THE LINUX FOUNDATION | Europe

THE LINUX FOUNDATION | Research

Pathways to Cybersecurity Best Practices in Open Source

How the Civil Infrastructure Platform, Yocto Project, and Zephyr Project are Closing the Gap to Meeting the Requirements of the Cyber Resilience Act

March 2025

Mirko Boehm, PhD, The Linux Foundation
Hilary Carter, The Linux Foundation
Cailean Osborne, PhD, The Linux Foundation

Foreword by Miriam Seyffarth,
Open Source Business Alliance

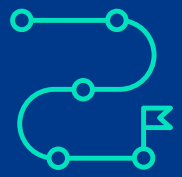
SPONSORED BY:

CIVIL INFRASTRUCTURE PLATFORM · OpenSSF OPEN SOURCE SECURITY FOUNDATION · yocto PROJECT · Zephyr™

The **CRA** introduces regulatory **oversight on products with digital elements (PDEs)**, with important implications for OSS development across stakeholder groups.

The CRA defines a new role of **OSS steward**, which are organisations that systematically support the development of open source technologies which they do not monetize.

Under the CRA, **OSS stewards** are responsible for **cybersecurity policy, processes for handling & reporting vulnerabilities**, cooperation with **MSAs**, & voluntary **security attestations.**

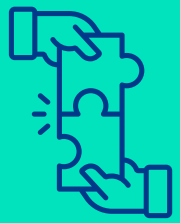


Open source projects need to establish a **5-year security roadmap**, investing in PSIRT teams & security policies to prepare for CRA timeline requirements.

Standardized security tooling accelerates CRA compliance, with **SPDX 3.0, OpenSSF Scorecard, & OpenChain** frameworks helping projects implement security best practices.

Semantic versioning helps manufacturers track CRA compliance by mapping substantial modifications, minor updates, & bug fixes to clear versioning.

SBOMs must provide greater granularity, as file-level tracking improves security visibility, risk assessment, & vulnerability response for manufacturers.



Open source **security requires cross-industry collaboration**, where manufacturers, governments, & projects co-develop policies, fund security, & ensure long-term software maintenance.

The **CRA** presents **an opportunity** to strengthen OSS security by improving security practices, documentation, and collaboration across the ecosystem.

AI introduces new security risks, requiring frameworks to mitigate threats from AI-generated code & poisoned training datasets.

Leadership drives open source resilience, as project maintainers, directors, & steering committee members must actively **build cultures of security** through advocacy & outreach.

The Linux Foundation & **OpenSSF** support CRA readiness by helping developers, manufacturers, & stewards align with cybersecurity regulations through collaboration & best practices.

Contents

Foreword

When the Cyber Resilience Act (CRA) regulation was drafted in 2023, the common fear among open source developers at the time was that the new regulation might unintentionally harm global open source projects. Open source advocates and lawmakers subsequently engaged in a productive dialogue to build a mutual understanding of the open source ecosystem's complexities. This effort was eventually crowned by success: The CRA now takes the different commercial and volunteer actors in the open source ecosystem into account. And the open source community learned once more that together we can make things happen.

Many open source organizations are keeping this momentum going and are translating it into efforts to make open source businesses, foundations, and communities CRA-ready. Many remember the spring of 2018 when the full compliance with the then new General Data Protection Regulation (GDPR) went into force seemingly overnight. Most organizations simply had not used the time until that date to prepare and become compliant. The open source community at large seems determined to not repeat this mistake with the CRA.

Organizations like the Linux Foundation, the Open Source Business Alliance, and many others are currently developing guidance materials to help everyone in their respective communities understand what they have to do to become CRA-compliant.

This common effort is quite needed, as the CRA comes with challenges and opportunities alike. The CRA introduces for example new roles like the open source software steward. Open source developers have to grapple with these new concepts to understand how the requirements differ for manufacturers and software stewards and which of these roles apply to them. The new report „Pathways to Cybersecurity Best Practices in Open Source" by the Linux Foundation promotes better understanding of the CRA and helps to alleviate prevalent worries and uncertainties in open source communities.

Despite these challenges and uncertainties, we should however also use the new regulation to confidently point out the advantages of open source. Generally speaking, open source software has quite the lead on other software when it comes to transparency and the potential to fulfill the CRA requirements. Already today some open source projects are going above and beyond what the CRA is asking, with the case studies in this report being a testament to this.

There is still a lot to do until the CRA comes into full force in 2027. While some open source projects are already prepared for the CRA, others are not yet ready to fulfill its requirements. We should therefore do what we do best: cooperate, share knowledge, and support each other. Together we can spearhead global efforts to make software development and distribution more secure.

This report contributes to this bigger effort, spreads awareness, and gives valuable recommendations. I hope that readers will be inspired by the featured projects and their leadership teams and that they will engage in their respective open source communities to ensure open source security and sustainability.

Let's go and get CRA-ready!

MIRIAM SEYFFARTH
HEAD OF POLITICAL COMMUNICATIONS
OPEN SOURCE BUSINESS ALLIANCE

Executive summary: Cyber resilience in open source

The European Union's Cyber Resilience Act (CRA) presents a watershed moment for the open source ecosystem, imposing rigorous cybersecurity requirements on products with digital elements (PDEs) commercialized in the EU. While the regulation will not fully apply until December 2027, with certain provisions taking effect earlier, the stakes are enormous—penalties reaching €15 million or 2.5% of global annual turnover for non-compliance.

The Linux Foundation's analysis of three flagship projects—Civil Infrastructure Platform (CIP), Yocto Project, and Zephyr Project—reveals both the readiness and challenges facing open source software stewards under the new regulatory framework. The CRA introduces a novel distinction between commercial manufacturers who bear primary responsibility for product compliance, and open source software stewards who develop and maintain open source software without monetization. This acknowledges the reality that open source components often constitute up to 96% of modern software while respecting the fundamental openness of the development model.

Each project examined demonstrates advanced security practices that align substantially with CRA requirements. For example, CIP has pioneered adoption of IEC 62443-4-1 industrial cybersecurity standards. Yocto Project provides reproducible builds that create independently-verifiable paths from source to binary code. Zephyr functions as a CVE Numbering Authority with an established Product Security Incident Response Team. All three implement robust vulnerability management processes, though the mandatory five-year support window exceeds some projects' current long-term support commitments.

The regulation's impact extends beyond documentation and vulnerability reporting. It fundamentally alters the relationship between upstream open source projects and downstream commercial adopters, demanding greater collaboration for sustainable security maintenance. Neither manufacturers nor stewards can meet CRA requirements in isolation—manufacturers must conduct due diligence when integrating open source components, while stewards must implement and document cybersecurity policies that facilitate secure development.

Challenges remain significant. Some industrial systems have lifecycles spanning 30 to 50 years, far exceeding typical software support periods. Standardization gaps persist, particularly around software bill of materials (SBOM) formats and consistent naming conventions for vulnerability tracking. The regulation also introduces considerable uncertainty regarding which entities qualify as stewards and how they should handle market surveillance authority (MSA) requests.

The Linux Foundation and Open Source Security Foundation have launched initiatives to address these challenges, focusing on standards development, awareness building, and tooling improvement. Key recommendations include adopting semantic versioning to communicate when substantial modifications trigger new conformity assessments, integrating security practices into development workflows, generating standardized SBOMs, and implementing automated security scanning.

Beyond technical solutions, leadership emerges as the critical factor in cybersecurity readiness. Projects with visible, advocating leaders attract attention, resources, and institutional support essential for long-term security improvements. As regulation creates a more stringent operating environment, such leadership will determine which projects thrive.

The CRA represents not merely a compliance burden but an opportunity to strengthen cybersecurity across the digital ecosystem. Despite initial implementation challenges, it establishes a framework where security becomes an explicit priority rather than an optional consideration. If properly implemented with appropriate collaboration between manufacturers, stewards, and regulators, it may achieve its ambitious goal of "playing a leading international role in the field of cybersecurity" while preserving the innovation engine that open source development represents.

Introduction: A policy challenge catalyzes the open source ecosystem

Today, there is a new impetus for greater collaboration between the communities that develop open source software and the downstream users of that software. That impetus is the Cyber Resilience Act (CRA)[1], recent legislation in the European Union (EU) whose impact on manufacturers, maintainers, and open source stewards will be significant. This report is among the numerous initiatives of the Linux Foundation to prepare stakeholders for the changing global cybersecurity policy landscape.

The goal of the CRA is to establish cybersecurity requirements for devices and software commercialized in the EU. While the language of the CRA is confusing in some areas, it has nonetheless passed into law, with some parts in force by 2026 and full compliance in force by 2027. The penalties for failing to comply with the CRA are steep, as high as EUR 15 million or 2.5% of global annual turnover, whichever is higher. As the proverbial clock for parties subject to the CRA is now ticking, the time to shore up compliance is now. But what is needed for compliance, and where do stakeholders begin?

