WHITEPAPER | ELITE EDITION

Regulation as Code

Operationalising CRA, NIS2 and DORA into Engineering Reality

The CODIFY Framework: 204-Requirement Executable Policy Catalogue

Kieran Upadrasta
CISSP, CISM, CRISC, CCSP | MBA | BEng
27 Years' Cyber Security Experience | Big 4 Consulting (Deloitte, PwC, EY, KPMG)
21 Years Financial Services | AI Cyber Security Programme Lead
Professor of Practice (Cybersecurity, AI & Quantum Computing), Schiphol University
Honorary Senior Lecturer, Imperials | UCL Researcher

www.kie.ie | info@kieranupadrasta.com | March 2026

Table of Contents

> **UNIQUE CONTRIBUTION: CODIFY treats regulation as a software dependency — 204+ requirements compiled into OPA/Rego policies with version control, testing, and lifecycle management.**

CONFORM System Position: This paper (WP07) extends the CONFORM master theory (WP01) for 204-requirement executable policy catalogue. See WP01 for foundational methodology.

Executive Summary

CODIFY transforms 204+ regulatory requirements into machine-executable OPA/Rego policies. v10.0 adds three critical elements: real Rego code examples, the 204-requirement catalogue structure, and a complete policy lifecycle model with failure scenarios.

1. The Regulation-as-Code Paradigm

Traditional compliance interprets regulatory text through human judgement, creating inconsistency. CODIFY eliminates interpretive variance: a CRA requirement either passes or fails. Of the 204+ requirements catalogued, approximately 85% are fully deterministic; the remaining 15% contain subjective language ("appropriate measures") requiring human governance overlay through DOCTRINE (WP04) and INSTITUTE (WP05).

2. Worked Rego Policy Example

The following OPA/Rego policy enforces CRA Article 13(5) SBOM completeness:

```
package cra.art13.sbom_completeness  # CRA Article 13(5): SBOM must document all
components default allow = false  allow {     input.sbom.format == "spdx-2.3"
input.sbom.component_count > 0     input.sbom.missing_components == 0
input.sbom.unsigned == false }  deny[msg] {     input.sbom.missing_components > 0     msg
:= sprintf("CRA-13.5 violation: %d components missing from SBOM",
[input.sbom.missing_components]) }
```

This policy evaluates at every build gate. A missing component triggers a deny response with a specific CRA article reference, blocking deployment until the SBOM is complete. The signed evaluation result enters the EVIDENCE chain automatically.

3. The 204-Requirement Catalogue

Each requirement has a unique ID, regulatory source, control objective, policy package, and defined inputs:

Req ID	Regulation	Article	Control Objective	Policy Package	Inputs
CRA-13.5-01	CRA	Art. 13(5)	SBOM completeness for all components	cra.art13.sbom_ completeness	sbom.format; component_count
CRA-13.6-03	CRA	Art. 13(6)	Patch delivery within regulatory SLA	cra.art13. patch_sla	hours_since_ disclosure; severity
CRA-14.1-01	CRA	Art. 14	Vulnerability reporting within 24 hours	cra.art14. vuln_reporting	hours_since_ awareness
NIS2-21.2e-01	NIS2	Art. 21(2)(e)	Vulnerability handling and disclosure	nis2.art21. vuln_handling	remediation_status; disclosure_status
DORA-17.1-01	DORA	Art. 17	Incident classification within 4 hours	dora.art17. incident_class	detection_time; classification_time
DORA-28.1-01	DORA	Art. 28	Third-party risk register maintained	dora.art28. tpr_register	provider_count; assessed_count

Table 1: 204-Requirement Catalogue — Sample Entries (6 of 204)

---

4. Policy Lifecycle Model

Policies follow a five-stage lifecycle from creation through retirement:

