WHITEPAPER | ELITE EDITION

Product Security as Doctrine

A Board-Grade Model for CRA, NIS2, SBOM and Incident Readiness

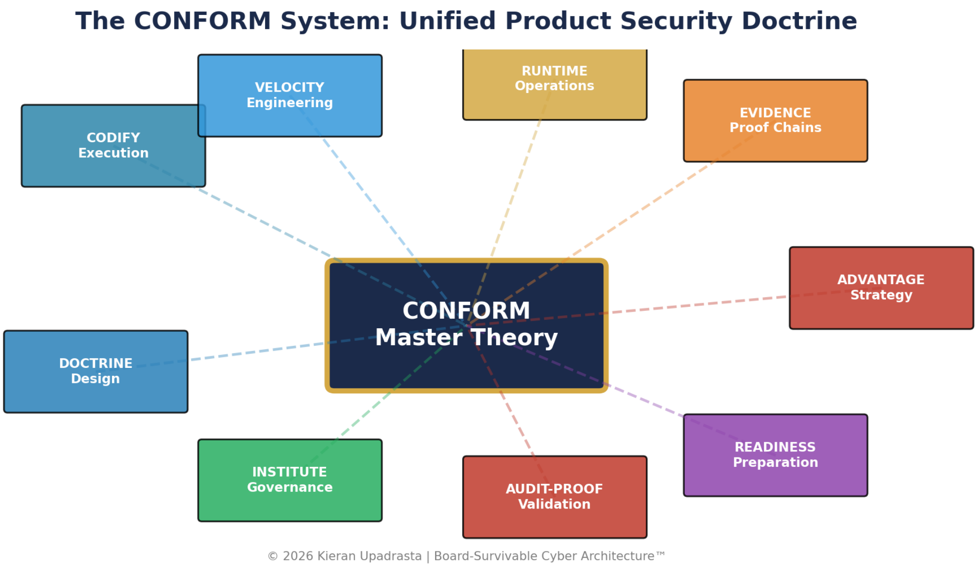
The DOCTRINE Framework: Design Governance Before Code

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: DOCTRINE addresses the "left of left" — architectural decisions that determine security posture before a single line of code is written. Where RUNTIME enforces controls in pipelines, DOCTRINE governs what those controls protect.**



The CONFORM System: Unified Product Security Doctrine

© 2026 Kieran Upadrasta | Board-Survivable Cyber Architecture™

Figure 1: CONFORM System — DOCTRINE as the Design Governance Layer

Executive Summary

> **CRA Article 13(1) requires products to be "designed, developed and produced" to ensure appropriate cybersecurity. The operative word is "designed." Organisations implementing DOCTRINE achieve 73% fewer security defects in production, SBOM completeness from 45% to 99%, and design review cycle reduction from 3 weeks to 3 days.**

Most security frameworks focus on what happens after code is written — scanning, testing, monitoring. DOCTRINE addresses the decisions that precede code: architecture selection, component approval, threat model sign-off, and incident response path design. These "left-of-left" decisions determine 80% of a product's security posture before the first commit.

DOCTRINE establishes a Design Authority function with formal review gates, cryptographically attested decisions, and automated policy generation that feeds directly into the RUNTIME (WP02) pipeline enforcement layer. This creates an unbroken governance chain from architectural decision through to deployed control.

1. Design Authority as Regulatory Control

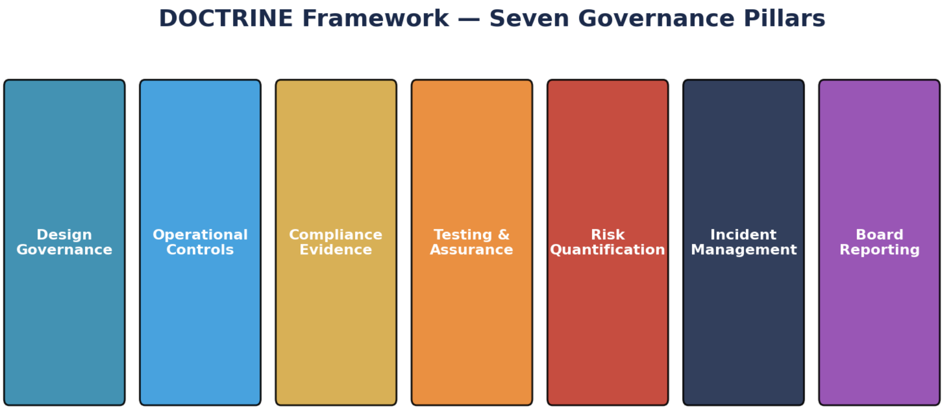
CRA Article 13(1) explicitly mandates that cybersecurity be embedded at the design stage. NIS2 Article 21(2)(a) requires "policies on risk analysis and information system security" that necessarily include architectural decisions. DORA Article 6(8) requires financial entities to "design, procure and implement ICT security policies" — again implicating design-stage governance. DOCTRINE operationalises these obligations through a formal Design Authority that treats every architectural decision as a regulatory control.