Under the CRA, the cybersecurity and fitness for purpose of a product - specifically a product with digital elements (PDE) - falls under the responsibility of downstream manufacturers.[2] Most PDEs build on a foundation of open source components that often make up the majority of software in a device, reported in some instances to be as high as 96%.[3] In order to maintain the software in PDEs in an efficient and agile fashion, manufacturers will have to rely on and consume documentation, collaboration processes, and other support measures offered by upstream open source communities. While the CRA prescribes little about this relationship between downstream manufacturers and upstream open source projects, this relationship will be crucial for manufacturers to comply with long-term support and rapid cybersecurity response requirements imposed by the CRA. To make the situation more complex, release cadences, quality assurance practices, data formats and many other aspects of the software supply chain may differ between different open source projects and the downstream manufacturers' processes.

The good news is that a number of widely used open source projects have long taken cybersecurity seriously. Many projects have reliable, well-documented development and collaboration practices. They have invested considerable effort into supporting their downstream users in the adoption of their software. And since all of these aspects are implemented in an open source environment, downstream users have the opportunity to study them as good practises or even to adopt the same (or similar) time-tested tooling and processes for their own development. This is particularly true for open source projects that are intended to serve as the pre-competitive layer for creating devices or products, including operating systems and construction kits for device software.

In this report, we narrowed our focus to three widely used open source projects hosted at the Linux Foundation, each recognized for their security practices and, following a methodical research process, held up as effective stewards under the CRA and beyond it. The first project, the Civil Infrastructure Platform: Industrial Grade Linux (CIP) project, provides a base layer of industrial grade core open source software components, tools, and methods to create Linux-based embedded systems that meet the safety, reliability, and other requirements of modern municipal infrastructure and industrial automation. The second is the Yocto Project, the de-facto industry standard "tool kit" for building custom embedded Linux operating systems, regardless of the hardware architecture. Third, the Zephyr Project is a real-time operating system optimized for resource-constrained devices that supports multiple computer architectures.

With cooperation from key contributors to these projects, this report captures both the extent to which their practices match the requirements of the CRA for stewards and where there may be potential gaps, but also identifies the areas in which these critical projects go above and beyond the CRA requirements and push cybersecurity further than required by the new regulation. From these collective insights, combined with a textual analysis of the CRA and insights from other stakeholders described in the methodology below, this report provides a number of recommendations regarding what actions other open source projects and stakeholders can take, and which priority areas to focus on.

We hope that these insights inspire other open source projects and leaders to better understand their own cybersecurity posture vis-a-vis the requirements of the CRA, and to take the necessary actions to improve them. Through greater collective understanding and collaboration, not only is there a pathway to achieving CRA compliance, but there has never been a greater opportunity to further the resilience, sustainability, and security of open source software.

Methodology

This report is intended to provide orientation, guidance, recommendations, and inspiration to practitioners, manufacturers, and open source software stewards in terms of improving their overall cybersecurity posture and readiness for the CRA. It combines an analysis of the CRA text, a review of the cybersecurity practices of key open source projects, qualitative insights from interviews with project stakeholders from three Linux Foundation hosted projects, and considers takeaways from workshops and stakeholder engagements like the December 2024 Stewards and Manufacturers Workshop. We describe each in detail below.

Textual analysis of the CRA

The analysis of the CRA is based on the CRA text as published by the EU Council on 23 October 2024.[4] It discusses the economic actor roles referenced in the law that are relevant to the workings of the open source ecosystem, and reviews the relationships between them. In particular, the analysis investigates the explicit and implied relationships between economics actors.

The CRA explicitly describes the relationships between manufacturers and consumers, manufacturers and market surveillance authorities (MSAs), and open source software stewards and MSAs. The CRA implies, but does not prescribe in detail, the relationship between open source software stewards and manufacturers. It also does not provide much detail on the relationship of unincorporated open source communities, individual maintainers or contributors with downstream manufacturers or users of the software. Where possible, we attempt to bridge these gaps based on the practices and governance norms typically applied in Linux Foundation projects.

The textual analysis concludes with a brief summary of the obligations of the different actors imposed by the CRA, including an overview of which activities that are covered by the CRA and which are not.

Qualitative interviews with CIP, Yocto Project, and Zephyr Project leaders

Next, through qualitative interviews with key contributors, we reviewed the cybersecurity practices in three essential open source projects: Civil Infrastructure Platform, Yocto Project, and Zephyr Project. The qualitative interviews were guided by the following primary research questions:

1. What is the current state of cybersecurity best practices, documentation, and support processes in your open source project commonly used in products that will fall under CRA scope?

2. How do the roles and obligations defined in the CRA map onto cybersecurity best practices in the relationship between open source communities and downstream manufacturers?

3. What organizational and technical measures should open source stewards and manufacturers implement to enable efficient CRA compliance?

While many projects under the Linux Foundation umbrella have prioritized cybersecurity best practices, these specific projects were chosen from the Linux Foundation's large portfolio as representative examples of how open source communities can embed security into their development lifecycles as a means to improving not only their security posture, but to meet the requirements of stewards under the CRA. Having invested significant effort into their cybersecurity, quality assurance, and documentation practices, they are widely recognized for their strong cybersecurity posture and long-term commitment to security best practices. In addition, each project serves as a foundation for numerous downstream products, giving them a particular impact on the state of cybersecurity across the globe. Crucially, key contributors from each project were available and willing to collaborate on this report, providing firsthand insights into their security strategies. For these reasons, their selection as case studies allows us to highlight concrete practices and lessons learned that other projects and industry stewards may consider adopting, and give us confidence that the recommendations drawn from this research represent the state of the art of cybersecurity best practices.

While these three projects offer valuable perspectives, it should be noted that the conclusions about cybersecurity best practices cannot be easily generalized, as these three projects do not represent the whole open source ecosystem in all its nuances, nor do they encompass the full spectrum of open source security approaches, nor are they the only LF projects in a strong position for CRA compliance. For example, all three projects represent lower-level platforms providing operating system components including an operating system kernel. Other projects—such as Kubernetes, SPDX, or OpenSSF initiatives—could have provided

different angles on CRA readiness. Nonetheless, our sample provides a meaningful cross-section of approaches to cybersecurity and compliance in open source software development. We hope that they provide useful guidance as a starting point for a wide variety of open source community stakeholders.

In the respective project case studies, we analyze their classification under the CRA, how their current practices compare to what will be required by the CRA, what steps may be needed to close the gap between current practice and CRA obligations, and where current practices cover cybersecurity needs not referenced in the CRA or go beyond the baseline requirements of the CRA.

Insights from stakeholder workshops with stewards and manufacturers

The Linux Foundation, embracing its role as a leading open source software steward, is actively engaging manufacturers and open source projects in the implementation of the CRA. To advance the community's understanding of the respective roles and responsibilities of stewards and manufacturers the Open Source Security Foundation (OpenSSF) and Linux Foundation Europe jointly held the Open Source Software Stewards and Manufacturers Workshops shortly after the official publication of the CRA in December 2024. These workshops convened 50 leaders from across the Linux Foundation, other upstream open source foundations, community experts, and government officials.[5] The group shared understandings of the obligations of both manufacturers and stewards, and explored opportunities for greater collaboration over the course of the next three years, as the CRA ultimately comes into effect.

In the plenary and workshop sessions, the participants charted the road to implementing the CRA through three thematic work streams:

1. **Awareness:** Exploring pathways to building greater awareness of the CRA, its timelines, requirements, and overall readiness for when the legislation comes into force;

2. **Standards:** Mapping the formalization and standardization of community best practices into recognized specifications, as well as developing processes;

3. **Tooling:** Identifying the formats and tooling that support software supply chain flows, and identifying opportunities for greater impact.

The workshop was both a catalyst for the establishment of the **Global Cyber Policy working group** under the OpenSSF, and also validated many of the themes raised during the qualitative interviews. This information ultimately influenced and reinforced the recommendations of this report.

Through the above empirical approaches, this report's findings are intended to accelerate the necessary actions for open source project communities to implement the essential requirements of the CRA, as well as create broader awareness of the best practices that make all software more secure.

The EU Cyber Resilience Act: What open source stakeholders need to know

The CRA is a landmark piece of legislation which aims to improve cybersecurity across the board for the internal EU market and world-wide. It aims to achieve three policy goals (Recital 2):

1. To reduce the number and severity of vulnerabilities in digital products;

2. To ensure that cybersecurity is maintained throughout a well-defined product's life cycle;

3. To enable users to make informed decisions based on cybersecurity criteria when selecting and operating digital products.

These requirements apply to all products with digital elements, covering software, hardware, or a combination of both, if they are commercially made available in the EU market. Assuming that international manufacturers as well will respond with stronger cybersecurity practices to be able to offer their products in the EU, the law aims to improve baseline cyber security globally.

