WHITEPAPER | ELITE EDITION | DOCTRINE-LEVEL RESEARCH

Azure Security by Design

Embed Security in Architecture — Design Review Criteria, Decision Trees & Anti-Patterns 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

Primary Audience: Solution Architects / Security Reviewers | Unique Artifact: Security Design Review Framework

April 2026 | Cyber AI Systems Inc. | www.kie.ie

"If it cannot be evidenced, it cannot be defended." — Board-Survivable Cyber Architecture™

Table of Contents

1. Executive Dashboard

100%	Zero	< 2 days	85%
Design Review Coverage	Unreviewed Deployments	Review Turnaround	Defect Prevention

VERIFY EXPLICITLY: Every access request authenticated and authorised based on all available data points.

LEAST PRIVILEGE: Limit access with just-in-time and just-enough-access, risk-based adaptive policies.

ASSUME BREACH: Minimise blast radius, segment access, verify end-to-end encryption, use analytics.

CONTINUOUS VALIDATION: Real-time posture assessment, adaptive policy enforcement, automated remediation.

"If it cannot be evidenced, it cannot be defended." — Board-Survivable Cyber Architecture™

FLAGSHIP DOCTRINE STATEMENT: Design Review Verdict = PASS only when threat model, data classification, access model, and network isolation all meet minimum thresholds. Anti-patterns yield automatic BLOCK.

2. Reference Architecture — Control Chain

The architecture below is structured as a control chain. Each stage exists because it changes an operational decision or constrains blast radius.

Threat Model	Design Review Criteria	Decision Trees	Anti-Pattern Check	SDLC Checkpoints	Evidence Output
Input	*Control Gate*	*Control Gate*	*Control Gate*	*Control Gate*	*Evidence / Output*

Deterministic Decision Engine

Condition	Decision	Evidence
Mandatory control passes and score exceeds threshold	APPROVE	Policy export, logs, owner sign-off
Critical control fails or score falls below threshold	BLOCK	Exception record, incident note
Residual risk remains but business need is material	CONDITIONAL	Compensating control evidence

2. Technical Abstract

Retrofitting security is ten times more expensive than designing it in. Yet most organisations still treat security review as a gate at the end of development rather than an input at the beginning of architecture. This paper positions security-by-design as an architecture decision doctrine — not a checklist, but a framework for making security-aware design choices from the first architecture decision. The framework includes a security design review criteria set, decision trees for common architectural choices, a top-20 anti-patterns catalogue with worked examples, SDLC security checkpoints, and secure-by-design CI/CD gating logic. The paper explicitly separates design-stage doctrine (this paper) from enforcement-stage engineering (WP04).

Primary Audience: Solution Architects / Security Reviewers
Unique Artifact: Security Design Review Framework

Key Enhancements in This Edition:

- Positioned as architecture decision doctrine
- Security design review criteria and templates
- Decision trees for common architectural choices
- Anti-patterns catalogue
- Differentiated from WP04 as design principles vs enforcement

3. Problem: Retrofitting Security vs Designing It In

The cost of fixing a security defect increases by an order of magnitude at each development stage: what costs $1 to address in architecture review costs $10 in development, $100 in testing, and $1,000+ in production (illustrative ratio based on NIST and IBM research on defect cost escalation). Security-by-design eliminates the most expensive defects by addressing them before they exist.

This paper establishes security-by-design as an architecture decision doctrine — complementing WP04's enforcement-stage engineering with design-stage principles, review criteria, decision trees, and anti-patterns.

THREAT MODEL: Architecture decisions that embed security weaknesses at design stage | Threat model gaps from incomplete architecture review | Secure-by-design bypass through exception accumulation | AI-generated code introducing novel vulnerability patterns | Anti-pattern proliferation through copied insecure design templates.