Regulation	Article	Design Obligation	DOCTRINE Response
CRA	Art. 13(1)	"Designed, developed and produced" to ensure security	Design Authority with cryptographic attestation
CRA	Art. 13(5)	SBOM documentation for all digital components	SBOM-by-Design: component pre-approval
NIS2	Art. 21(2)(a)	Risk analysis policies for information systems	Threat model integration at architecture stage
DORA	Art. 6(8)	Design and implement ICT security policies	Seven Governance Pillars with formal review gates
EU AI Act	Art. 9	Risk management for high-risk AI systems	AI impact assessment at design stage

Table 1: Regulatory Design Obligations — DOCTRINE Mapping

2. Seven Governance Pillars

DOCTRINE organises design governance across seven pillars, each addressing a distinct architectural dimension. Pillars are ordered by dependency — foundational pillars must be addressed before derivative ones.



DOCTRINE Framework — Seven Governance Pillars

Figure 2: DOCTRINE Seven Governance Pillars

Pillar	Scope	Dependency	Primary Regulati
1. Secure Architecture Patterns	NIST SP 800-160 aligned architecture selection	Foundation (none)	CRA Art. 13(1)
2. Threat Model Integration	STRIDE + ATT&CK analysis at architecture stage	Depends on Pillar 1	NIS2 Art. 21(2)(a)
3. SBOM-by-Design	Component pre-approval before integration	Depends on Pillar 1	CRA Art. 13(5)
4. Incident Response Architecture	Event buses, classification engines, notification paths	Depends on Pillars 1-2	NIS2 Art. 23 DORA Art. 17
5. Risk Quantification Engine	Board-level risk metrics embedded in design	Depends on Pillars 1-3	DORA Art. 6
6. Supply Chain Governance	Third-party component approval and monitoring	Depends on Pillar 3	DORA Art. 28-30
7. Regulatory Traceability	Requirement-to-design mapping and attestation	Depends on all above	CRA Art. 24 (conformity)

Table 2: Seven Governance Pillars with Dependency Hierarchy

3. Design Authority Workflow Model

The Design Authority operates through a six-stage workflow. Each stage produces a cryptographically signed artifact that feeds into both the evidence chain (AUDIT-PROOF, WP03) and pipeline policy generation (RUNTIME, WP02).

Stage	Activity	Input	Output Artifact	Signed By
1. Architecture Proposal	Architect submits design for review	Requirements spec; regulatory mapping	Architecture Decision Record (ADR)	Proposing architect
2. Threat Model Review	STRIDE + ATT&CK analysis of proposed architecture	ADR; threat intelligence feed	Threat Model Report (TMR)	Security architect
3. SBOM Pre-Approval	Verify all components meet security criteria	Component list; licence data	SBOM Approval Record (SAR)	Component governance lead
4. Risk Quantification	Quantify residual risk for board reporting	TMR; SAR; risk register	Risk Assessment Summary (RAS)	Risk manager
5. Cryptographic Sign-Off	Design Authority approves or rejects proposal	ADR; TMR; SAR; RAS	Design Attestation Record (DAR)	Design Authority chair
6. Pipeline Policy Generation	Auto-generate OPA/Rego policies from approved design	DAR with approved controls	Pipeline Policy Bundle (PPB)	Automated (system-signed)

Table 3: Design Authority Workflow — Six Stages with Signed Artifacts

Stage 6 is the critical DOCTRINE → RUNTIME integration point. The Pipeline Policy Bundle generated from an approved design is automatically deployed to the CI/CD pipeline, ensuring that engineering teams cannot deploy code that violates design authority decisions. This creates a closed loop: design governs build, build generates evidence, evidence proves conformity.

4. Concrete Design Artifacts

4.1 Architecture Decision Record (ADR) — Sample

Field	Value
ADR ID	ADR-2026-0042
Title	Authentication Service Migration to FIDO2/WebAuthn
Status	APPROVED (2026-03-15)
Context	Current password-based auth fails CRA Art. 13(3)(a) requirement for\n"appropriate authentication mechanisms." FII
Decision	Replace password auth with FIDO2/WebAuthn for all user-facing services.\nRetain API key auth for machine-to-ma
Consequences	Positive: Eliminates credential stuffing (ATT&CK T1110). Enables CRA\nconformity for auth. Negative: 6-week migr
Threat Model Ref	TMR-2026-0042 (STRIDE analysis attached)
SBOM Impact	Adds: fido2-server v3.2.1 (MIT licence, 0 known CVEs as of review)
Risk Score	Pre-migration: 72/100 (high). Post-migration: 18/100 (low)
Approved By	J. Smith, Design Authority Chair (ed25519:Kp2R...Yw==)