With this horizontal and mandatory approach, the EU requires all products with digital elements commercially available in the EU market to adhere to minimum cybersecurity standards, introduces sweeping vulnerability reporting requirements, and imposes a mandatory support period for security fixes, typically five years or longer. While the scope of the CRA is clearly the internal EU market, its ambitions extend beyond Europe, as stated in Recital 7 of the law's introductory text, where it states it aims "to play a leading international role in the field of cybersecurity."[6] The CRA aligns with broader global trends, as other jurisdictions implement cybersecurity regulations with similar goals. For example, the U.S. Cyber Trust Mark program and IoT Cybersecurity Improvement Act, Singapore's Cybersecurity Labelling Scheme, Japan's Cybersecurity Strategy, and Australia's Security of Critical Infrastructure Act all reflect growing international efforts to strengthen the security of digital products and infrastructure.

After entering into force on 11 December 2024, a three year implementation period will begin during which additional guidance will be communicated. In particular, European standards will be adopted that elaborate the essential cybersecurity requirements in general and for specific types of products in particular. Manufacturers will refer to these standards to indicate conformity of their products with the CRA requirements with the CE mark. As stated in Article 71, most obligations introduced by the CRA will apply from 11 December 2027, however manufacturers need to be aware that the reporting obligations concerning actively exploited vulnerabilities and severe incidents will apply 11 September 2026, and the provisions on notification of conformity assessment bodies beginning 11 June 2026.

Understanding the Cyber Resilience Act: Core definitions and frameworks

The CRA introduces a number of novel approaches to cybersecurity regulation which will likely have a significant impact on digital supply chains. Some of them are novel compared to prior regulation, like the idea of products with digital elements. Others represent concepts that reflect the new dynamics of software development to account for the changes introduced by open source software development. In this chapter, we highlight selected concepts in the CRA that manufacturers and open source projects need to be aware of.

Products with digital elements

Products with digital elements (PDEs) define the scope of products covered by the CRA.[7] PDEs include software or hardware products, no matter if they are placed on the market as devices that integrate software and hardware or separately (Article 3(1)). The CRA only applies if the PDE's intended purpose or reasonably foreseeable use includes a direct or indirect logical or physical data connection to a device or network (Article 2(1)), but this applies to many products. PDEs also include their remote data processing solutions to the extent that they are necessary for their function (Article 3(2)). Software is defined as those parts of a product based on computer code (Article 3(4)), while hardware includes all physical components (Article 3(5)). The definition of PDEs does not differentiate between open source software and other products. Instead, obligations differ based on whether PDEs are made available in the market with commercial intent or not, as discussed in more detail below.

Making products commercially available

The CRA defines the "making available on the market" as the supply of a PDE in the course of a commercial activity (Article 3(22)). It further defines "placing on the market" as the first such making available of a PDE (Article 3(21)). In the digital world, "placing on the market" can be understood as the initial release or introduction of a new product, while later software or hardware updates would be "made available." It may be necessary to relate this terminology to the development practices of manufacturers and open source projects.

The recitals clarify that supply in the course of a commercial activity is not limited to charging a price for the product, but also includes indirect monetization, for example, through the acquisition of consumer data or other mechanisms. Another essential concept that the CRA introduces piecemeal and without a definition in Article 3 is that of provisioning software without commercial intent. For example, the provision of open source software that is not monetized is not considered to be a commercial activity (Recital 18). In contrast, making products available means supplying them for distribution or use in the course of a commercial activity. This distinction does not change the nature of what is being supplied, as in closed source or open source products or related services. Instead, it assigns roles and responsibilities to manufacturers (and to a limited extent to other actors) based on the commercial character of their activities.

Monetization includes direct monetization, whereby a price is charged for a software license or device, as well as indirect monetization, as for example in acquiring user data from the use of software provided free of charge. It can be expected that what constitutes indirect monetization will be a subject to some debate, however it is clear that practices common today like the provision of software development environments or web browsers as sales channels for user account subscriptions fall under it. Indirect monetization also includes accepting donations that exceed the costs associated with the design, development and provision of a product with digital elements. Accepting donations without the intention of making a profit, however, is not considered to be a commercial activity (Recital 15).

Manufacturers and Open Source Software Stewards

The CRA is the first EU regulation that explicitly differentiates the roles of manufacturers and open source software stewards. Manufacturers are those originally responsible for making a product with digital elements commercially available in the EU market. Open source software stewards (or stewards, in short) are organizations that systematically support the specific development of open source products and ensure their viability.

Manufacturers develop or manufacture products with digital elements or have them designed, developed or manufactured. Open source software stewards (aka stewards) are organisations other than manufacturers that systematically support the development of specific open source products and ensure their viability.

This distinction reflects an important market reality: Stewards make open source software available to anybody for any purpose, without necessarily engaging directly with the users of their products. While stewards usually do not know exactly or to a full extent how widely their software is used, they provide essential open source software building blocks that are widely used to make up the majority of code on most modern digital products. An important cornerstone of the open source ecosystem is that users of the software do not need to request permission from or register their use with the upstream software communities. Nor are the open source software components provided by stewards monetized or commercialized by them. Instead, stewards are typically financed by donations or membership fees that facilitate their operations and do not constitute a payment for using the software. With this in mind, it would be close to impossible for stewards to shoulder the responsibility for the cybersecurity capabilities of the software that they provide free of charge. The CRA recognizes this challenge and places the responsibility to perform due diligence evaluations for the fitness for purpose of an open source component in a particular commercial product firmly with that product's manufacturer.

Similarly to the typical open source development process, the intended relationship between manufacturers and stewards is based on voluntary participation and cooperation. Manufacturers are encouraged to ensure the viability of their open source dependencies via the requirements to provide security updates for their products over extended periods of time. Stewards are encouraged to provide release and vulnerability documentation with their products. The CRA for the most part leaves it to the market to shape this future manufacturer-steward relationship, hinting however that manufacturers should assume their shared responsibility for the viability and maintenance of their open source dependencies.

Coverage of open source software in the CRA

Recital 18 makes reference to the provision of products with digital elements qualifying as free and open source software (FOSS). Such products should only fall within the scope of the CRA if they are monetized by their manufacturers and supplied in the course of commercial activities. Participating in the development, maintenance, or distribution of open source software - including via online platforms - does not constitute in itself a commercial activity, regardless of how the development is funded or structured. Additionally, contributions to open source projects under somebody else's responsibility and the work of nonprofit organizations developing open source software are not considered to be commercial activities. This important clause should provide certainty to participants in open source projects and communities that their contributions to those are not covered by the CRA. The CRA does not pose a barrier to making contributions to existing open source projects.

No explicit open source exemption

Rather than providing a broad exemption for open source software, the CRA introduces an exemption specifically for software that is provisioned without commercial intent. This distinction has significant implications. Single-vendor open source companies, for example, may not fall under the concept of "provisioning" without commercial intent, as it can be assumed that all activities conducted in the regular course of business are driven by commercial objectives. Furthermore, offering consulting or other services related to open source components developed and owned by the same entity could be interpreted as a form of indirect monetization, potentially bringing such companies within the scope of CRA requirements.

The CRA does not reference the open source definition as maintained by the Open Source Initiative.[8] Open source software is widely understood as software whose source code is released under a license that enables four freedoms: the freedom to study the source code, the freedom to use the source code, the freedom to modify the source code, and the freedom to redistribute it. In Article 3(48), the CRA defines free and open source software as "software the source code of which is openly shared and which is made available under a free and open-source licence which provides for all rights to make it freely accessible, usable, modifiable and redistributable."[9]

The role of open source software stewards under the CRA

The CRA defines open source software stewards as organisations that systematically support the sustained development of free and open source software and ensure their viability. By explicitly requiring that stewards to be a "legal person other than a manufacturer", the CRA effectively shields stewards from the cybersecurity obligations imposed on manufacturers, but also defines constraints on the activities of stewards, which should benefit the development of open source software and not constitute monetization. Even though individuals are considered a legal person and therefore could qualify as a steward, European Commission representatives have indicated their expectation that stewards are juridical persons or incorporated organisations.

A number of important obligations for open source software stewards informed the questions posed during the qualitative research with case study stakeholders. Under the CRA, open source software stewards must implement and document a cybersecurity policy that promotes secure development practices, effective vulnerability handling, and the voluntary reporting of vulnerabilities (Article 24(1)). They are also required to cooperate with market surveillance authorities upon request, providing necessary documentation to address cybersecurity risks (Article 24(2)). Additionally, stewards must report any known, actively exploited vulnerabilities, notify severe incidents, and inform impacted users while providing mitigation measures (Article 24(3)). To support manufacturers integrating open source components, the CRA enables the establishment of voluntary security attestation programs, allowing developers and users to assess conformity with cybersecurity requirements (Article 25).

