling behind. The deeper issue isn't speed, though. It's memory. An AI agent that plans, generates, and deploys code keeps no durable record of why it made each decision, which components it pulled in, or what it changed along the way.

The CRA doesn't mention AI or agents, but its obligations don't depend on how code or assets are produced. The manufacturer is accountable for every component it integrates into the product and must exercise due diligence on third-party components and document its risk assessments. When an agent leaves no trace, that accountability has nothing to stand on. Without a system of record for what AI contributes, agentic development makes it harder to produce the evidence the CRA expects.

No Single Chain of Evidence



Modern software is often built across disconnected toolchains, with code in GitHub, tickets in Jira, and builds spread across multiple CI/CD runners. Without a single source of truth for binaries, auditors get a maze of disjointed logs instead of a verifiable chain from code to cloud. Each gap in that chain is a question you may not be able to answer.

The SBOM is meant to close part of that gap, but a static SBOM goes stale quickly and is easy to falsify. Unless it is cryptographically bound to the artifact it describes, it proves very little about what you actually shipped. That leaves a supply chain blind spot that won't survive regulatory scrutiny.

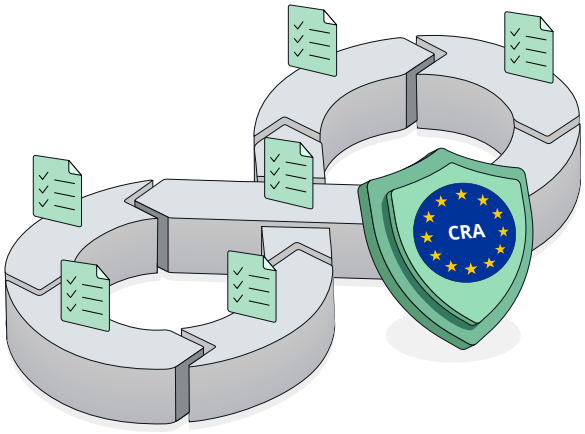
Practitioner's Corner: The Cost of Fragmentation

When an auditor asks for evidence on a single release, teams often spend weeks pulling records from Git, ticketing systems, and standalone scanners. The process is slow, error-prone, and incompatible with a 24-hour reporting window.

2026 JFrog Ltd. All rights reserved.

CHAPTER 03

Enter **DevGovOps**



Surviving the CRA requires a strategic shift: uniting Development, Governance, and Operations into a single automated workflow known as DevGovOps. Instead of treating compliance as an afterthought handled by an external audit team, DevGovOps builds governance into the software delivery process itself. Compliance becomes an automated layer that runs with every build, with no separate process bolted on at the end.

In practice, this means continuous governance. Policy-driven checks at every pipeline stage replace spreadsheets and manual checklists. When code violates a policy, the system flags it in real time, while the fix is still cheap and the context is still fresh.

Governance Across the Full Lifecycle

Security has to start in the developer's IDE, where shift-left practices catch vulnerabilities early. Shift-left alone isn't enough under the CRA, though. The regulation mandates a support period of at least five years, or the product's expected use time if shorter. Each security update issued during that period must then stay accessible to users for a minimum of ten years, or the remainder of the support period, whichever is longer. DevGovOps therefore shifts right as well, extending continuous monitoring and governance to software already running in the field and ongoing software lifespans.

Continuous Post-Release Governance shows what shifting right looks like in practice. Policy enforcement doesn't stop at the release gate. It continues monitoring each released version after deployment, so a newly disclosed CVE in a component you shipped six months ago triggers the same governance response as those caught before release.

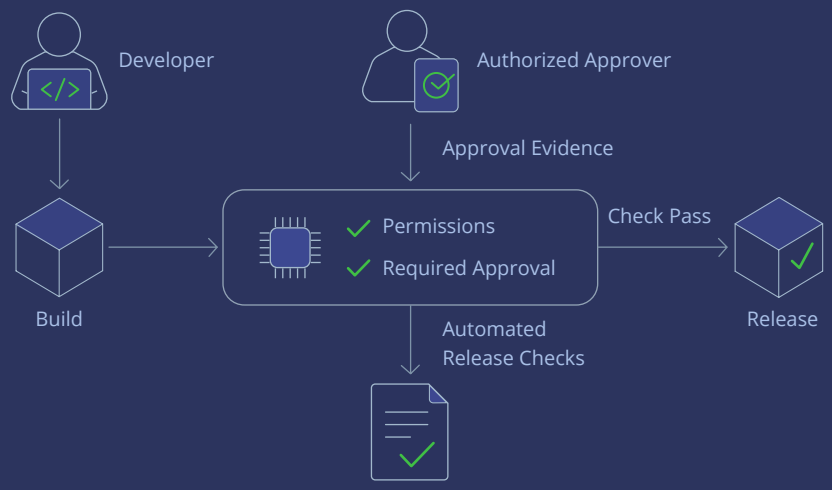
2026 JFrog Ltd. All rights reserved.