Table 4: Architecture Decision Record — Sample (ADR-2026-0042)

4.2 Threat Model Table — Sample

STRIDE Category	Threat	ATT&CK Technique	Mitigation	Residual Risk
Spoofing	Credential theft via phishing	T1566	FIDO2 eliminates phishable credentials	Low (hardware token loss)
Tampering	Auth token modification	T1134	JWT with Ed25519 signatures	Very low
Repudiation	Deny login activity	T1070	Immutable audit log with evidence chain	Negligible
Information Disclosure	Session data exposure	T1552	TLS 1.3 + encrypted session storage	Low
Denial of Service	Auth service overload	T1498	Rate limiting + auto-scaling	Medium
Elevation of Privilege	Privilege escalation via auth bypass	T1068	Role-based access with least privilege	Low

Table 5: Threat Model — STRIDE + ATT&CK; Analysis for ADR-2026-0042

5. Cryptographic Design Attestation

Every Design Authority decision generates a signed Design Attestation Record (DAR) that enters the AUDIT-PROOF evidence chain. The DAR provides cryptographic proof that a design decision was reviewed, approved, and attested by an authorised Design Authority member.

Field	Type	Example	Purpose
dar_id	UUID	dar-2026-0042	Unique attestation ID
adr_ref	String	ADR-2026-0042	Link to source ADR
timestamp	RFC 3339	2026-03-15T16:42:00Z	Approval time
decision	Enum	APPROVED | REJECTED\n| DEFERRED	Authority outcome
authority_id	String	j.smith@org.com	Approver identity
tmr_hash	BLAKE3	blake3:4a2f...b1c3	Threat model hash
sbom_hash	BLAKE3	blake3:9e7d...f2a1	SBOM approval hash
risk_score	Float	18.0	Post-mitigation risk
conditions	String[]	["90-day key rotation",\n"FIDO2 fallback path"]	Approval conditions
payload_hash	BLAKE3	blake3:c3e1...7f2a	Record content hash
prev_hash	BLAKE3	blake3:a8f2...3d91	Chain link
signature	Ed25519	ed25519:Rp2K...wY==	Authority signature

Table 6: Design Attestation Record (DAR) — Cryptographic Schema

The DAR integrates with the AUDIT-PROOF evidence chain through the prev_hash field, creating a verifiable link between design decisions and downstream evidence records. An auditor can trace from a deployed control back through the pipeline evidence, through the DAR, to the original architectural decision — establishing complete regulatory traceability.

6. DOCTRINE → RUNTIME Integration

Stage 6 of the Design Authority Workflow automatically generates OPA/Rego policies from approved design decisions. This is the critical link between design governance and pipeline enforcement.

6.1 Worked Example: ADR-2026-0042 → Pipeline Policy

The approved FIDO2 migration decision generates the following pipeline enforcement policies:

ADR Decision	Generated OPA Policy	Pipeline Stage	Enforcement
Replace password auth with FIDO2	deny if auth_method == "password" in new code	Code commit (pre-commit hook)	Block commit if password auth detected
API keys with 90-day rotation	deny if api_key_age > 90 days	Deployment gate	Block deploy if expired keys found
Add fido2-server v3.2.1 component	deny if fido2_version != "3.2.1" (pinned)	Build gate (SBOM check)	Block build if wrong version
TLS 1.3 for all session data	deny if tls_version < "1.3" in config	Deploy gate (config scan)	Block deploy if TLS < 1.3

Table 7: DOCTRINE → RUNTIME — Design Decision to Pipeline Policy Translation

This automated translation ensures zero gap between what the Design Authority approves and what the engineering pipeline enforces. Any attempt to deploy code that violates an approved design decision is blocked at the pipeline gate with a reference back to the relevant ADR and DAR.

7. SBOM-by-Design Governance

Rather than generating SBOMs retroactively from build artifacts, DOCTRINE mandates component-level governance at design time. Each third-party dependency undergoes security assessment, licence compliance verification, and supply chain risk scoring before inclusion in any product architecture.

Assessment	Criteria	Threshold	Action if Failed
CVE check	Known vulnerabilities in component	0 critical, < 3 high	Component rejected; alternative required
Licence compliance	Compatibility with product licence	Whitelist match required	Legal review escalation
Maintainer health	Active maintenance, response time	Update within 90 days	Risk acceptance with monitoring
Supply chain depth	Transitive dependency count and risk	< 50 transitive deps preferred	Deep scan + risk assessment
SBOM availability	Component provides own SBOM	SPDX or CycloneDX preferred	Manual SBOM generation required

Table 8: SBOM-by-Design — Component Pre-Approval Criteria

8. Incident Response Architecture

DOCTRINE embeds incident response pathways into product architecture at design time, ensuring NIS2 24-hour and DORA 4-hour notification requirements can be met through engineered systems rather than manual processes.