Due diligence when integrating open source components into commercial products

Manufacturers that incorporate open source components into their commercial products inherently depend on the cybersecurity practices of the open source products they rely on. The security, maintainability, and compliance of open source components are shaped by established open source development practices such as semantic versioning, continuous integration/continuous deployment (CI/CD), and distributed version control systems (DVCS). Additionally, the extent of modifications made to upstream components affects both the security posture of the final product, and the ability to receive timely updates.

Because of this, manufacturers must exercise due diligence when integrating components from third parties including open source software (Article 13(5)), and if a component vulnerability is found, the manufacturer must report it to the component maintainer (Article 13(6)). The need for due diligence regarding open source dependency management was influential in the formation of the critical research question: "To what extent are selected projects ready to support downstream manufacturers?"

Case studies: Leading projects in CRA readiness and cybersecurity best practices

Civil Infrastructure Platform

The **Civil Infrastructure Platform** (CIP) is an open source software project hosted by the Linux Foundation that is focused on establishing a base layer of industrial grade core open source software components to enable the use and implementation of software building blocks in civil infrastructure projects. The CIP project intends to create reusable building blocks that meet the safety, reliability, and other requirements of industrial and civil infrastructure. Additionally, the CIP is committed to providing long-term support (LTS) for its software components, targeting a minimum maintenance period of 10 years to accommodate the extended life cycles common in such systems. The CIP Governing Board is responsible for financial matters with respect to the project while the Technical Steering Committee oversees the technical direction of the project.

Classification

The CIP operates as an open source software steward rather than a manufacturer, which releases open source software that enterprises use to develop their own PDEs and services. It provides updates (i.e. substantially modified versions) via tags. The CIP itself does not release PDEs or services that are monetized and it does not make commercial offerings that complement its open source software. However, interviews with project leadership revealed uncertainty of whether CIP qualifies as a steward. This uncertainty highlights the need for more guidance about the CRA; for example, who will take on the responsibility of being a steward: the project, or the foundation that hosts the project? In the case of the CIP, it is our understanding that the Linux Foundation as the legal entity that hosts the project will be the steward, which delegates the CRA related compliance activities to the CIP.

Compliance with essential requirements of the CRA

DEVELOPMENT PRACTICES

The CIP implements development practices that leverage established open source infrastructure, publishing all software source code on kernel.org and **GitLab**. The project maintains communication with users through subscription-based mailing lists for release notifications. The release process is thoroughly documented on the CIP website, ensuring public auditability, and aligns with IEC 62443-4-1:2018 requirements for secure development of industrial automation and control systems. In fact, the CIP, to the best of the project leader's knowledge, was the first OSS project to adopt IEC 62443-4-1 cybersecurity requirements. "We are trailblazers in this regard," comments Stefan Schroeder, a member of the CIP's Technical Steering Committee and Security Working Group, who was among our interviewees for this report. This **IEC standard** specifies the process requirements for the secure development of products used in industrial automation and control systems. In addition, CIP maintains clear delineation between development and release versions through a tag and branch system, with ongoing development taking place in the main branch and dedicated feature branches.

CYBERSECURITY POLICY

The CIP's cybersecurity policy is maintained in GitLab and complies with IEC 62443-4-1 requirements. Vulnerability reporting operates through public channels, but there is also a private email address for responsible disclosure. The reporting mechanisms are primarily mailing lists, integrating with established Linux kernel and Debian community processes. During the interview, Schroeder raises the concern that documentation is always evolving and there may be a need to version documentation and software together. In addition, he advocates for releases to be treated as comprehensive bundles of both software and documentation to ensure users can identify what documentation was applicable at the time of any given release.

COOPERATION WITH MSAS

The CIP retains contracted maintainers and benefits from significant commitment from member companies, enabling timely and diligent handling of requests. However, Schroeder acknowledges challenges in coordinating responses within a distributed community structure, suggesting that dedicated employed personnel might be beneficial for handling MSA requests. He strongly advocates for the adoption of the **OpenSSF Scorecard** and **security.txt** files as easily implementable best practices that communicate project security status effectively.

VOLUNTARY CYBERSECURITY ATTESTATION PROGRAMMES

The CIP's position as an integrator presents unique challenges, particularly in managing upstream dependencies where they cannot mandate security practices of upstream projects. The project conducts due diligence on upstream projects to compensate for this limitation. While currently not providing SBOMs with releases, this capability is under discussion. For vulnerability disclosure, the CIP relies on kernel.org as the numbering authority for the kernel and on Debian for the rest. But as the CIP is not releasing specific package sets, this coupling is even more loose. The project's approach in many cases exceeds some CRA requirements through additional third-party attestation. However, there remains significant uncertainty about how open source software projects, which typically rely on community-driven development models, should approach cybersecurity attestation programs. The lack of clear guidance on roles, responsibilities, and practical implementation creates challenges for projects like CIP.

Cybersecurity practices that go beyond the CRA

The CIP implements advanced security practices beyond CRA requirements, particularly in its kernel working group, where all release tags and tarballs must be cryptographically signed. This non-repudiation principle ensures commit integrity and unambiguous committer identification, though the project notes concerns about developers using individual private keys rather than a more standardized approach. Additionally, the CIP aims to provide long-term support (LTS) for its software components, targeting a minimum maintenance period of 10 years. This commitment helps ensure that civil infrastructure systems can operate safely and reliably over extended lifecycles. As with all open source, development and source code results are available for public review and critique, enabling others to more easily find and report defects.

Further insight

The CIP identifies several significant gaps in the CRA, particularly regarding the unique characteristics of open source software development and industrial applications.

A primary concern is the regulation's inadequate recognition of the diversity of governance models among open source projects within the open source ecosystem. Projects which do not implement the structures envisioned by the CRA (e.g. the role of open source foundations as stewards) may face similar expectations to their cybersecurity posture without the necessary support organisation. While many major open source software projects are hosted by open source foundations which act as stewards, others are not. For example, the Debian project develops a widely used Linux distribution with a community of over 1600 contributors in 2024, without any hosting organisation that qualifies as a steward.

Furthermore, since OSS contributors have no obligation to deliver any feature or patch in an OSS project at any particular time, industrial users cannot plan the roadmaps of their products that depend on given OSS projects. Users can contribute to OSS projects, but typical medium sized projects use at least thousands of OSS libraries and components - just looking after these dependencies and monitoring their licenses without any contribution is a hassle. No single industrial user could support all OSS projects that they benefit from. As yet another example of how traditional IT workflows and business practices do not transfer to the open source way, there is the issue of sunsetting open source projects at their end of life (EOL). This is a complex topic in OSS, since even if the lead maintainer declared some OSS as EOL, others could fork and maintain it. New regulation forces industrial users to ensure freedom from vulnerabilities. If regulations begin to set deadlines, it will be challenging to enforce such deadlines.

In addition, there are concerns about the CRA's support requirements as set in Article 13 in the context of industrial systems with extremely long life cycles. Specifically, the CRA requires that the support period reflect "the length of time during which the product is expected to be in use." However, railway systems typically operate for 30 to 50 years, and other systems are also long-lived. This makes the CRA's requirement for free security patches throughout a product's reasonable lifecycle potentially unsustainable from a business perspective. In particular, the obligation to provide free security patches for such extended periods creates unsustainable business planning scenarios, as organizations would need to budget for maintenance activities spanning multiple decades. This misalignment between regulatory requirements and the practical realities of long-lived industrial systems highlights a critical challenge in implementing the CRA's security maintenance provisions in sectors with extended operational timeframes.

There is also a potential misalignment between the CRA's security requirements and the realities of highly regulated industries. For example, Schroeder explained that product assessment and approval processes in these sectors often take several months, during which new vulnerabilities may emerge. Schroeder suggests a more nuanced approach to security management, proposing that different components could have different security handling processes based on their attack surface and criticality. For instance, a more robust certification process could be applied to critical components like the kernel, while other system components could follow a regular monthly update cycle. However, the project notes that industry practices and expectations would need to evolve to accommodate such an approach.

Many industrial manufacturers are primarily OSS users rather than contributors, which has led to insufficient funding for long-term maintenance. This imbalance often places a disproportionate burden on upstream developers, such as semiconductor vendors, and risks undermining the sustainability of critical open source infrastructure.

The Yocto Project

The **Yocto Project** is an open source software project hosted by the Linux Foundation that enables developers to create custom Linux-based systems regardless of the hardware architecture. It is the de facto industry standard "tool kit" for building custom embedded Linux operating systems. The project provides a flexible set of tools and a space where embedded developers worldwide can share technologies, software stacks, configurations, and best practices that can be used to create tailored Linux images for embedded and IoT devices, or anywhere a customized Linux OS is needed. It operates with a hierarchical governance structure led by maintainers and coordinated by the Yocto Governance Board.

Classification

Yocto Project operates as an open source software steward, which releases open source software that enterprises use to develop their own PDEs and services. The project itself does not release PDEs or services that are monetized nor does it make commercial offerings that complement its open source software.