Segregation of Duties as Code

The CRA requires manufacturers to apply effective and regular security testing and reviews to their products. One practical way to evidence those reviews is Segregation of Duties, which ensures the person who writes the code isn't the person who approves its release. Traditionally, organizations enforced this through manual IT ticketing, which added delay without guaranteeing compliance. DevGovOps codifies those approvals into the pipeline, so approval requirements can be applied consistently across releases.

Architect's View: Enforcing SoD as Code

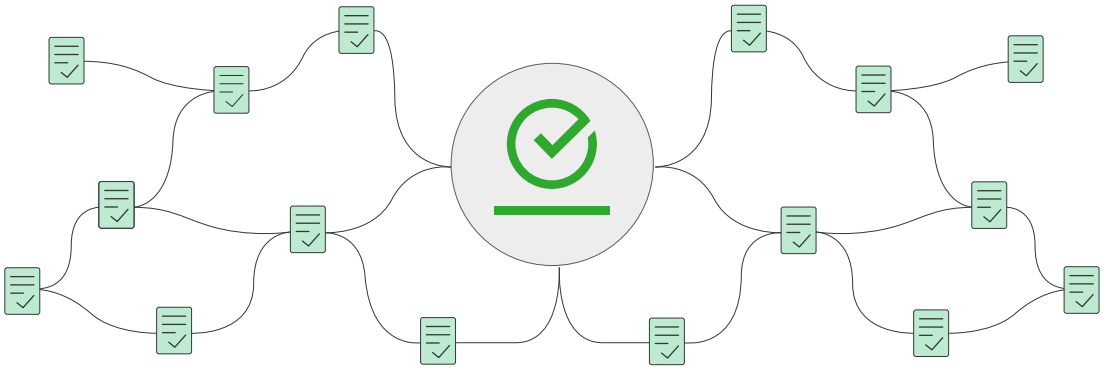
Tie role-based access control (RBAC) to pipeline automation to make it much harder for an individual developer to bypass a security gate. When approval workflows are written into the pipeline, unapproved deployments are much harder to push through, and access-control evidence is generated automatically.



2026 JFrog Ltd. All rights reserved.

CHAPTER 04

Architecting Trust with **JFrog AppTrust**



Translating strategy into execution requires the right platform. JFrog AppTrust maps dense regulatory requirements directly onto your automated workflows, starting with policy-as-code. Security leaders define exactly what a compliant release looks like, including vulnerability severity thresholds, allowed licenses, and required approvals. The platform then enforces those rules automatically across your pipelines.

Evidence Bound to Every Artifact

Every action in the build process matters to an auditor. JFrog binds security scans, pull request approvals, and SBOMs directly to each artifact as signed evidence. For agents using the JFrog agent plugins, this creates a tamper-evident chain that can run from the original prompt to production, whether a developer or an AI agent generates the first commit. When a release reaches the end of the pipeline, it carries verifiable evidence of the checks and approvals it passed.

That evidence only matters if it's enforced. Dashboards flag risk, but cannot physically stop non-compliant software from shipping. Because JFrog is built to be the system of record where binaries live, it's automated gating can actively block any release missing required attestations.

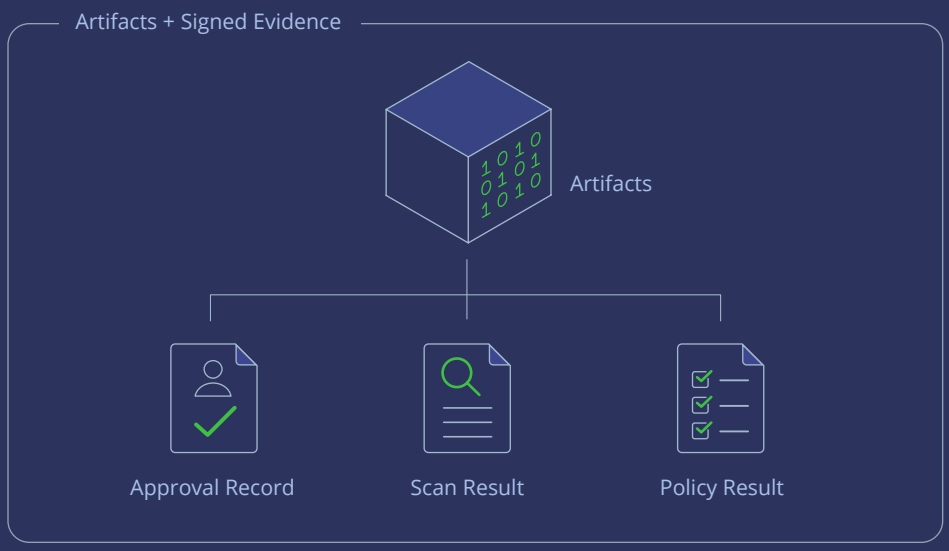
2026 JFrog Ltd. All rights reserved.

