Beyond NIS2:

Software Supply Chains, and the Cyber Resilience Act (CRA)

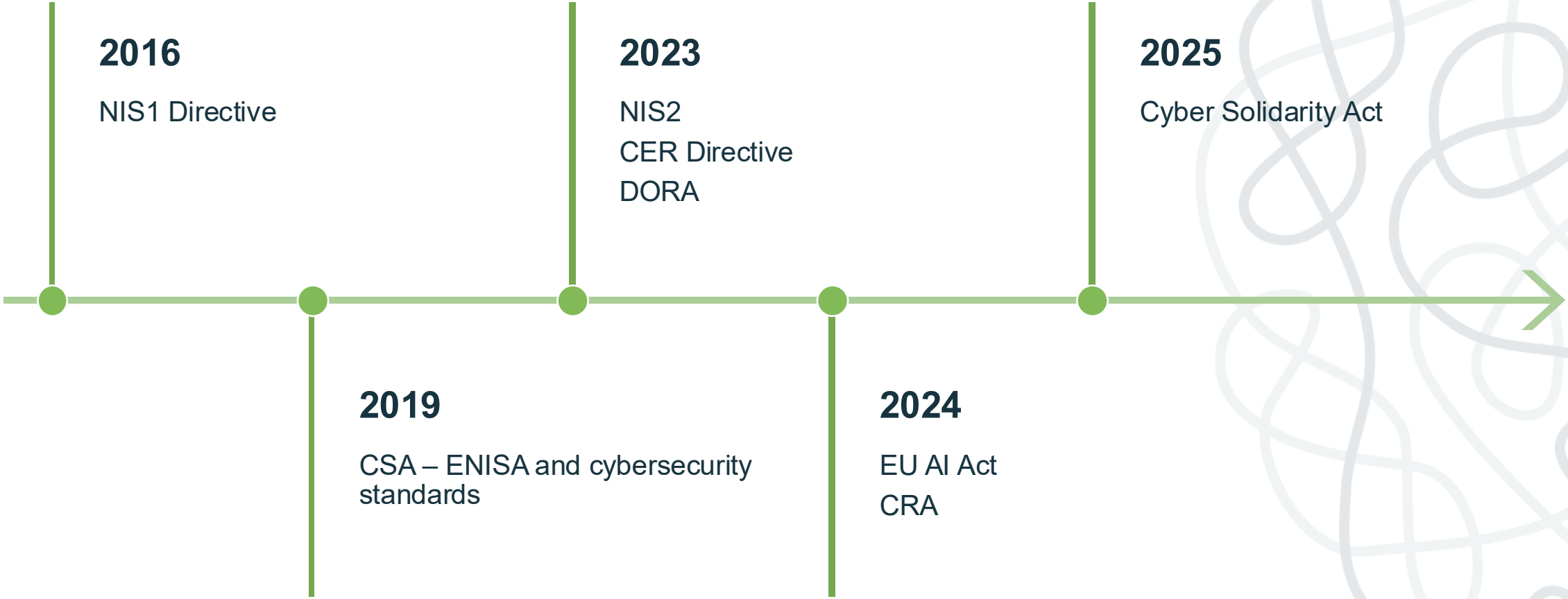
Barry O'Donovan, INEX

RIPE 93, Sofia, Bulgaria, October 28th, 2026

Interconnecting networks
and people since 1996

Regulatory Tsunami

Legislative Milestones



Political Backlash & Competitive Alarms

- French President Emmanuel Macron explicitly called for a European **regulatory pause**, arguing that Europe must digest existing legislation rather than imposing new burdens…

- Mario Draghi's landmark 2024 report on EU Competitiveness delivered a **scathing critique** of the digital regulatory regime.
  - the EU has generated around 100 tech-related laws
  - And over 270 digital networks regulatory bodies

Subscriber Only Politics

Irish civil servants criticised in private missive for delays in transposing EU rules

Department of Taoiseach secretary general privately chastised colleagues for delays in transposing European legislation



NIS2 & the CRA

NIS2 and the CRA

- NIS2 now applies to many of us as operators of ISPs and IXPs.
- This creates a legal obligation to manage operational cyber risk and supply chain vulnerabilities.

- What about the software manufacturers and open-source projects supplying us?