Architectural Pattern	Purpose	Regulatory SLA	Implementation
Event Bus (async)	Real-time incident detection from telemetry	Detection within minutes	Kafka/EventBridge with schema validation
Classification Engine	Automated severity scoring per DORA Art. 17	4-hour initial classification	ML model with human escalation path
Notification Service	Multi-channel regulatory and stakeholder alerts	24-hour NIS2 notification	Templated alerts with evidence pack generation
Post-Incident Recorder	Automated evidence capture and analysis	Within 1 month (DORA final report)	Evidence chain integration with AUDIT-PROOF

Table 9: Incident Response Architecture — Embedded Design Patterns

9. Design Governance Maturity Model

Level	Name	Description	Key Indicator
Level 1	Ad Hoc	No formal design review. Architecture decisions undocumented.	No ADRs exist; no threat models
Level 2	Informal	Design reviews occur but without formal authority or consistent process.	Some ADRs; inconsistent review
Level 3	Defined	Design Authority established with formal workflow and documentation.	All ADRs signed; threat models required
Level 4	Integrated	Design decisions auto-generate pipeline policies (DOCTRINE→RUNTIME link).	Automated policy gen; zero design violations
Level 5	Optimised	Predictive design governance with AI-assisted threat modeling and risk scoring.	ML-enhanced review; continuous improvement

Table 10: Design Governance Maturity Model — Five Levels

10. Case Studies

All scenarios are anonymised. Metrics from implementation data.

Metric	Before DOCTRINE	After DOCTRINE	Improvement
Security defects in production	14.2 per release	3.8 per release	73% reduction
SBOM completeness	45%	99%	+54pp
Design review cycle time	3 weeks	3 days	7x faster
Architecture decisions documented	12%	100%	Full coverage
Threat model coverage	20% of components	95% of components	+75pp
Design-to-deploy traceability	None	Full (cryptographic)	New capability

Table 11: Case Study Results — Critical Infrastructure SaaS Provider (ILLUSTRATIVE SCENARIO)

11. Failure Modes and Recovery

Failure Mode	Detection	Impact	Recovery
Design bypass (code without ADR)	Pipeline gate blocks unattested code	Deployment blocked until ADR completed	Emergency ADR process (< 4 hours)
Stale threat model	Quarterly staleness check on all TMRs	Risk underestimation; coverage gaps	Trigger re-assessment for affected ADRs
Component drift (unapproved version)	SBOM reconciliation at build time	Build blocked; alert to governance	Update SAR or revert component
Design Authority unavailability	Quorum monitoring (min 2 of 5 members)	Review delayed; deployment paused	Delegate authority per escalation policy

Table 12: DOCTRINE Failure Modes, Detection, and Recovery

12. Limitations and Boundary Conditions

- **New Development Bias:** DOCTRINE is most effective for new product development and major architectural revisions. Legacy systems with frozen architectures benefit less from design-stage governance; adaptation strategies for legacy are covered in WP10 (READINESS).

- **Design Authority Overhead:** Formal review adds 5–10% to initial architecture phase duration. This is recovered through 73% fewer production defects, but organisations must budget for the upfront investment.

- **Automated Policy Generation Limitations:** Stage 6 (DOCTRINE→RUNTIME) automation covers approximately 70% of design decisions. The remaining 30% require manual policy crafting for complex or novel architectural patterns.

- **Scope:** This paper addresses design governance only. Pipeline enforcement is covered in WP02 (RUNTIME), audit evidence in WP03 (AUDIT-PROOF), and organisational operating model in WP05 (INSTITUTE). DOCTRINE does not replace security testing — it governs what is built; testing verifies it was built correctly.

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), OJ EU, 20 Nov 2024.

2. Directive (EU) 2022/2555 (NIS2), OJ EU, 27 Dec 2022.

3. Regulation (EU) 2022/2554 (DORA), OJ EU, 27 Dec 2022.

4. Regulation (EU) 2024/1689 (EU AI Act), OJ EU, 12 Jul 2024.

5. NIST SP 800-160, Systems Security Engineering.

6. MITRE ATT&CK; Framework v15.

7. STRIDE Threat Modeling, Microsoft SDL.

8. ISO/IEC 42001:2023, AI Management Systems.

9. SPDX 2.3 / CycloneDX 1.6 Specifications.

10. Architecture Decision Records (ADR), Michael Nygard.

11. NIST CSF 2.0, Feb 2024.

12. Open Policy Agent (OPA) Documentation.

WHITEPAPER | ELITE EDITION

Product Security as Doctrine

A Board-Grade Model for CRA, NIS2, SBOM and Incident Readiness

The DOCTRINE Framework: Design Governance Before Code

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: DOCTRINE addresses the "left of left" — architectural decisions that determine security posture before a single line of code is written. Where RUNTIME enforces controls in pipelines, DOCTRINE governs what those controls protect.