Blast Radius Visibility

The CRA's 24-hour early-warning window leaves no time for manual investigation. When a new CVE drops, JFrog shows which products, environments, and owners are affected. Security teams can begin remediation as quickly as possible and work toward filing their early warning within the window, with the supporting data already in hand.

Practitioner's Corner: Evidence That Travels with the Artifact

JFrog attaches signed metadata to immutable binaries in Artifactory. When an auditor questions a production release, you retrieve the artifact and its evidence together. That record shows who approved the release, which tools scanned it, and which policies it passed.



CHAPTER 05

The Execution Roadmap

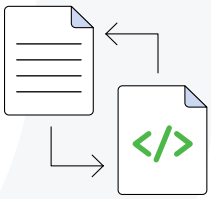
The CRA's obligations are fixed. Reporting requirements, including the 24-hour early warning for actively exploited vulnerabilities, are already in force, and full conformity is required by December 2027. Building the automation to meet these standards reliably and at scale takes time.

The roadmap below breaks the journey into four phases, moving from visibility to policy, then to enforcement, and finally to continuous audit readiness. Each phase builds on the last, so the work you do today to meet the reporting obligations already in force lays the foundation for conformity by December 2027.



Phase 1: Discovery & Inventory

You can't secure or report on what you can't see. Consolidate binary management in JFrog and generate dynamic SBOMs for every build. This inventory is the foundation for everything that follows, and your reporting obligations already depend on it.



Phase 2: Policy Translation

Start from JFrog AppTrust's Out-of-the-Box Compliance Frameworks, which pre-map many CRA controls to policy-as-code rules, so AppSec and engineering teams aren't translating articles into rules from scratch. Where your risk profile calls for it, customize thresholds and exceptions in Rego, the policy language behind Open Policy Agent (OPA) and test policies against your real evidence to validate where pipelines stand before turning on enforcement.

2026 JFrog Ltd. All rights reserved.



Phase 3: Automated Gating & Attestation

Move from policy definition to active enforcement. Attach cryptographic evidence to artifacts and configure release gates designed to prevent promotion when required attestations are missing or non-compliant with customer-defined policies.

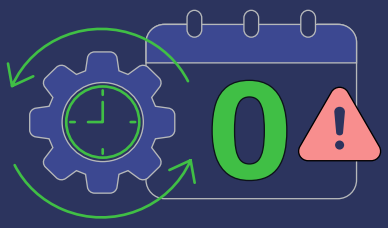


Phase 4: Audit-Ready by Default

Stress-test your audit trails and provenance records well before December 2027. Run regular drills against the 24/72-hour reporting workflow that is already in force, to confirm your blast radius analysis works under pressure.

Architect's View: Tabletop Exercises for the CRA

Simulate an actively exploited zero-day in a core open-source library. Give your team two hours to identify every affected application in JFrog, block new vulnerable builds, and draft the early-warning report. If they can do it in two hours, you have useful evidence that your 24-hour workflow can hold up under pressure.



2026 JFrog Ltd. All rights reserved.

Conclusion



The Cyber Resilience Act is a forcing function for the entire software industry, and part of it is already in force. Organizations that rely on manual compliance risk losing access to one of the world's largest markets. Organizations that adopt DevGovOps on the JFrog Platform can build much of their compliance evidence into how they ship, turning it into a competitive advantage.

Start Your CRA Readiness Journey

The reporting clock is already running. See how JFrog AppTrust turns CRA controls into automated policy, signed evidence, and release gates that start on day one.

Book a Demo **Take a Tour**

Important Notice: This ebook is provided for general informational purposes only. It reflects information available as of the publish date, and JFrog may not update it. It is provided 'as is,' without warranty of any kind, and is not legal, compliance, or security advice. Your use of JFrog products is governed by your agreement with JFrog.

This document is provided for general informational purposes only. It is not legal, compliance, or security advice and does not guarantee any security outcome. Results depend on your environment and configuration. Product capabilities vary by edition and deployment model and are governed by your agreement with JFrog.

JFrog is a registered trademark of JFrog Ltd. in the United States. GitHub, Jira, and Open Policy Agent are trademarks of their respective owners. All other trademarks referenced in this document are the property of their respective owners. Use of these names does not imply any affiliation with or endorsement by them.

JFrog empowers thousands of DevSecOps organizations globally to build, secure, distribute, and connect any software artifact to any environment using the universal, hybrid, multi-cloud JFrog Platform.