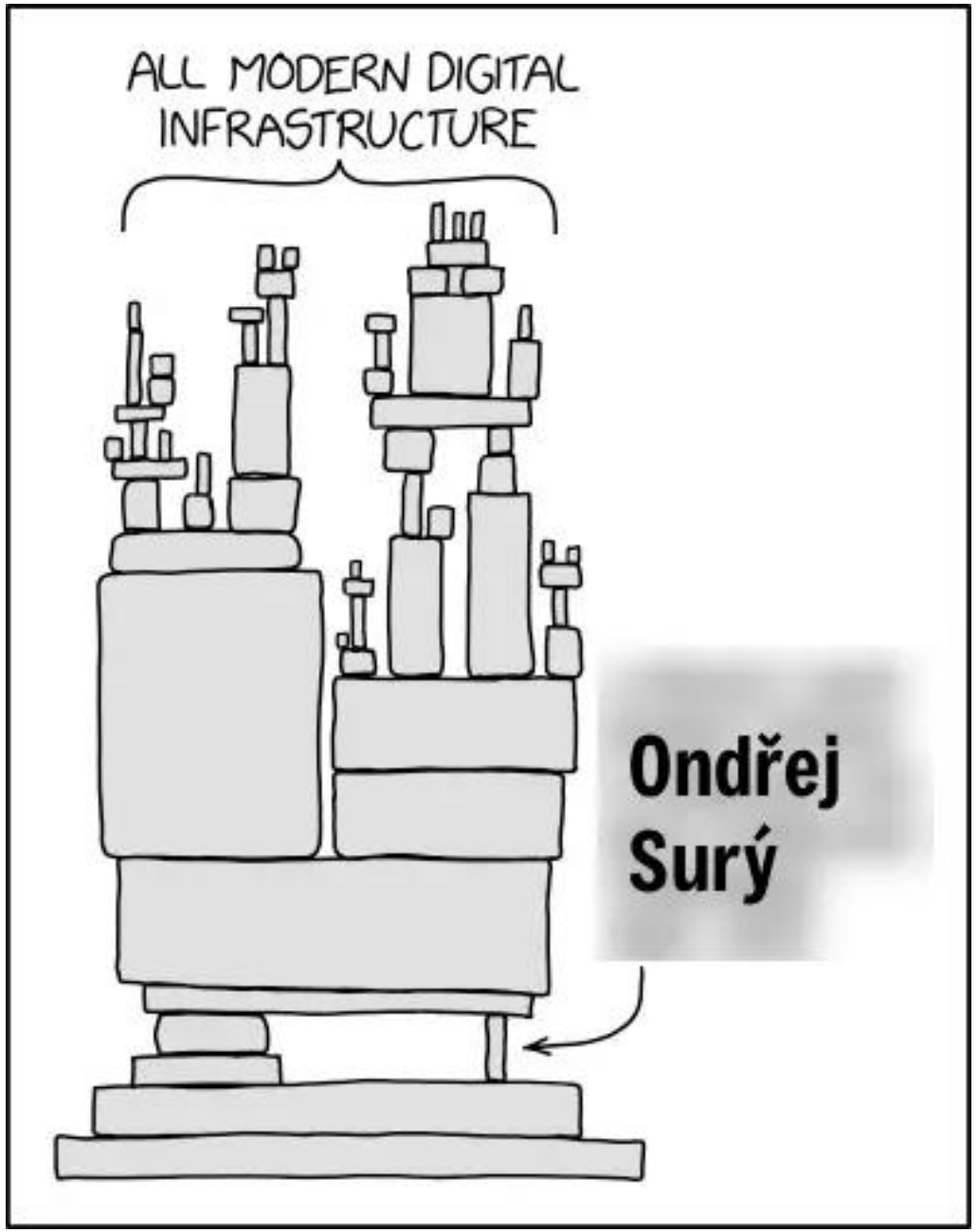
- Enter the Cyber Resilience Act (CRA):
  - Imposing direct cybersecurity and vulnerability reporting obligations on software made available on the EU market.

NIS2 and the CRA

NIS2 governs how you run your network.

The CRA governs the software you run it with.

Open Source Under the CRA



https://codeberg.org/oerdnj/deb.sury.org/issues/145#issuecomment-26297270

https://xkcd.com/2347/

Individual FOSS Developers Not In Scope

- Software is only regulated if it is supplied "in the course of a commercial activity."
- The CRA explicitly states that the mere act of hosting software on open repositories, collaboration platforms (like GitHub or GitLab), or distributing it through package registries (like PyPI or npm) does not constitute making software available on the commercial market.
- None of the following activities push an ad-hoc project into CRA scope:
  - Accepting voluntary donations or tips, provided they are intended to offset hosting or development costs rather than generate commercial profit.
  - Maintaining a project, including tagging regular software releases
  - Receiving bug fixes or PRs from engineers employed by commercial corporations.
  - Receiving non-commercial public research grants or academic funding.