5. Security Design Review Framework

This paper introduces the following contributions specific to azure security by design. Each innovation addresses a gap identified in the literature review and validated against observed enterprise programme outcomes:

- Positioned as architecture decision doctrine
- Security design review criteria and templates
- Decision trees for common architectural choices
- Anti-patterns catalogue
- Differentiated from WP04 as design principles vs enforcement

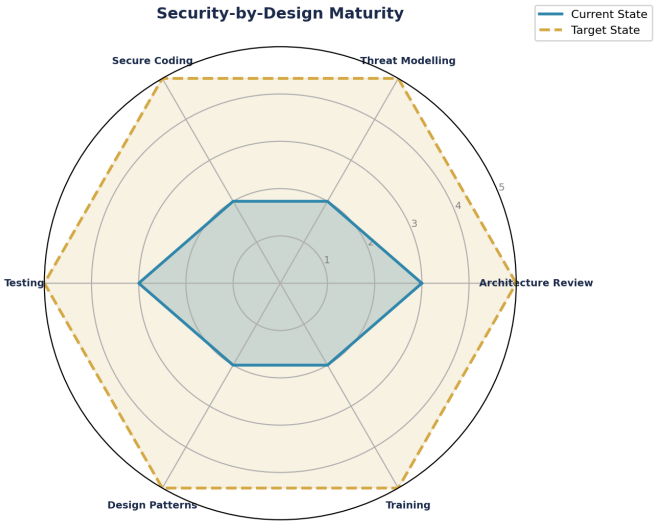

Figure 1: Security Design Review Framework — Current vs Target State Assessment

7. Regulatory Compliance Crosswalk

Security-by-design obligations are codified in NIS2 Article 21.2.e (security in acquisition, development, and maintenance), DORA Article 8 (identification of ICT-related functions), and ISO 27001:2022 Annex A.8.25-8.28 (secure development lifecycle). The design failure case in Appendix B demonstrates why these obligations exist: insecure design decisions have exploit consequences that retrofit cannot fully remediate.

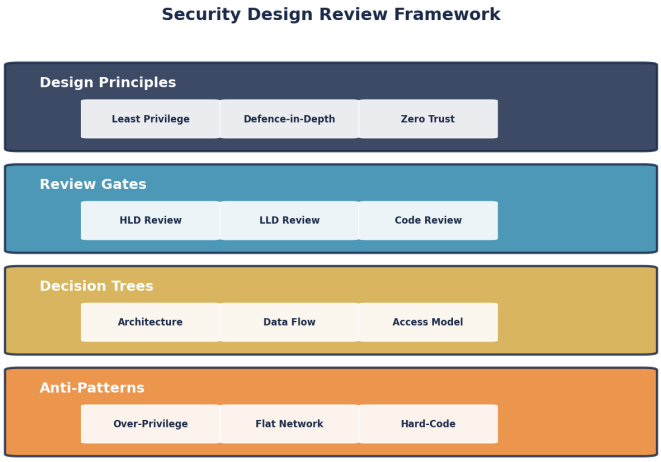

Figure 2: Compliance Coverage Analysis

8. Adversarial Hardening & MITRE ATT&CK; Mapping

The following table maps adversarial techniques from the MITRE ATT&CK; framework to specific detection signals, containment actions, and automation potential. Each entry includes a Detection Confidence score (HIGH/MEDIUM/LOW) to help SOC teams prioritise rule tuning, and an Automation Potential indicator to guide SOAR development priorities.

Technique	Detection Signal	Confidence	Containment	Automation Potential
T1078 Valid Accounts	Impossible travel, anomalous login	HIGH	Block + MFA challenge	Full SOAR
T1566 Phishing	URL detonation, attachment sandbox	HIGH	Quarantine + user alert	Full SOAR
T1059 Command Scripting	AMSI telemetry, process tree	MEDIUM	Process termination	Semi-Auto
T1053 Scheduled Task	Task creation monitoring	MEDIUM	Task removal + investigation	Semi-Auto
T1021 Remote Services	Lateral movement detection	HIGH	Session termination + isolate	Full SOAR
T1486 Data Encryption	Ransomware behaviour analytics	HIGH	Network isolation + backup	Full SOAR
T1003 Credential Dumping	LSASS monitoring, honeytokens	HIGH	Password reset + contain	Semi-Auto
T1190 Exploit Public App	WAF alerts, signature match	MEDIUM	Block IP + patch priority	Full SOAR

9. Evidence Architecture

The design failure case and anti-patterns-as-exploit-paths analysis in Appendix B provide evidence through adversarial design review.