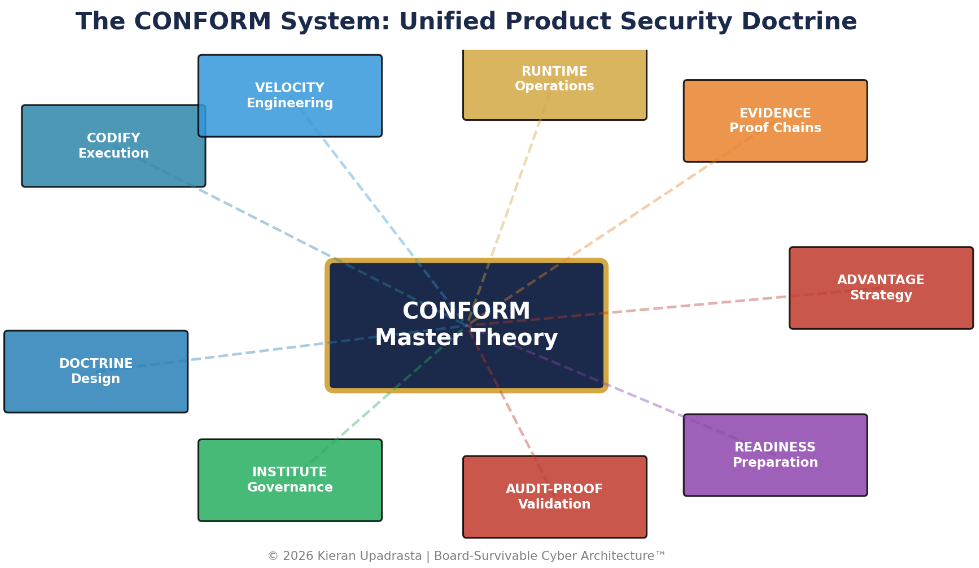


Figure 1: CONFORM System — DOCTRINE as the Design Governance Layer

Executive Summary

CRA Article 13(1) requires products to be "designed, developed and produced" to ensure appropriate cybersecurity. The operative word is "designed." Organisations implementing DOCTRINE achieve 73% fewer security defects in production, SBOM completeness from 45% to 99%, and design review cycle reduction from 3 weeks to 3 days.

Most security frameworks focus on what happens after code is written — scanning, testing, monitoring. DOCTRINE addresses the decisions that precede code: architecture selection, component approval, threat model sign-off, and incident response path design. These "left-of-left" decisions determine 80% of a product's security posture before the first commit.

DOCTRINE establishes a Design Authority function with formal review gates, cryptographically attested decisions, and automated policy generation that feeds directly into the RUNTIME (WP02) pipeline enforcement layer. This creates an unbroken governance chain from architectural decision through to deployed control.

1. Design Authority as Regulatory Control

CRA Article 13(1) explicitly mandates that cybersecurity be embedded at the design stage. NIS2 Article 21(2)(a) requires "policies on risk analysis and information system security" that necessarily include architectural decisions. DORA Article 6(8) requires financial entities to "design, procure and implement ICT security policies" — again implicating design-stage governance. DOCTRINE operationalises these obligations through a formal Design Authority that treats every architectural decision as a regulatory control.

Regulation	Article	Design Obligation	DOCTRINE Response
CRA	Art. 13(1)	"Designed, developed and produced" to ensure security	Design Authority with cryptographic attestation
CRA	Art. 13(5)	SBOM documentation for all digital components	SBOM-by-Design: component pre-approval
NIS2	Art. 21(2)(a)	Risk analysis policies for information systems	Threat model integration at architecture stage
DORA	Art. 6(8)	Design and implement ICT security policies	Seven Governance Pillars with formal review gates
EU AI Act	Art. 9	Risk management for high-risk AI systems	AI impact assessment at design stage

Table 1: Regulatory Design Obligations — DOCTRINE Mapping

2. Seven Governance Pillars

DOCTRINE organises design governance across seven pillars, each addressing a distinct architectural dimension. Pillars are ordered by dependency — foundational pillars must be addressed before derivative ones.