Open-Source Software Stewards

The CRA recognises that non-profit foundations are different from commercial vendors.

‘open-source software steward’ means a legal person, other than a manufacturer, that has the purpose or objective of systematically providing support on a sustained basis for the development of specific products with digital elements, qualifying as free and open-source software and ***intended for commercial activities***, and that ensures the viability of those products

The CRA: Stewards vs. Commercial

	Open-Source Software Steward	Open-Source Manufacturer
Legal & Financial Nature	Any legal person (foundation, association or company) that doesn't monetise the software	Anyone who monetises it, individual or company
Monetisation Model	Code is freely distributed without charging license fees or selling enterprise "paywalled" tiers.	"Open Core" (free basic / paid enterprise), commercial support contracts tied directly to product delivery, or paid dual-licensing.
Examples in our Ecosystem	INEX (IXP Manager), NLnet Labs (Unbound, Routinator, …) …	Enterprise Linux Vendors (e.g., Red Hat RHEL, Canonical Ubuntu Pro), Commercial Appliance Vendors (packaging FRR/Linux into proprietary network boxes)

“Intended for Commercial Activities”

- Recital 19: FOSS *“ultimately intended for commercial activities”*
  - *“integration into commercial services or into monetised products”*
  - “Ultimately”: indirect use counts

1. Do you make money from the software itself?
   - => manufacturer.
2. If not, are you an organisation that keeps software going for businesses to build products or services on?
   - => steward.
3. Neither? You’re out of scope

The Commission's Examples

- **Not-for-profit, membership-funded, members contribute code**
  - => steward (Example 33)
- **Foundation maintaining a component for others' commercial products**
  - => steward (Example 28)
- **Company publishing a library it uses itself, not monetised**
  - => steward, *not* manufacturer (Examples 25, 29)
- **Individual developer, even if companies donate**
  - => no obligations (Examples 27, 34)

CRA Timelines

CRA Timelines

- Enacted as Regulation (EU) 2024/2847 in Oct 2024.
  - Regulation vs. Directive - applies without national transposition acts.
- Application follows a staggered timeline under Article 71.
- Mandatory Reporting for Manufacturers: September 11, 2026
- Full Market Entry: December 11, 2027
  - Including reporting obligations for open-source software stewards
- Market Surveillance and Fines - fines reach up to €15 million or 2.5% of total worldwide annual turnover

CRA Reporting Obligations for Stewards

- Any **actively exploited vulnerability**
  - *where the steward provides engineering resources;*
  - And, where appropriate, to inform all users
  - And, where a direct relationship exists, inform them directly

- Any **severe incident having an impact on the security** of the product.
  - *where a steward provides the underlying IT infrastructure*
  - where appropriate inform all users by way of a general announcement

- *Reporting obligations continue to apply after a version is no longer supported*

CRA Reporting Timelines

- The 3-stage notification clock (Article 14):
  - **Within 24 hours:** “Early warning notification” to the national CSIRT and ENISA via a dedicated Single Reporting Platform.
  - **Within 72 hours:** Detailed notification with sensitivity and initial assessment (unless provided in the early warning).
  - **Final report:** within 14 days after a corrective patch/remediation or within 1 month after the 72-hour notification (for severe incidents).

Other Obligations of Open-Source Software Stewards

- Establish a cybersecurity policy “to foster the development of a secure product” … “as well as an effective handling of vulnerabilities”
  - “foster the voluntary reporting of vulnerabilities”
  - Should cover secure development practices, vulnerability intake, patching workflows, and the remediation of discovered flaws.
  - tailored to the steward's organizational setup
  - "and promote the sharing of information concerning discovered vulnerabilities within the open-source community”

Other Obligations of Open-Source Software Stewards

- Cooperate with the market surveillance authorities, at their request, with a view to mitigating the cybersecurity risks
  - Includes the provision of the cybersecurity policy
  - Corrective action orders (i.e., create the policy)

So where's the stick?

- No fines, and no other money penalties "for any infringement"
  - "should not impose other kinds of penalties with pecuniary character"

- Persuasion, rather than punishment
  - Your users' NIS2 obligations.
  - Manufacturers' due diligence.
  - Your reputation.
  - A CSIRT can go to your users over your head.

- Monetise it and you're a manufacturer, with full fines.

Consuming Open-Source Under NIS2 (Art 21)

Consuming Open Source Under NIS2 (Art. 21)

"It's free software" is no longer an excuse for unmanaged supply chain risk.