Stage	Activity	Trigger	Owner	Output
Create	Encode regulatory requirement as Rego policy + test cases	New regulation or amendment published	Policy Engineer	Policy + tests in version control
Test	Validate against known-good and known-bad inputs	Every policy change (CI pipeline)	Automated (pipeline)	Test results; coverage report
Deploy	Release to production policy engine (OPA)	Passing tests + peer review approval	Policy Engineer + reviewer	Deployed policy; version tag
Monitor	Track pass/fail rates; false positive analysis	Continuous (real-time)	Platform Team	Telemetry; alert on anomaly
Update	Revise policy based on regulatory change or feedback	Regulatory amendment; false positive report	Policy Engineer	Updated policy; changelog entry

Table 2: Policy Lifecycle — Five Stages with Triggers and Owners

---

5. Failure Scenarios

CODIFY explicitly addresses five failure modes:

Failure Mode	Detection	Impact	Recovery
Incorrect policy encoding	Test case failure; false positive reports	Legitimate deployments blocked (false positive)	Emergency policy update; bypass log
Policy conflict (contradictory rules)	Conflict detection in CI test suite	Unpredictable gate behaviour	Conflict resolution by Policy Engineer
Stale policy (regulation changed)	Regulatory change monitoring alert	Non-compliance with updated requirement	Priority policy update cycle
OPA engine failure	Health check monitoring; failover detection	Pipeline gates non-functional	Failover to backup; manual approval
Subjective requirement misinterpretation	Periodic legal review of codified policies	Compliance gap for ambiguous requirements	Legal + Policy joint review

Table 3: CODIFY Failure Modes — Detection and Recovery

6. Case Studies

ILLUSTRATIVE SCENARIO: Pan-European fintech (DORA scope, 120 microservices). DORA compliance in 9 months vs projected 24 months (2.7x acceleration). Audit cycle from 6 months to 2 weeks. 89% reduction in manual compliance effort. Zero policy violations sustained for more than 4 hours. 12 false-positive policy triggers in first quarter, resolved through test case refinement.

7. Limitations

Rego learning curve: 2-4 weeks for experienced engineers. Policy maintenance requires dedicated ownership. 15% of regulatory requirements contain subjective language requiring human interpretation before codification. OPA engine adds 2-5ms per policy evaluation. Policy catalogue requires update within 30 days of regulatory amendment publication.

About the Author

Kieran Upadrasta

CISSP, CISM, CRISC, CCSP | MBA | BEng

Kieran Upadrasta is a distinguished cyber security architect with 27 years of professional experience, including 21 years specialising in financial services and banking. His career spans all four major consulting firms—Deloitte, PwC, EY, and KPMG—where he has advised board members and senior executives across global institutions on regulatory compliance, cyber risk governance, and digital operational resilience.

Mr. Upadrasta has worked with the largest corporations to achieve compliance with OCC, SOX, GLBA, HIPAA, ISO 27001, NIST, PCI, and SAS70. His expertise spans business analysis, consulting, technical security strategy, architecture, governance, security analysis, threat assessments, and risk management.

Professional Memberships & Affiliations

- Professor of Practice in Cybersecurity, AI, and Quantum Computing, Schiphol University
- Honorary Senior Lecturer, Imperials
- Lead Auditor, ISF Auditors and Control
- Platinum Member, ISACA London Chapter
- Gold Member, ISC² London Chapter
- Cyber Security Programme Lead, PRMIA
- Researcher, University College London (UCL)

Contact: info@kieranupadrasta.com | www.kie.ie | linkedin.com/in/kieranupadrasta

References

1. Regulation (EU) 2024/2847 (CRA).
2. Directive (EU) 2022/2555 (NIS2).
3. Regulation (EU) 2022/2554 (DORA).
4. Regulation (EU) 2024/1689 (EU AI Act).
5. ISO/IEC 42001:2023.
6. NIST CSF 2.0, Feb 2024.
7. NIST FIPS 204 (ML-DSA).
8. MITRE ATT&CK; v15.
9. OWASP ASI Top 10, 2025.
10. ISO/IEC 27001:2022.
11. ENISA Threat Landscape 2025.
12. OPA/Rego Documentation.