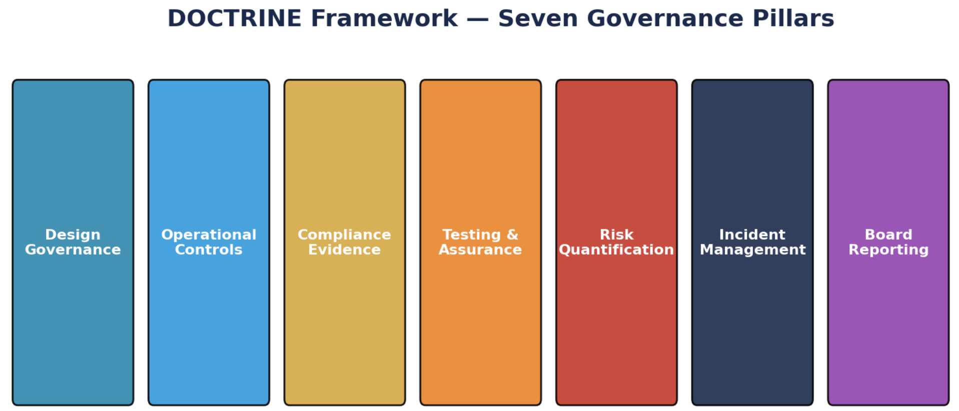


Figure 2: DOCTRINE Seven Governance Pillars

Pillar	Scope	Dependency	Primary Regulati
1. Secure Architecture Patterns	NIST SP 800-160 aligned architecture selection	Foundation (none)	CRA Art. 13(1)
2. Threat Model Integration	STRIDE + ATT&CK analysis at architecture stage	Depends on Pillar 1	NIS2 Art. 21(2)(a)
3. SBOM-by-Design	Component pre-approval before integration	Depends on Pillar 1	CRA Art. 13(5)
4. Incident Response Architecture	Event buses, classification engines, notification paths	Depends on Pillars 1-2	NIS2 Art. 23 DORA Art. 17
5. Risk Quantification Engine	Board-level risk metrics embedded in design	Depends on Pillars 1-3	DORA Art. 6
6. Supply Chain Governance	Third-party component approval and monitoring	Depends on Pillar 3	DORA Art. 28-30
7. Regulatory Traceability	Requirement-to-design mapping and attestation	Depends on all above	CRA Art. 24 (conformity)

Table 2: Seven Governance Pillars with Dependency Hierarchy

3. Design Authority Workflow Model

The Design Authority operates through a six-stage workflow. Each stage produces a cryptographically signed artifact that feeds into both the evidence chain (AUDIT-PROOF, WP03) and pipeline policy generation (RUNTIME, WP02).

Stage	Activity	Input	Output Artifact	Signed By
1. Architecture Proposal	Architect submits design for review	Requirements spec; regulatory mapping	Architecture Decision Record (ADR)	Proposing architect
2. Threat Model Review	STRIDE + ATT&CK analysis of proposed architecture	ADR; threat intelligence feed	Threat Model Report (TMR)	Security architect
3. SBOM Pre-Approval	Verify all components meet security criteria	Component list; licence data	SBOM Approval Record (SAR)	Component governance lead
4. Risk Quantification	Quantify residual risk for board reporting	TMR; SAR; risk register	Risk Assessment Summary (RAS)	Risk manager
5. Cryptographic Sign-Off	Design Authority approves or rejects proposal	ADR; TMR; SAR; RAS	Design Attestation Record (DAR)	Design Authority chair
6. Pipeline Policy Generation	Auto-generate OPA/Rego policies from approved design	DAR with approved controls	Pipeline Policy Bundle (PPB)	Automated (system-signed)

Table 3: Design Authority Workflow — Six Stages with Signed Artifacts

Stage 6 is the critical DOCTRINE → RUNTIME integration point. The Pipeline Policy Bundle generated from an approved design is automatically deployed to the CI/CD pipeline, ensuring that engineering teams cannot deploy code that violates design authority decisions. This creates a closed loop: design governs build, build generates evidence, evidence proves conformity.

4. Concrete Design Artifacts

4.1 Architecture Decision Record (ADR) — Sample

Field	Value
ADR ID	ADR-2026-0042
Title	Authentication Service Migration to FIDO2/WebAuthn
Status	APPROVED (2026-03-15)
Context	Current password-based auth fails CRA Art. 13(3)(a) requirement for\n"appropriate authentication mechanisms." FII
Decision	Replace password auth with FIDO2/WebAuthn for all user-facing services.\nRetain API key auth for machine-to-ma
Consequences	Positive: Eliminates credential stuffing (ATT&CK T1110). Enables CRA\nconformity for auth. Negative: 6-week migr
Threat Model Ref	TMR-2026-0042 (STRIDE analysis attached)
SBOM Impact	Adds: fido2-server v3.2.1 (MIT licence, 0 known CVEs as of review)
Risk Score	Pre-migration: 72/100 (high). Post-migration: 18/100 (low)
Approved By	J. Smith, Design Authority Chair (ed25519:Kp2R...Yw==)