Consuming Open Source Under NIS2 (Art. 21)

- **Mandatory Supply Chain Due Diligence:**
  - NIS2 requires entities to assess the security posture of third-party software, including open-source routing daemons, automation scripts, and management platforms.

- **What Operators Must Implement:**
  - **Software Asset Inventory:** Clear mapping of all open-source packages in the production path.
  - **Vulnerability Tracking:** A structured process to ingest advisories and evaluate reachability/exploitability.
  - **Timely Patch Management**: Defined change-management procedures to deploy security patches without undue delay.

Case Study: How IXP Manager Aligns as an OSS Steward

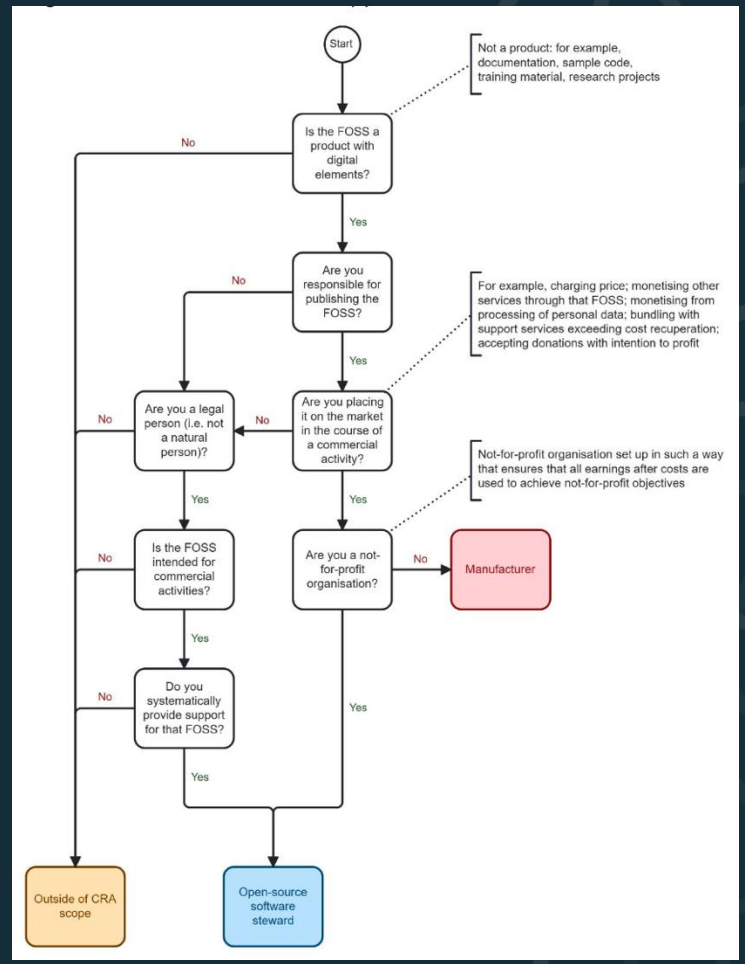
- **Secure Application Development Policy** (docs.ixpmanager.org)
- **Formal CVD Policy:** Clear intake via PGP/Signal/GHSA, 48-hour response target, and explicit support lifecycle tables. (SECURITY.md)
- **GitHub Security Advisories (GHSA):** Generates standardised CVEs and machine-readable data (OSV format) that can feed directly into IXP automated scanners.
- **Proactive Code Hardening:** Using third-party pen tests and responsible disclosures as catalysts for broader internal code audits (6 consecutive security-focused releases).
- **CRA Escalation Playbook:** Documented procedures to alert the ENISA Single Reporting Platform and national CSIRTs within 24–72 hours if active exploitation is confirmed. *(WIP)*

Publishing Software?

- **Scenario A:** Internal Scripts & Code Dumps (Exempt)
  - Sharing an open-source automation script or BGP filter generator without commercial monetisation? CRA out of scope.
- **Scenario B:** Sustained Infrastructure Tooling (Steward)
  - Maintaining an industry tool used in a commercial context? You fall into the Steward bracket: publish a security policy, and mandatory disclosures.
- **Scenario C:** Monetised / Packaged Software (Manufacturer)
  - Selling a software solution or managed appliance to other networks? You are a Manufacturer and must comply with full CE-marking, conformity assessments, and SBOM requirements.

Resources

- The Cyber Resilience Act
- NIS 2 Article 21
- FAQs for the CRA SRP
- Commission Guidance for the CRA
- IXP Manager's Secure Application Development Policy (CA BY-SA)
- IXP Manager's Security Policy




inex
Thank You.
inex
Interconnecting networks
and people since 1996