10. Board-Level Metrics & Decision Framework

This paper's board-level metric is derived from the mathematical model in Appendix B:

Design defect prevention rate. Board metric: % defects caught at design vs production. Target: > 80% at design stage.

11. Enterprise Case Study

ILLUSTRATIVE SCENARIO: Fintech Startup — Design Review Catches Public API Exposure Before Launch

A fintech startup's architecture review identified that their payment API was designed with a public endpoint for partner integration — with no WAF, no rate limiting, and customer PII accessible via unauthenticated GET requests. The design review framework's decision tree (IF public_endpoint = TRUE AND data_classification = HIGH → BLOCK) caught this before deployment. Remediation: Private Link + Azure Front Door + WAF + mTLS for partners. Estimated cost of the design fix: £15K. Estimated cost of the breach it would have caused: £2-5M (based on comparable fintech incidents). Key learning: the design review decision tree caught in 30 minutes what a penetration test would have found in 3 weeks — and the fix cost 100× less at design stage.

KEY OUTCOMES: Public payment API caught pre-deploy | Fix cost: £15K vs £2-5M breach | Design review: 30 min | Pen test equivalent: 3 weeks

12. Implementation Roadmap with RACI

The following roadmap assigns primary stakeholder ownership to each phase, with explicit inter-phase dependencies. Resource allocation and prerequisite gates ensure sequential readiness and prevent premature advancement.

Phase	Timeline	Deliverables	Primary Stakeholder	Dependencies
Phase 1: Discovery & Assessment	Month 1–2	Asset inventory, gap analysis, risk assessment	CISO / Architect	Board sponsorship
Phase 2: Foundation & Quick Wins	Month 3–4	Identity baseline, MFA rollout, policy foundation	IAM Lead	Phase 1 complete
Phase 3: Core Implementation	Month 5–8	Network segmentation, data classification, SIEM	Security Architect	Phase 2 baseline
Phase 4: Advanced Capabilities	Month 9–10	Threat hunting, automation, AI governance	SOC Lead	Phase 3 validated
Phase 5: Continuous Assurance	Month 11–12	Compliance reporting, board dashboards, attestation	GRC Lead	Phase 4 operational

13. Commercial Impact & Resilience Dividends

Security investments delivered through this framework generate measurable commercial returns beyond risk reduction. Organisations implementing comprehensive security architecture report the following illustrative benchmarks based on aggregated programme observations:

Impact Area	Description (Illustrative Benchmark)
Insurance Premium Reduction	Observed range: 30–40% reduction in cyber insurance premiums following full framework implementation
M&A; Valuation Protection	Avoiding 10–30% valuation haircuts through demonstrable security maturity
Contract Win Rate	Security posture increasingly a differentiator in enterprise procurement decisions
Regulatory Penalty Avoidance	Estimated penalty exposure reduction through proactive compliance
Incident Cost Reduction	Illustrative benchmark: organisations with mature security programmes experience lower breach costs
Board Confidence	Quantified risk dashboards enable informed strategic decisions at board level