WHITEPAPER | ELITE EDITION

Regulation as Code

Operationalising CRA, NIS2 and DORA into Engineering Reality

The CODIFY Framework: 204-Requirement Executable Policy Catalogue

Kieran Upadrasta
CISSP, CISM, CRISC, CCSP | MBA | BEng
27 Years' Cyber Security Experience | Big 4 Consulting (Deloitte, PwC, EY, KPMG)
21 Years Financial Services | AI Cyber Security Programme Lead
Professor of Practice (Cybersecurity, AI & Quantum Computing), Schiphol University
Honorary Senior Lecturer, Imperials | UCL Researcher

www.kie.ie | info@kieranupadrasta.com | March 2026

Table of Contents

> **UNIQUE CONTRIBUTION: CODIFY treats regulation as a software dependency — 204+ requirements compiled into OPA/Rego policies with version control, testing, and lifecycle management.**

CONFORM System Position: This paper (WP07) extends the CONFORM master theory (WP01) for 204-requirement executable policy catalogue. See WP01 for foundational methodology.

Executive Summary

CODIFY transforms 204+ regulatory requirements into machine-executable OPA/Rego policies. v10.0 adds three critical elements: real Rego code examples, the 204-requirement catalogue structure, and a complete policy lifecycle model with failure scenarios.

1. The Regulation-as-Code Paradigm

Traditional compliance interprets regulatory text through human judgement, creating inconsistency. CODIFY eliminates interpretive variance: a CRA requirement either passes or fails. Of the 204+ requirements catalogued, approximately 85% are fully deterministic; the remaining 15% contain subjective language ("appropriate measures") requiring human governance overlay through DOCTRINE (WP04) and INSTITUTE (WP05).

2. Worked Rego Policy Example

The following OPA/Rego policy enforces CRA Article 13(5) SBOM completeness:

```
package cra.art13.sbom_completeness  # CRA Article 13(5): SBOM must document all
components default allow = false  allow {     input.sbom.format == "spdx-2.3"
input.sbom.component_count > 0     input.sbom.missing_components == 0
input.sbom.unsigned == false }  deny[msg] {     input.sbom.missing_components > 0     msg
:= sprintf("CRA-13.5 violation: %d components missing from SBOM",
[input.sbom.missing_components]) }
```

This policy evaluates at every build gate. A missing component triggers a deny response with a specific CRA article reference, blocking deployment until the SBOM is complete. The signed evaluation result enters the EVIDENCE chain automatically.

3. The 204-Requirement Catalogue

Each requirement has a unique ID, regulatory source, control objective, policy package, and defined inputs:

Req ID	Regulation	Article	Control Objective	Policy Package	Inputs
CRA-13.5-01	CRA	Art. 13(5)	SBOM completeness for all components	cra.art13.sbom_completeness	sbom.format; component_count
CRA-13.6-03	CRA	Art. 13(6)	Patch delivery within regulatory SLA	cra.art13.patch_sla	hours_since_disclosure; severity
CRA-14.1-01	CRA	Art. 14	Vulnerability reporting within 24 hours	cra.art14.vuln_reporting	hours_since_awareness
NIS2-21.2e-01	NIS2	Art. 21(2)(e)	Vulnerability handling and disclosure	nis2.art21.vuln_handling	remediation_status; disclosure_status
DORA-17.1-01	DORA	Art. 17	Incident classification within 4 hours	dora.art17.incident_class	detection_time; classification_time
DORA-28.1-01	DORA	Art. 28	Third-party risk register maintained	dora.art28.tpr_register	provider_count; assessed_count

Table 1: 204-Requirement Catalogue — Sample Entries (6 of 204)

4. Policy Lifecycle Model

Policies follow a five-stage lifecycle from creation through retirement:

Stage	Activity	Trigger	Owner	Output
Create	Encode regulatory requirement as Rego policy + test cases	New regulation or amendment published	Policy Engineer	Policy + tests in version control
Test	Validate against known-good and known-bad inputs	Every policy change (CI pipeline)	Automated (pipeline)	Test results; coverage report
Deploy	Release to production policy engine (OPA)	Passing tests + peer review approval	Policy Engineer + reviewer	Deployed policy; version tag
Monitor	Track pass/fail rates; false positive analysis	Continuous (real-time)	Platform Team	Telemetry; alert on anomaly
Update	Revise policy based on regulatory change or feedback	Regulatory amendment; false positive report	Policy Engineer	Updated policy; changelog entry

Table 2: Policy Lifecycle — Five Stages with Triggers and Owners

5. Failure Scenarios

CODIFY explicitly addresses five failure modes:

Failure Mode	Detection	Impact	Recovery
Incorrect policy encoding	Test case failure; false positive reports	Legitimate deployments blocked (false positive)	Emergency policy update; bypass log
Policy conflict (contradictory rules)	Conflict detection in CI test suite	Unpredictable gate behaviour	Conflict resolution by Policy Engineer
Stale policy (regulation changed)	Regulatory change monitoring alert	Non-compliance with updated requirement	Priority policy update cycle
OPA engine failure	Health check monitoring; failover detection	Pipeline gates non-functional	Failover to backup; manual approval
Subjective requirement misinterpretation	Periodic legal review of codified policies	Compliance gap for ambiguous requirements	Legal + Policy joint review

Table 3: CODIFY Failure Modes — Detection and Recovery

6. Case Studies

ILLUSTRATIVE SCENARIO: Pan-European fintech (DORA scope, 120 microservices). DORA compliance in 9 months vs projected 24 months (2.7x acceleration). Audit cycle from 6 months to 2 weeks. 89% reduction in manual compliance effort. Zero policy violations sustained for more than 4 hours. 12 false-positive policy triggers in first quarter, resolved through test case refinement.

7. Limitations

Rego learning curve: 2-4 weeks for experienced engineers. Policy maintenance requires dedicated ownership. 15% of regulatory requirements contain subjective language requiring human interpretation before codification. OPA engine adds 2-5ms per policy evaluation. Policy catalogue requires update within 30 days of regulatory amendment publication.

About the Author

Kieran Upadrasta

CISSP, CISM, CRISC, CCSP | MBA | BEng

Kieran Upadrasta is a distinguished cyber security architect with 27 years of professional experience, including 21 years specialising in financial services and banking. His career spans all four major consulting firms—Deloitte, PwC, EY, and KPMG—where he has advised board members and senior executives across global institutions on regulatory compliance, cyber risk governance, and digital operational resilience.

Mr. Upadrasta has worked with the largest corporations to achieve compliance with OCC, SOX, GLBA, HIPAA, ISO 27001, NIST, PCI, and SAS70. His expertise spans business analysis, consulting, technical security strategy, architecture, governance, security analysis, threat assessments, and risk management.

Professional Memberships & Affiliations

- Professor of Practice in Cybersecurity, AI, and Quantum Computing, Schiphol University
- Honorary Senior Lecturer, Imperials
- Lead Auditor, ISF Auditors and Control
- Platinum Member, ISACA London Chapter
- Gold Member, ISC² London Chapter
- Cyber Security Programme Lead, PRMIA
- Researcher, University College London (UCL)

Contact: info@kieranupadrasta.com | www.kie.ie | linkedin.com/in/kieranupadrasta

References

1. Regulation (EU) 2024/2847 (CRA).

2. Directive (EU) 2022/2555 (NIS2).

3. Regulation (EU) 2022/2554 (DORA).

4. Regulation (EU) 2024/1689 (EU AI Act).

5. ISO/IEC 42001:2023.

6. NIST CSF 2.0, Feb 2024.

7. NIST FIPS 204 (ML-DSA).

8. MITRE ATT&CK; v15.

9. OWASP ASI Top 10, 2025.

10. ISO/IEC 27001:2022.

11. ENISA Threat Landscape 2025.

12. OPA/Rego Documentation.