Despite operating as a steward, Yocto Project implements several cybersecurity practices that align with manufacturer obligations under Article 13. The project conducts cybersecurity risk assessments through systematic Common Vulnerabilities and Exposures (CVE) monitoring and implements build-time CVE checks. This approach provides a foundation for downstream manufacturers who typically derive their operating system environment from Yocto Project. The project uses a bug-tracking system, Bugzilla, for reporting and fixing bugs as well as a machine-readable CVE checker. Regarding long-term security support, Yocto Project's current four-year LTS support window falls short of the CRA's five-year minimum requirement for security updates and ten-year availability period. The project's recent extension of the Long Term Support period from two to four years may serve as an incentive for organizations to engage with the project, particularly those seeking long-term security support for their products. However, the project acknowledges this gap and indicates readiness to extend its LTS period to meet the five-year threshold. Yocto Project supports a "co-traveller" model, where manufacturers using the platform contribute collectively to

security through common tools, shared data and shared updates and fixes. This collaborative approach amplifies security benefits across the user base, as vulnerability reports or fixes from any single manufacturer benefit the entire ecosystem.

Compliance with essential requirements of the CRA

Development practices

The Yocto Project's existing cybersecurity practices align well with the CRA's requirements, reflecting the project's longstanding commitment to robust security practices. This alignment is not coincidental; the project was initially conceived to bring order to software customization processes through carefully-designed and documented build procedures with input from manufacturers and downstream users across the ecosystem.

Yocto Project maintains a structured release cadence with two primary release types: standard releases every six months (April and October) with six-month support windows, and Long Term Support (LTS) releases every two years that receive four years of project support. Updates are managed through a Git-based workflow that distinguishes between functional and security updates. Functional updates are typically reserved for the six-monthly releases, while security updates are provided between releases as needed. The project employs a branch and tag system where each release becomes a stable branch, governed by explicit maintenance policies. All development occurs in the master branch and undergoes continuous integration testing, indicating robust quality assurance practices. Communication about updates flows through multiple community channels including IRC, mailing lists, blog posts, and member meetings.

The project's comprehensive development practices are centered on a Git-based workflow with clear release management processes. Software releases are communicated through multiple channels, including email announcement lists and weekly status reports that keep downstream consumers informed of upcoming releases. The release process is thoroughly documented on the project's wiki and docs.yoctoproject.org, ensuring transparency and auditability. The Yocto Project documentation is versioned to correspond to releases, making it simple to match a release to the relevant version of the documentation. The project maintains a clear delineation between development versions and releases through its tagging system, with only tagged versions considered official releases. Notably, Yocto Project's build process includes detailed documentation of customizations for integrated upstream components, providing transparency in its software supply chain.

Cybersecurity policy

Cybersecurity governance is structured around several key components. The project maintains a bug tracker in Bugzilla with security-specific tagging capabilities, and its security processes have been strengthened through an audit funded by Germany's Sovereign Tech Fund. The project provides CVE scanning tools that benefit both the core project and downstream manufacturers. Community engagement in security matters occurs through regular bug triage calls, where priorities are collectively identified and addressed. Vulnerability information is shared with the community through mailing lists, and the project maintains documented best practices for bug reporting, including a recommendation for "security files" in all layers. In handling actively exploited vulnerabilities, Yocto Project follows a systematic approach coordinated by its Technical Steering Committee. The process begins with determining whether the vulnerability originates in user-added code or project layers, followed by collabo-

rative development of fixes through open source channels. This transparent approach to vulnerability management ensures that security improvements benefit the entire ecosystem of downstream users.

Cooperation with MSAs

Yocto Project maintains a security team which includes its founder and Linux Foundation Fellow, Richard Purdie. The team works closely with the U.S. National Vulnerability Database (NVD), a repository maintained by the U.S. National Institute of Standards and Technology (NIST). The project's community is prepared to handle security requests, with one full-time employee and several contractors able to handle security-related matters. Purdie explains that while they may be set up to answer the question from a technical perspective, they may not have sufficient time or people on staff to deal with these diligently and in a timely fashion. They would need funding to employ additional staff to do this more rapidly.

Voluntary cybersecurity attestation programmes

Yocto Project approaches attestation through systematic documentation and codification of its processes, with particular emphasis on reproducible builds and compliance verification. However, Purdie notes that the attestation terminology in the CRA is not clear as the attestation programmes are voluntary and yet to be established. In addition, the **Yocto Project holds an OpenSSF silver badge**, and maintains robust security verification practices, including regular checks against vulnerability databases and automated compliance monitoring. The project generates a software bill of materials (SBOM) compliant with the **SPDX** (System Package Data Exchange) 3.0 SBOM standard as part of its build process. It adopted SPDX because it was identified as an efficient path to meeting CRA requirements for traceability and transparency. In addition, Yocto Project provides tooling for CVE analysis and regularly generates reports using it through its **Valkyrie automated testing system**. These tools offer detailed patch metrics for both the core project and meta-layers, enabling thorough security monitoring. While the project doesn't currently provide vulnerability disclosure information (e.g., **Vulnerability Exploitability eXchange (VEX) and Common Security Advisory Framework (CSAF)**) with its releases (because it was not needed until now), it states that it could provide this information. In addition, it maintains comprehensive CVE checking capabilities that can be performed at build time, and offers public resources for tracking security status through its wiki.

Cybersecurity practices that go beyond the CRA

Yocto Project implements several advanced cybersecurity practices that extend beyond the CRA requirements. A cornerstone of these practices is the project's comprehensive commitment to reproducible builds. A build is reproducible if given the same source code, build environment, and build instructions, any party can recreate bit-by-bit identical copies of all specified artifacts. Reproducible builds create an independently-verifiable path from source to binary code, countering many attacks.[10] Yocto Project is one of the only projects to provide reproducible builds for source-based builds on arbitrary host systems, independent of build location. The reproducibility feature serves as a crucial security mechanism by enabling thorough verification of software integrity and supporting detailed analysis of how security might be compromised through specific build changes. Furthermore, its proactive security posture is exemplified by its full support for SPDX 3.0 and its emphasis on build auditability. Looking

beyond current CRA requirements, Yocto Project maintainers recommend a requirement for software manifests to contain sufficient information to enable rebuilding of open source software layers. This aligns with a broader vision of software repairability, where users should retain the ability to rebuild and repair software independently, particularly in scenarios where manufacturers may no longer be available. Additionally, they identify a need for standardized component naming in CVE processing, suggesting this standardization should extend beyond the EU to align with global vulnerability databases like the NVD, creating a more unified approach to security vulnerability management across jurisdictions and the global software industry.

Zephyr Project

The **Zephyr Project** (Zephyr) is an open source project focused on building a best-in-class small, scalable, real-time operating system (RTOS) optimized for resource-constrained devices across multiple architectures. It is a vendor-neutral project where silicon vendors, OEMs, ODMs, ISVs, and OSVs can contribute technology to reduce costs and accelerate time to market for billions of connected embedded devices. Its community members support new hardware, developer tools, sensors, and device drivers. Improvements are frequently delivered to incorporate enhancements in security, device management capabilities, connectivity stacks, and file systems. The Zephyr Governing Board is responsible for financial matters with respect to the project while the Technical Steering Committee oversees the technical direction of the project.

Classification

Zephyr operates as an open source software steward, releasing open source software that enterprises use to develop their own PDEs and services. Zephyr itself does not release PDEs or services that are monetized and it does not make commercial offerings that complement its open source software. The project software is available for download for free from the project website and is developed openly on GitHub.

Compliance with essential requirements of the CRA

Development practices

Zephyr implements structured development practices centered on GitHub-based release management. New versions come out every four months. Zephyr provides a support window for each release and every 2.5 years they come out with a long term stable version which is supported for 2.5 years. Software releases are communicated through multiple channels, including repository tagging, mailing list announcements, Discord notifications, and dedicated Q&A sessions. The release process is comprehensively documented in the project's online **documentation**, ensuring transparency and auditability. The project maintains a clear distinction between development and release versions through its main branch development approach, semantic versioning practices, and detailed "getting started" guides for new contributors.

Cybersecurity policy

The project has a mature cybersecurity posture, with extensive **documentation** which includes a security overview, secure coding guide, and a **sensor device threat model**. In particular, Zephyr maintains robust security oversight as a CVE Numbering Authority with an established Product Security Incident Response Team (PSIRT). The project provides clear channels for **voluntary vulnerability reporting** through a dedicated email address and maintains a vulnerability registry that includes remediation information. This comprehensive security framework ensures effective sharing of vulnerability information with the community and downstream consumers.

Cooperation with MSAs

Zephyr's status as a CVE Numbering Authority facilitates direct communication with PSIRT authorities for vulnerability notifications. The project actively monitors direct reports and maintains a responsive volunteer-based system with typical response times of one to two days for security-related requests.