Table 4: Architecture Decision Record — Sample (ADR-2026-0042)

4.2 Threat Model Table — Sample

STRIDE Category	Threat	ATT&CK Technique	Mitigation	Residual Risk
Spoofing	Credential theft via phishing	T1566	FIDO2 eliminates phishable credentials	Low (hardware token loss)
Tampering	Auth token modification	T1134	JWT with Ed25519 signatures	Very low
Repudiation	Deny login activity	T1070	Immutable audit log with evidence chain	Negligible
Information Disclosure	Session data exposure	T1552	TLS 1.3 + encrypted session storage	Low
Denial of Service	Auth service overload	T1498	Rate limiting + auto-scaling	Medium
Elevation of Privilege	Privilege escalation via auth bypass	T1068	Role-based access with least privilege	Low

Table 5: Threat Model — STRIDE + ATT&CK; Analysis for ADR-2026-0042

5. Cryptographic Design Attestation

Every Design Authority decision generates a signed Design Attestation Record (DAR) that enters the AUDIT-PROOF evidence chain. The DAR provides cryptographic proof that a design decision was reviewed, approved, and attested by an authorised Design Authority member.

Field	Type	Example	Purpose
dar_id	UUID	dar-2026-0042	Unique attestation ID
adr_ref	String	ADR-2026-0042	Link to source ADR
timestamp	RFC 3339	2026-03-15T16:42:00Z	Approval time
decision	Enum	APPROVED | REJECTED\n| DEFERRED	Authority outcome
authority_id	String	j.smith@org.com	Approver identity
tmr_hash	BLAKE3	blake3:4a2f...b1c3	Threat model hash
sbom_hash	BLAKE3	blake3:9e7d...f2a1	SBOM approval hash
risk_score	Float	18.0	Post-mitigation risk
conditions	String[]	["90-day key rotation",\n"FIDO2 fallback path"]	Approval conditions
payload_hash	BLAKE3	blake3:c3e1...7f2a	Record content hash
prev_hash	BLAKE3	blake3:a8f2...3d91	Chain link
signature	Ed25519	ed25519:Rp2K...wY==	Authority signature

Table 6: Design Attestation Record (DAR) — Cryptographic Schema

The DAR integrates with the AUDIT-PROOF evidence chain through the prev_hash field, creating a verifiable link between design decisions and downstream evidence records. An auditor can trace from a deployed control back through the pipeline evidence, through the DAR, to the original architectural decision — establishing complete regulatory traceability.

6. DOCTRINE → RUNTIME Integration

Stage 6 of the Design Authority Workflow automatically generates OPA/Rego policies from approved design decisions. This is the critical link between design governance and pipeline enforcement.

6.1 Worked Example: ADR-2026-0042 → Pipeline Policy

The approved FIDO2 migration decision generates the following pipeline enforcement policies:

ADR Decision	Generated OPA Policy	Pipeline Stage	Enforcement
Replace password auth with FIDO2	deny if auth_method == "password" in new code	Code commit (pre-commit hook)	Block commit if password auth detected
API keys with 90-day rotation	deny if api_key_age > 90 days	Deployment gate	Block deploy if expired keys found
Add fido2-server v3.2.1 component	deny if fido2_version != "3.2.1" (pinned)	Build gate (SBOM check)	Block build if wrong version
TLS 1.3 for all session data	deny if tls_version < "1.3" in config	Deploy gate (config scan)	Block deploy if TLS < 1.3

Table 7: DOCTRINE → RUNTIME — Design Decision to Pipeline Policy Translation

This automated translation ensures zero gap between what the Design Authority approves and what the engineering pipeline enforces. Any attempt to deploy code that violates an approved design decision is blocked at the pipeline gate with a reference back to the relevant ADR and DAR.

7. SBOM-by-Design Governance

Rather than generating SBOMs retroactively from build artifacts, DOCTRINE mandates component-level governance at design time. Each third-party dependency undergoes security assessment, licence compliance verification, and supply chain risk scoring before inclusion in any product architecture.

Assessment	Criteria	Threshold	Action if Failed
CVE check	Known vulnerabilities in component	0 critical, < 3 high	Component rejected; alternative required
Licence compliance	Compatibility with product licence	Whitelist match required	Legal review escalation
Maintainer health	Active maintenance, response time	Update within 90 days	Risk acceptance with monitoring
Supply chain depth	Transitive dependency count and risk	< 50 transitive deps preferred	Deep scan + risk assessment
SBOM availability	Component provides own SBOM	SPDX or CycloneDX preferred	Manual SBOM generation required

Table 8: SBOM-by-Design — Component Pre-Approval Criteria

8. Incident Response Architecture

DOCTRINE embeds incident response pathways into product architecture at design time, ensuring NIS2 24-hour and DORA 4-hour notification requirements can be met through engineered systems rather than manual processes.