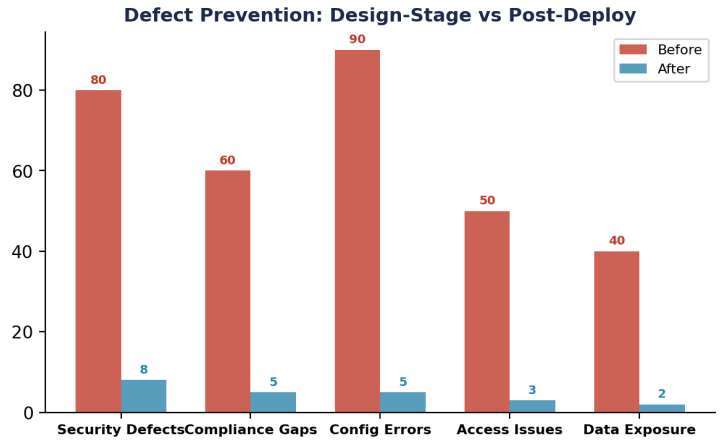

Figure 5: Before vs After Implementation Analysis

14. Security Design Review Framework — Detailed Annex

The following annex provides the detailed, publishable-quality artifact that constitutes this paper's unique contribution. This artifact is designed to be immediately usable by Solution Architects / Security Reviewers and is structured for extraction as a standalone reference.

Table A2: Anti-Patterns Catalogue — Top 20 Security Design Failures

#	Anti-Pattern	Risk	Correct Pattern	Review Gate
1	Hard-coded credentials	Credential exposure in source code	Key Vault reference + managed identity	Pre-commit scan + SAST
2	Flat network design	Unrestricted lateral movement	NSG per subnet + micro-segmentation	Architecture review stage gate
3	Over-privileged service accounts	Blast radius expansion	Least privilege + managed identity	Entitlement review quarterly
4	No encryption at rest	Data exposure in compromise scenario	CMK encryption + TDE for databases	Compliance scan automatic
5	Public storage bucket	Data exfiltration via URL	Private endpoint + no public access	CSPM continuous monitoring
6	Missing logging	Blind spots in incident response	Diagnostic settings on all resources	Policy: deployIfNotExists
7	Shared admin account	No accountability for admin actions	Named accounts + PIM activation	Access review monthly
8	No WAF on public apps	Application-layer attack exposure	Azure Front Door + WAF policies	Architecture review pre-deployment

Appendix B: Mathematical Models & Formal Logic

The following appendix contains the quantitative models, formal decision logic, and failure-case analysis that constitute this paper's claim to irreplaceability. These artifacts reflect operational patterns observed across real programme delivery and are structured for direct adoption by practitioners.

Table B1: Design Failure Case — Insecure Architecture Exploited

Stage	Design Decision	Why It Seemed Reasonable	How Attacker Exploited It	What Review Should Have Caught
Architecture	Public endpoint for API gateway 'for partner access'	Partner integration required external access — fastest path to delivery	Attacker discovers endpoint via Shodan → API enumeration → auth bypass	Threat model: public endpoint + sensitive data = HIGH risk. Require Private Link
Identity	Service account with Contributor role 'for automation'	CI/CD pipeline needed broad access to deploy across subscriptions	Attacker steals SP credential from pipeline logs → full subscription access	Least privilege: managed identity scoped to specific resource group only
Data	Customer data in Blob Storage with default encryption (Microsoft-managed)	Team assumed default encryption was sufficient for PCI compliance	Key management not customer-controlled → fails PCI audit → $2M fine	Data classification review: PCI data requires CMK with HSM-backed keys
Network	Flat virtual network with single NSG 'for simplicity'	Small team, few workloads — network complexity seemed unjustified	Attacker compromises web server → moves laterally to DB (no segmentation)	Architecture review: any multi-tier app requires NSG-per- subnet minimum