Regarding the mandatory five-year support period, Zephyr's Kate Stewart anticipates a shift in relationships with third-party maintenance organizations but doesn't expect significant changes in interactions between stewards and manufacturers. More specifically, Stewart contends that manufacturers are unlikely to increase their direct involvement in open source development despite the extended support requirements. Any particular version of Zephyr is supported for 2.5 years; an alternative for manufacturers beyond that timeframe is to upgrade to a newer version of Zephyr. This requires manufacturers to be prepared to upgrade and have a testing infrastructure in place for their application.

Voluntary cybersecurity attestation programmes

Zephyr already participates in voluntary attestation programs. The project utilizes the OpenSSF **Scorecard** and has achieved gold status in the OpenSSF **Best Practices Badge program**, with **annual reviews** ensuring continued compliance. The project enables effortless generation of build-specific SBOMs in SPDX format. A **public dashboard** containing SBOMs for a wide set of build targets is made available by a project member, and illustrates that SBOM generation can happen seamlessly as part of the build process.

Cybersecurity practices that go beyond the CRA

Zephyr implements several advanced cybersecurity practices that extend beyond the CRA requirements. A key element is the project's **embargo** policy, complemented by a structured PSIRT that deliberately moves beyond single-person responsibility. The project maintains a proactive approach to security standards through self-attestation and regular evaluation of new policies emerging from the OpenSSF scorecard for potential project implementation. Kate Stewart emphasizes the importance of recognising the evolving nature of security best practices, and in turn the necessity to continuously update the project's PSIRT processes to align with changes in CVE and National Vulnerability Database (NVD) infrastructure. In addition, Stewart highlights various channels for continuous security improvements, including automated prevention of security regressions through Motor Industry Software Reliability Association (MISRA) scans and dedicated secure practices training for contributors. This comprehensive approach ensures that security considerations are embedded throughout the development process while building security expertise within the contributor community.

Further insight

Regarding potential gaps in the CRA's regulatory framework, Stewart raises two significant concerns. First, the devolution of guidance to member states, particularly regarding SBOMs, is seen as a missed opportunity for standardization. The potential emergence of different SBOM variants across member states could create unnecessary complexity in security documentation and compliance. Second, there is a gap in addressing software provenance risk assessment. Current regulations do not adequately address the challenges that open source stewards face in evaluating and ensuring the integrity of code contributions, particularly regarding repository poisoning attacks and control over code submissions.

Insights from stewards and manufacturers engagement

The December 2024 Stewards and Manufacturers Workshop in Amsterdam was an invitation-only forum hosted by the Linux Foundation Europe and OpenSSF for open source stakeholders to advance the open source community's readiness for CRA compliance. Discussions were structured around three critical workstreams—Standards, Awareness, and Tooling—each focused on defining actionable steps to support cybersecurity best practices in open source software development and integration, and to support the implementation of the CRA over the course of the next three years. These workstreams provided insights into the collaborative efforts required to align open source governance with regulatory expectations, the need for broader industry awareness, and the importance of developing standardized processes and tools for vulnerability management and compliance. The workshop served as a vital forum for translating CRA obligations into practical, community-driven initiatives that will help both stewards and manufacturers navigate the evolving regulatory landscape. Below are the takeaways from each.

Awareness workstream

The Awareness workstream addresses the need to improve understanding of the CRA among open source projects, manufacturers, and the broader ecosystem. Key initiatives include launching a worldwide survey to assess CRA awareness levels, developing an interactive decision tree to help organizations determine their regulatory classification, and updating and improving guidance provided by Linux Foundation about the CRA. Participants highlighted the need for the OpenSSF to lead efforts to create educational materials, including a "CRA 101" course, corporate training modules, and persona-based guidelines tailored to different stakeholders such as manufacturers, market surveillance authorities, and open source software stewards. Other planned activities include dedicated CRA-focused events, workshops, and a comprehensive FAQ and glossary to standardize terminology and address common concerns.

Standards workstream

The Standards workstream focuses on the urgent need to establish recognized cybersecurity standards aligned with the CRA based on best practices developed by the open source community. Participants discussed leveraging existing ISO processes to create standards that address the CRA, ensuring broad adoption and regulatory compliance. Given the tight timeline—the standards related to the CRA need to be in effect before the CRA fully applies (11 December 2027)—participants emphasized the

need for swift action in coordinating efforts with these standardization bodies as well as the risks involved with the accelerated development of standards. Building relationships with European regulatory groups and ensuring liaison status with national bodies were highlighted as crucial steps. The overarching goal is to develop widely accepted standards that provide clarity for open source stewards and manufacturers while streamlining compliance efforts across industries.

Tooling workstream

The Tooling workstream focused on equipping open source projects and manufacturers with the necessary resources to meet CRA obligations. OpenSSF, alongside other Linux Foundation projects, will develop standardized cybersecurity policies and vulnerability reporting templates, ensuring that open source projects have clear guidelines for secure development and disclosure practices. Efforts will also be made to establish communication channels for cooperation with market surveillance authorities (MSAs) and to document best practices for handling MSA requests. Additionally, the group explored the possibility of a voluntary cybersecurity attestation program that could provide manufacturers with greater confidence in open source software security. This initiative may involve collaboration with OpenChain to develop standardized conformance artifacts, helping projects demonstrate their adherence to security best practices and regulatory requirements.

Workshop outcomes

The Amsterdam workshop underscored the critical need for collaboration among open source stewards and manufacturers to meet the CRA requirements. Participants emphasized the importance of establishing recognized standards, enhancing awareness, and developing practical tools to ensure compliance. However, long-term challenges persist, particularly the potential mismatch between regulatory approaches across jurisdictions (i.e., beyond the EU), which could create friction for global open source development. Given that open source software operates across borders, fragmented cybersecurity regulations risk imposing conflicting requirements on projects and manufacturers. To ensure open source continues to thrive as a pillar of innovation and security, regulatory frameworks must strive for greater harmonization, fostering an environment where compliance efforts are aligned rather than fragmented. The lessons learned and actions identified in this workshop will play a crucial role in shaping best practices for secure and sustainable open source development in a rapidly evolving regulatory landscape.

As a concrete outcome of the Stewards and Manufacturer's Workshop discussions, Linux Foundation Europe and the OpenSSF have recently launched a **global initiative** to prepare maintainers, manufacturers, and open source stewards for the CRA and other emerging cybersecurity legislation worldwide. This initiative focuses on developing community-driven cybersecurity standards, providing compliance guidance, and implementing necessary processes and tooling. Additionally, the initiative will be informed by research conducted to empirically evaluate levels of CRA awareness across stakeholder groups. By aligning these efforts with the CRA's objectives, the initiative aims to equip the open source community to navigate the evolving regulatory landscape effectively. For further details, see Global Cyber Policy Working Group Resources in the Resources section of this document.

Conclusion: Strengthening open source security and CRA readiness

This report has explored the key challenges and opportunities for open source projects in meeting the requirements of the CRA through the cybersecurity posture of three leading LF projects. Through these case studies, as well as LF and community-led workshop insights, and analysis of best practices in the CIP, Yocto Project and Zephyr Project, it has become clear that compliance with the CRA is not simply about adhering to a regulatory framework—it is an opportunity to proactively improve security practices, standardize security workflows, and strengthen collaboration across the open source ecosystem.

The CRA represents a sea change for open source communities, introducing regulatory oversight in ways that will have a long-term and pervasive impact on the structure of the upstream-downstream network. Unlike previous regulatory approaches that focused primarily on licensing considerations, the CRA shifts attention to how open source software is provided, maintained, and secured. It formally recognizes the role of open source stewards as distinct from manufacturers, establishing dedicated responsibilities for those who maintain and develop open source projects outside of direct commercial activity. By focusing on the non-commercial provision of open source software rather than licensing alone, the CRA acknowledges the complexity of how modern software is built, integrated, and maintained.

Additionally, the CRA confronts a long-standing pain point in the open source community: projects that are open source in name but not in practice. Many software components are technically open source by license but lack meaningful community collaboration or transparent security processes. By imposing a baseline of cybersecurity requirements on projects that provide critical software components, the CRA seeks to elevate security standards across the open source ecosystem, ensuring that widely used software is not just shared under open source licenses but actively maintained in a secure and responsible manner.

While significant challenges remain—including fragmented regulatory approaches, resource constraints, and evolving security threats—there are clear steps that open source projects, governments, and enterprises can take to improve readiness. These steps include long-term planning for security sustainability, investment in education and training, adoption of standardized security tooling, deeper engagement in collaborative organizations and standards development, and, above all, strong public-facing leadership that drives projects forward.

Recommendations

Building a sustainable security roadmap

One of the key takeaways from this report is the need for a long-term, structured approach to security planning in open source projects. In particular, larger projects need to think beyond short-term compliance goals and work toward a five-year security strategy that includes proactive vulnerability management, sustained funding for security personnel, and clear processes for engaging with downstream users and regulators.