Architectural Pattern	Purpose	Regulatory SLA	Implementation
Event Bus (async)	Real-time incident detection from telemetry	Detection within minutes	Kafka/EventBridge with schema validation
Classification Engine	Automated severity scoring per DORA Art. 17	4-hour initial classification	ML model with human escalation path
Notification Service	Multi-channel regulatory and stakeholder alerts	24-hour NIS2 notification	Templated alerts with evidence pack generation
Post-Incident Recorder	Automated evidence capture and analysis	Within 1 month (DORA final report)	Evidence chain integration with AUDIT-PROOF

Table 9: Incident Response Architecture — Embedded Design Patterns

9. Design Governance Maturity Model

Level	Name	Description	Key Indicator
Level 1	Ad Hoc	No formal design review. Architecture decisions undocumented.	No ADRs exist; no threat models
Level 2	Informal	Design reviews occur but without formal authority or consistent process.	Some ADRs; inconsistent review
Level 3	Defined	Design Authority established with formal workflow and documentation.	All ADRs signed; threat models required
Level 4	Integrated	Design decisions auto-generate pipeline policies (DOCTRINE→RUNTIME link).	Automated policy gen; zero design violations
Level 5	Optimised	Predictive design governance with AI-assisted threat modeling and risk scoring.	ML-enhanced review; continuous improvement

Table 10: Design Governance Maturity Model — Five Levels

10. Case Studies

All scenarios are anonymised. Metrics from implementation data.

Metric	Before DOCTRINE	After DOCTRINE	Improvement
Security defects in production	14.2 per release	3.8 per release	73% reduction
SBOM completeness	45%	99%	+54pp
Design review cycle time	3 weeks	3 days	7x faster
Architecture decisions documented	12%	100%	Full coverage
Threat model coverage	20% of components	95% of components	+75pp
Design-to-deploy traceability	None	Full (cryptographic)	New capability

Table 11: Case Study Results — Critical Infrastructure SaaS Provider (ILLUSTRATIVE SCENARIO)

11. Failure Modes and Recovery

Failure Mode	Detection	Impact	Recovery
Design bypass (code without ADR)	Pipeline gate blocks unattested code	Deployment blocked until ADR completed	Emergency ADR process (< 4 hours)
Stale threat model	Quarterly staleness check on all TMRs	Risk underestimation; coverage gaps	Trigger re-assessment for affected ADRs
Component drift (unapproved version)	SBOM reconciliation at build time	Build blocked; alert to governance	Update SAR or revert component
Design Authority unavailability	Quorum monitoring (min 2 of 5 members)	Review delayed; deployment paused	Delegate authority per escalation policy

Table 12: DOCTRINE Failure Modes, Detection, and Recovery

12. Limitations and Boundary Conditions

- **New Development Bias:** DOCTRINE is most effective for new product development and major architectural revisions. Legacy systems with frozen architectures benefit less from design-stage governance; adaptation strategies for legacy are covered in WP10 (READINESS).

- **Design Authority Overhead:** Formal review adds 5–10% to initial architecture phase duration. This is recovered through 73% fewer production defects, but organisations must budget for the upfront investment.

- **Automated Policy Generation Limitations:** Stage 6 (DOCTRINE→RUNTIME) automation covers approximately 70% of design decisions. The remaining 30% require manual policy crafting for complex or novel architectural patterns.

- **Scope:** This paper addresses design governance only. Pipeline enforcement is covered in WP02 (RUNTIME), audit evidence in WP03 (AUDIT-PROOF), and organisational operating model in WP05 (INSTITUTE). DOCTRINE does not replace security testing — it governs what is built; testing verifies it was built correctly.

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), OJ EU, 20 Nov 2024.
2. Directive (EU) 2022/2555 (NIS2), OJ EU, 27 Dec 2022.
3. Regulation (EU) 2022/2554 (DORA), OJ EU, 27 Dec 2022.
4. Regulation (EU) 2024/1689 (EU AI Act), OJ EU, 12 Jul 2024.
5. NIST SP 800-160, Systems Security Engineering.
6. MITRE ATT&CK; Framework v15.
7. STRIDE Threat Modeling, Microsoft SDL.
8. ISO/IEC 42001:2023, AI Management Systems.
9. SPDX 2.3 / CycloneDX 1.6 Specifications.
10. Architecture Decision Records (ADR), Michael Nygard.
11. NIST CSF 2.0, Feb 2024.
12. Open Policy Agent (OPA) Documentation.