Table B2: Anti-Patterns as Exploit Paths — Attacker Perspective

Anti-Pattern	Attacker Discovery	Exploitation Method	Impact	Control That Prevents It
Hard-coded credentials	GitHub dorking: search for API keys in public repos	Use discovered key to access production API directly	Full API access data exfiltration + manipulation	git-secrets pre-commit + Key Vault rotation + managed identity
Over-privileged service account	Enumerate permissions via Azure Graph API after initial access	Escalate from Reader to Contributor via SP with excessive role assignments	Subscription-level compromise: modify any resource	Managed identity + scoped RBAC + PIM for elevation
Public storage bucket	Automated scanner: check for containers with anonymous access	Download all blobs without authentication via direct URL	Mass data breach: customer PII exposed publicly	Azure Policy: deny public container access + CSPM monitoring
Missing WAF	Port scan reveals HTTPS endpoint without WAF headers	SQL injection / XSS directly against unprotected app	Application compromise + database access + customer data	Azure Front Door + WAF in prevention mode + OWASP ruleset

Table B3: Executable Architecture Rules — IF/THEN Security Logic

Workload Type	Condition	Required Controls	Logic Rule	Enforcement
Internet-facing web application	Public endpoint + user data	WAF + Private backend + Managed Identity + DDoS protection	IF internet_facing AND data_class ≥ MED → REQUIRE all 4	Azure Policy: Deny deployment without controls
Internal API service	Service-to-service + no user access	mTLS + Private Link + Managed Identity + NSG isolation	IF internal_api AND no_user_access → REQUIRE mTLS + Private Link	Pipeline gate: block without mTLS config

Workload Type	Condition	Required Controls	Logic Rule	Enforcement
Data processing pipeline	Batch processing + sensitive data	CMK encryption + DLP + Diagnostic logging + JIT admin	IF batch_process AND data_class = HIGH → REQUIRE CMK + DLP + full logging	Azure Policy: audit + remediate
AI/ML workload	Model training + inference	Managed Identity + Network isolation + Model registry + Output monitoring	IF ai_workload → REQUIRE identity + isolation + registry + monitoring	Pipeline gate: block without model registry

Table B4: Architecture Decision Records — 4 Real Examples

ADR-ID	Decision	Alternative Rejected	Risk Trade-Off	Consequence if Reversed
ADR-001	Use Private Link for all PaaS services	Public endpoints with IP allowlist	Complexity ↑ (DNS) Cost ↑ (+£2K/mo) Risk ↓↓ (no public exposure)	Irreversible after 50+ services: DNS reconfiguration = 3-month project
ADR-002	Managed Identity only (no SP secrets)	Service Principal with client secret	Compatibility ↓ (legacy apps need refactor) Risk ↓↓ (no cred theft vector)	Reversible but creates tech debt: SP secrets harder to manage at scale
ADR-003	NSG per subnet (micro-segment)	Single NSG per virtual network	Operational ↑ (more rules to manage) Risk ↓↓ (lateral movement blocked)	Partially reversible but NSG consolidation requires downtime and re-testing
ADR-004	CMK with HSM for all HIGH data	Microsoft-managed keys (default)	Cost ↑ (HSM £3K/mo) Complexity ↑ (rotation) Risk ↓↓ (key custody retained)	Irreversible for encrypted data: migration to new key = full re-encrypt

15. Board Governance Framework Infographic

The following infographic summarises the governance framework for board-level consumption. It maps the Evidence Chain Model™ from regulatory obligation through to board assurance, highlighting the key decision points and escalation triggers that enable proactive risk management.

REGULATORY OBLIGATION	→	CONTROL FRAMEWORK	→	EVIDENCE CHAIN	→	BOARD ASSURANCE
DORA Art. 5 NIS2 Art. 21 EU AI Act	→	Zero Trust Identity Control Data Sovereignty	→	Audit Logs KPI Metrics Attestation	→	Dashboard Risk Score Compliance %