Security transparency is also critical. Open source projects should communicate their existing security capabilities and needs clearly, ensuring that users and stakeholders understand both their strengths and their resource gaps. Establishing a dedicated Product Security Incident Response Team (PSIRT) is a crucial step for larger projects in ensuring vulnerabilities are handled effectively and that security incidents are addressed systematically, rather than relying on ad-hoc efforts from individual maintainers.

Education is another essential component. Open source projects and maintainers need better training on CRA compliance, and best practices adoption in vulnerability disclosure and secure development methodologies. Manufacturers and enterprises integrating open source software should also take responsibility for educating their teams on how to work with upstream projects in a way that strengthens security across the supply chain.

Adopting cybersecurity best practices developed by broad collaborations from across the open source ecosystem is a proven approach to establishing a baseline level of security and supply-chain management practices. This includes applying the **OpenSSF Scorecard** to projects and maintaining **security.txt files**, as well as performing an **OpenChain self-certification assessment** as cybersecurity depends on well-defined supply chain processes.

Additionally, license transparency must be a priority. In the same way that security is now an essential concern in software development, clear and consistent licensing practices should be emphasized. Pursuing OpenSSF best practices accreditation is a good first step in the roadmap to improving security postures and aligning with industry-recognized security frameworks.

Align development practices to CRA concepts where appropriate

Since conformity assessment may have to be renewed in the case of substantial modifications to a PDE, projects should make it explicit if their releases should be considered substantial modifications or a minor functionality update or bug fix. One way to do this is to use semantic versioning to map substantial modifications according to the CRA to major versions, minor functionality updates to minor versions and bug or security fixes to patch versions. This practice will indicate to downstream users of the software when an updated conformity assessment may be necessary. If rolling software releases are made based on the project's main branch, it may be useful to label those as unfinished software (Recital 37), as otherwise it would be difficult for downstream users to differentiate substantial and minor changes.

Since documentation about the project's cybersecurity policy or vulnerability management procedures may change over time, documentation should be versioned and tagged with the project (and possibly maintained as part of the project in the same repository as its source code). This enables adopters to identify the documentation that matches the software version in use.

The long-term support requirements mandated by the CRA, and the even longer support periods in industries like rail transport or aviation, may exceed the lifespan of even well-maintained software projects. In these situations, organizations must not only ensure critical dependencies remain viable by actively participating in development communities, but also maintain the expertise and capacity to replace end-of-life components with newer technologies.

The CRA does not cover individual contributors or unincorporated, loosely organized communities. There i also uncertainty about whether public sector organizations that release software qualify as manufacturers or stewards under the regulation. Nevertheless, these entities may benefit from voluntarily following manufacturer and steward obligations as guidelines. This approach aligns with ecosystem best practices, helps establish a cybersecurity baseline, and prepares organizations for potential future regulatory requirements.

Investing in tooling for compliance and security

To effectively operationalize security best practices, security tooling should be integrated into the software development process as seamlessly as licensing best practices. Just as GitHub prompts developers to choose OSI-compliant licenses when initializing a new project, there is a need for equivalent nudges and recommendations for security best practices, including structured vulnerability reporting, adoption of security policies, and use of standardized software component registries. Where practical, enabling them by default can eliminate frictions from adoption.

Producing SBOMs, such as SPDX 3.0, will be critical for open source projects looking to meet the CRA's requirements. SPDX 3.0 makes it easier for projects to document component dependencies, track vulnerabilities, and provide manufacturers with the data they need for risk assessments. However, greater granularity in SBOMs is needed to improve visibility into specific source files present in deployed software images, rather than only tracking component-level dependencies.

Beyond SBOMs, open source projects should integrate automated security tooling into their development pipelines, including vulnerability tracking, dependency scanning, and supply chain security monitoring. These tools will not only help projects comply with regulatory requirements but also improve the project's resilience against security threats.

These efforts to reduce or limit costs, and make efforts as automatic as possible, will be especially important for smaller OSS projects. A one-person project with at most a few hundred lines of code will typically be unable to sustain costly investments in time and money. Even larger OSS projects have limited resources; ensuring that these efforts are "on by default" within development processes themselves can cause widespread improvements.

Governments and enterprises can support these efforts by investing in open source security tool development and ensuring that widely used security frameworks are accessible, well-documented, and easy for maintainers to adopt.

Standards development and cross-sector collaboration

Security and compliance cannot be solved in isolation. Open source projects must actively engage with other projects, governments, enterprises, and nonprofit organizations to drive security standardization and harmonization across the industry. The CRA provides an opportunity to align security expectations globally, but only if open source communities work together to shape regulations and best practices that reflect the realities of software development.

One area requiring urgent attention is the standardization of software and component naming conventions. Without consistency in how components are identified across vulnerability databases, package managers, and compliance tools, tracking security risks becomes unnecessarily complex. Historically, Common Platform Enumeration (CPE) has been used, but CPE's centralized assignment system cannot scale up to the modern software ecosystem. Alternatives such as package URLs (pURLs) could provide significant improvements, but only if they are agreed on and used. Projects should work together to establish unified naming schemas that align with industry-recognized vulnerability tracking frameworks like CVE/NVD.

Additionally, further clarity is needed around CRA terminology, particularly the distinction between "placing software on the market" and "making software available on the market." These definitions impact compliance requirements for open source projects and their downstream adopters, and collaboration between open source stakeholders and regulatory bodies is necessary to ensure practical, workable interpretations.

Open source projects should participate in security standards bodies, policy discussions, and cross-industry security initiatives to ensure that open source realities are represented in cybersecurity regulations. It is sometimes difficult to find a way to fund such activities. Enterprises and manufacturers should likewise collaborate with the open source projects they rely on, not just consuming software but also contributing to its security and sustainability.

To prevent regulatory fragmentation across jurisdictions, governments must work with industry leaders as well as the wider open source community to harmonize cybersecurity requirements. A globally aligned approach to security regulation will reduce compliance burdens, improve security outcomes, and support the long-term sustainability of open source development.

Addressing emerging security challenges

While many security best practices are well established, new challenges are emerging that current security frameworks do not fully address. One of the most pressing concerns is the security of AI models and the risk of poisoned training datasets. As AI-assisted and AI-generated code becomes more common in open source development, malicious actors may attempt to introduce vulnerabilities at the model-training level, making security verification far more complex.

Open source security frameworks must evolve to account for AI-generated code and provide mechanisms for assessing the integrity of AI-trained models. This includes establishing guidelines for dataset provenance, implementing verification processes to detect potential tampering, and developing new auditing mechanisms to ensure AI-driven contributions adhere to security best practices.

Governments, enterprises, and security researchers must collaborate with the open source community to develop new strategies for securing AI-assisted development, ensuring that open source software remains a safe and trustworthy foundation for innovation.

Leadership as the driving force for open source software security

While security tooling, education, and standards development are all critical, none of these efforts will succeed without strong, public-facing leadership. The most security-mature open source projects have one thing in common: leaders who proactively advocate for their needs, engage with external stakeholders, and push for real change, creating cultures of security through cross-project and industry collaborations that permeate their projects.

The leaders of projects like CIP, Yocto Project, and Zephyr are not just technical experts—they are security ambassadors. They write blogs, give countless presentations, host webinars, and work tirelessly to make the case for security improvements. Their advocacy attracts attention from policymakers, funders, and industry stakeholders, creating opportunities for collaboration and investment.

For open source security to advance, more projects must embrace this model of leadership. Open source maintainers should not wait for solutions to come to them—they should actively seek grants, reach out to regulators, engage with enterprise partners, and make their security needs heard. Richard Purdie's blog post on the challenges facing the Yocto Project, for example, directly led to financial backing from Germany's Sovereign Tech Fund, demonstrating how visible, proactive leadership can yield tangible security improvements.[11] Beyond her work with Zephyr, Kate Stewart's co-leadership of the SPDX project helped it to become an internationally recognized ISO standard.[12] And the leaders of the CIP speak regularly on security topics, a recent example being a webinar led by Yoshitake Kobayashi and Dinesh Kumar illustrating the project's value in delivering secure and robust infrastructure in the context of international standards and regulations.[13]

Governments and enterprises, in turn, must support this leadership by funding open source security initiatives, creating opportunities for collaboration, and listening to the voices of open source maintainers, directors, and other principle stakeholders when crafting policy and compliance frameworks.

The future of open source security depends not just on compliance with regulations like the CRA, but on a cultural shift that prioritizes security leadership, long-term investment, and global collaboration. By taking these steps today, open source projects, enterprises, and governments can work together to create a stronger, more resilient, and more secure software ecosystem for the future.