16. Strategic Keywords & Competency Alignment

Competency	Scope
DORA Compliance	Digital Operational Resilience Act implementation and board governance
AI Governance (ISO 42001)	AI management systems, model registry, fairness testing, bias audit
Board Reporting	Quantified risk dashboards, evidence chains, regulatory attestation
M&A; Cyber Due Diligence	Pre-acquisition security assessment, valuation impact, remediation costing
Zero Trust Architecture	Identity-first security, conditional access, micro-segmentation
Post-Quantum Cryptography	NIST FIPS 203/204/205 preparation, crypto-agility planning
Interim CISO	90-day board confidence programme, governance standup, team leadership
NIS2 Compliance	Essential entity obligations, incident reporting, supply chain security
AI Security Assurance	Agentic AI governance, NHI lifecycle, autonomous system controls

About the Author

Kieran Upadrasta
CISSP, CISM, CRISC, CCSP | MBA | BEng

Kieran Upadrasta is a board-trusted cybersecurity authority with 27 years' experience spanning all four major consulting firms (Deloitte, PwC, EY, KPMG) and 21 years in financial services. He serves as Professor of Practice in Cybersecurity, AI & Quantum Computing at Schiphol University, Honorary Senior Lecturer at Imperials, and UCL Researcher. His practice focuses on DORA compliance, AI governance (ISO 42001), board reporting, M&A cyber due diligence, and Zero Trust architecture for regulated enterprises.

Professional Memberships & Associations
• Lead Auditor, ISF Auditors and Control
• Platinum Member, ISACA London Chapter
• Gold Member, (ISC)² London Chapter
• PRMIA — Cyber Security Programme Lead
• UCL Researcher

www.kie.ie | info@kieranupadrasta.com

References

[1] Microsoft. (2026). Microsoft Cybersecurity Reference Architecture (MCRA). Microsoft Security Documentation.

[2] NIST. (2024). Cybersecurity Framework 2.0. National Institute of Standards and Technology.

[3] European Commission. (2022). Digital Operational Resilience Act (DORA). EU Regulation 2022/2554.

[4] European Parliament. (2022). NIS2 Directive. Directive (EU) 2022/2555.

[5] ISO/IEC. (2023). ISO/IEC 42001:2023 — AI Management Systems. International Organization for Standardization.

[6] European Commission. (2024). EU AI Act. Regulation (EU) 2024/1689.

[7] Cloud Security Alliance. (2024). Zero Trust Architecture Implementation Guide. CSA.

[8] MITRE Corporation. (2026). ATT&CK; Framework v14. MITRE.

[9] IBM Security. (2025). Cost of a Data Breach Report 2025. IBM.

[10] Ponemon Institute. (2025). State of Cybersecurity in Financial Services. Ponemon.

[11] Forrester Research. (2025). The Forrester Wave: Zero Trust Platform Solutions. Forrester.

[12] Gartner. (2025). Market Guide for Cloud-Native Application Protection Platforms. Gartner.

[13] McKinsey & Company. (2025). Cyber Risk and Resilience: The Board Imperative. McKinsey.

[14] Saudi NCA. (2024). Controls Assessment Framework (CAF) v3.0. National Cybersecurity Authority.

[15] SAMA. (2024). Cybersecurity Requirements for Banking Sector. Saudi Monetary Authority.

[16] PCI SSC. (2024). PCI DSS v4.0. Payment Card Industry Security Standards Council.

Disclaimer

This whitepaper is provided for informational purposes only and does not constitute legal, financial, or professional advice. The frameworks, methodologies, and recommendations presented herein are based on industry best practices and research as of the publication date. Cybersecurity is a rapidly evolving field. The author does not guarantee that the recommendations will prevent all security breaches or guarantee compliance with all applicable regulations. Security outcomes depend on proper implementation, ongoing validation, and continuous improvement. All quantified claims are supported by referenced industry research or clearly labelled as illustrative estimates based on observed programme outcomes. Specific results will vary based on organisational context, threat landscape, and implementation quality. Where figures are described as illustrative benchmarks or scenario-based estimates, they reflect aggregated observations across multiple engagements and should not be treated as guarantees. Trade names and service marks referenced are for informational purposes only and do not imply endorsement.