Resources

Global Cyber Policy Working Group Resources:

- Global Cyber Policy WG GitHub
- #wg-globalcyberpolicy on Slack
- Global Cyber Policy WG Mailing List
- CRA Readiness+Awareness SIG Mailing List
- CRA Tooling+Process+Formats SIG Mailing List
- CRA Spec Standardization SIG Mailing List

Vulnerabilities Reporting & Guidance:

- Guidelines on reporting vulnerabilities specific to LF projects and foundations.
- List of Linux Foundation projects
- Linux kernel security vulnerabilities should be reported to security@kernel.org as described in the Linux kernel security bugs page
- Report vulnerabilities specific to Linux Foundation infrastructure or the main LF website by emailing security@linuxfoundation.org
- Alert on social engineering takeovers

Security Best Practices and Tools:

- Alpha Omega partners with open source software project maintainers to systematically find new, as-yet-undiscovered vulnerabilities in open source code – and get them fixed
- CNCF fuzzing handbook describes what fuzzing is, and how to apply it
- OpenSSF Technical Initiatives, including Best Practices Badge, Scorecard, Sigstore and more
- System Package Data Exchange (SPDX) open SBOM standard (ISO/IEC 5692:2021)
- Post Quantum Cryptography Alliance for the adoption and advancement of post quantum cryptography
- Safer Languages discusses the benefits of programming languages designed with security in mind (NIST)
- Secure by Design principles prioritize the security of customers as a core business requirement (CISA)

Educational Resources:

Featured Certifications

- **Kubernetes and Cloud Native Security Associate** (KCSA)
- **Certified Kubernetes Security Specialist** (CKS)

Instructor-Led Training Courses

- **Security and the Linux Kernel** (LFD441)
- **Kubernetes Security Fundamentals** (LFS460)
- **Zero Trust Security with SPIFFE and SPIRE** (LFS482)
- **Security Coding Fundamentals** (WSKF601)
- **Understanding Vulnerabilities and Security Threats** (WSKF603)

Hands-On Learning Workshops

- **Securing Coding Fundamentals** (WSKF601)
- **Understanding Vulnerabilities and Security Threats** (WSKF603)

Featured Free Training

- **Developing Secure Software** (LFD121)
- **Developing Secure Software - Japanese version** (LFD121-JP)
- **Securing Your Software Supply Chain with Sigstore** (LFS182)
- **Understanding the OWASP® Top 10 Security Threats** (SKF100)
- **Introduction to DevSecOps for Managers** (LFS180)
- **Introduction to Zero Trust** (LFS183)
- **Cybersecurity Essentials** (A Must-Have for ALL Employees) (LFC108)

Free Express Learning (60-90 minutes)

- **Security Self-Assessments for Open Source Projects** (LFEL1005)
- **Securing Projects with OpenSSF Scorecard** (LFEL1006)
- **Automating Supply Chain Security: SBOMs and Signatures** (LFEL1007)

e-Learning Courses

- **Kubernetes Security Essentials** (LFS260)
- **Mastering Kubernetes Security with Kyverno** (LFS255)
- **Modern Air Gap Software Delivery** (LFS281)
- **Implementing DevSecOps** (LFS262)
- **Mastering Infrastructure Security: Strategies, Tools, and Practices** (SKF200)
- **Cloud Native Fuzzing Fundamentals** (LFS251)
- **Detecting Cloud Runtime Threats with Falco** (LFS254)

Research

- **Empirically-driven, security-specific insights from LF Research**

Endnotes

1 **https://eur-lex.europa.eu/eli/reg/2024/2847/oj**
2 The CRA only applies to PDEs if their purpose or foreseeable use has a data connection, and in a few cases (like medical devices) there are separate regulations, but in practice the Act covers most commercial software.
3 Frank Nagle, Kate Powell, Richie Zitomer, and David A. Wheeler, "Census III of Free and Open Source Software: Application Libraries," The Linux Foundation, December 2024. **https://www.linuxfoundation.org/hubfs/LF%20Research/lfr_censusiii_120424a.pdf?hsLang=en**
4 **https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02024R2847-20241120**
5 **https://openssf.org/blog/2024/12/23/cra-stewards-and-manufacturers-workshop-key-takeaways-and-next-steps/**
6 **https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202402847**
7 **https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202402847 p. 29**
8 **https://opensource.org/osd**
9 **https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L_202402847 p.31**
10 **https://reproducible-builds.org/**
11 **https://www.yoctoproject.org/blog/2024/03/28/maintainer-confidential-challenges-and-opportunities-one-year-on/**
12 **https://www.linuxfoundation.org/press/featured/spdx-becomes-internationally-recognized-standard-for-software-bill-of-materials**
13 **https://www.linuxfoundation.org/webinars/enhancing-cyber-resilience-with-cip**

Acknowledgments

The authors are grateful to our sponsoring project communities, and to all whose invaluable input and guidance enabled this report's publication, especially:

Dan Applequist
Gabriele Columbro
Marion Deveaud
Mike Dolan
Esther Garcia
Urs Gleim
Anna Hermansen
Christian "fukami" Horchert
Takehisa Katayama
Jan Kiszka
Megan Knight
Clara Kowalsky
Yoshitake Kobayashi
Todd Moore
Federica Nocerino
Richard Purdie
Christopher "CRob" Robinson
Stefan Schroeder
Kate Stewart
Christian Storm
Andrew Wafaa
David A. Wheeler

About the authors

Mirko Boehm, PhD

Mirko Boehm is a free and OSS contributor, community manager, licensing expert, and researcher, with contributions to major open source projects such as the KDE Desktop (since 1997, including several years on the KDE e.V. board), the Open Invention Network, the Open Source Initiative, and others. He is a visiting lecturer and researcher on free and OSS at the Technical University of Berlin. Mirko Boehm has a wide range of experience as an entrepreneur, corporate manager, software developer, and German Air Force officer. He joined the LF in June 2023 as senior director for community development for LF Europe, where he focuses on driving engagement and collaboration between all European open source stakeholders. Mirko speaks English and German and lives in the Berlin area.

Hilary Carter

Based in Canada with dual Irish and Canadian citizenship, Hilary Carter joined the Linux Foundation in March, 2021 to launch and lead LF Research, working extensively with stakeholders across the open source community. Since its inception, LF Research has published the definitive collection of empirical insights into open source as a paradigm for mass collaboration at scale. Systematically measuring the impact of open source software, open hardware, open standards, and open data, LF Research has provided decision-useful data for individual project communities, enterprises, governments, and society at large. Before joining the Linux Foundation, Hilary led a global, syndicated research institute focused on blockchain technology, and has years of professional experience in management consulting, communications, and financial services. She holds an MSc in Management from the London School of Economics.

Cailean Osborne, PhD

Cailean Osborne is a senior researcher at the Linux Foundation, who leads research projects on diverse open source trends and policy topics, from the impacts of open source funding to open source AI governance. Cailean has a PhD in Social Data Science from the Oxford Internet Institute at the University of Oxford, and in 2023-2024 he was a visiting researcher at Peking University's Open Source Software Data Analytics Lab in Beijing, China. Previously, Cailean worked in AI policy at the UK government and served as a UK government delegate at the OECD's Global Partnership on AI and the Council of Europe's Ad Hoc Committee on AI. Cailean is a polyglot who is based in Berlin, Germany.

THE LINUX FOUNDATION | Research

Founded in 2021, **Linux Foundation Research** explores the growing scale of open source collaboration and provides insight into emerging technology trends, best practices, and the global impact of open source projects. Through leveraging project databases and networks and a commitment to best practices in quantitative and qualitative methodologies, Linux Foundation Research is creating the go-to library for open source insights for the benefit of organizations the world over.

CIVIL INFRASTRUCTURE PLATFORM
INDUSTRIAL GRADE LINUX

The CIP is an open source project hosted by the Linux Foundation to develop and maintain a sustainable industrial-grade software platform for civil infrastructure systems. CIP's mission is to enable secure, reliable and long-lasting solutions that power critical systems around the world. For more information, visit **www.cip-project.org**.

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

The Open Source Security Foundation (OpenSSF) is a cross-industry initiative by the Linux Foundation that brings together the industry's most important open source security initiatives and the individuals and companies that support them. The OpenSSF is committed to collaborating and working upstream and with existing communities to advance open source security. For more information, please visit us at **openssf.org**.

yocto PROJECT

The Yocto Project is an open source collaboration project that provides templates, tools and methods to help you create custom Linux-based systems for embedded system deployments in connected edge devices, servers, or virtual environments, regardless of the hardware architecture. Please visit **yoctoproject.org**.

Zephyr®

The Zephyr® Project is an open source, scalable real-time operating system (RTOS) supporting multiple hardware architectures. To learn more, please visit **zephyrproject.org**.

To reference this work, please cite as follows: Mirko Boehm, Hilary Carter, and Cailean Osborne, "Pathways to Cybersecurity Best Practices in Open Source: How the Civil Infrastructure Platform, Yocto Project, and Zephyr Project are Closing the Gap to Meeting the Requirements of the Cyber Resilience Act," foreword by Miriam Seyffarth, The Linux Foundation, March 2025.