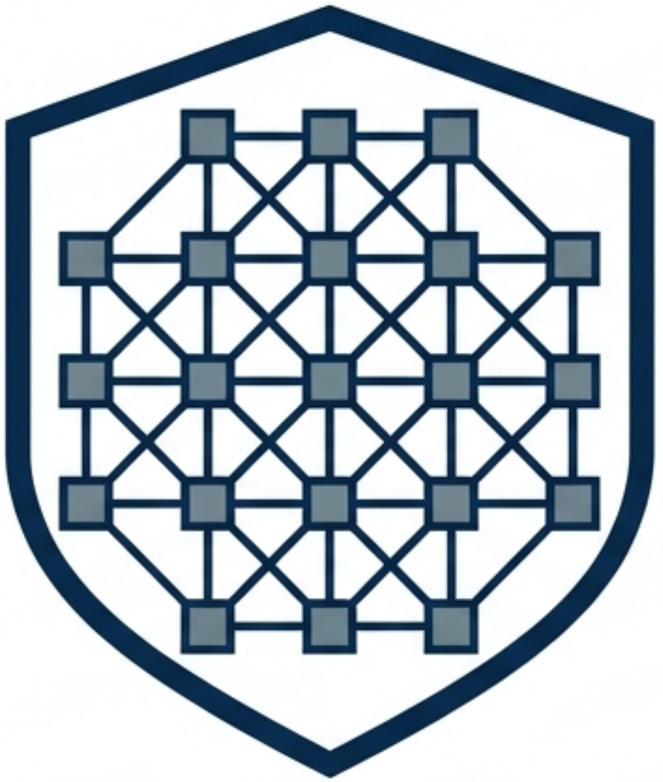
Baseline Cyber Security for AI Models and Systems

A Strategic Implementation Guide to ETSI EN 304 223 V2.1.1 (2025-12)

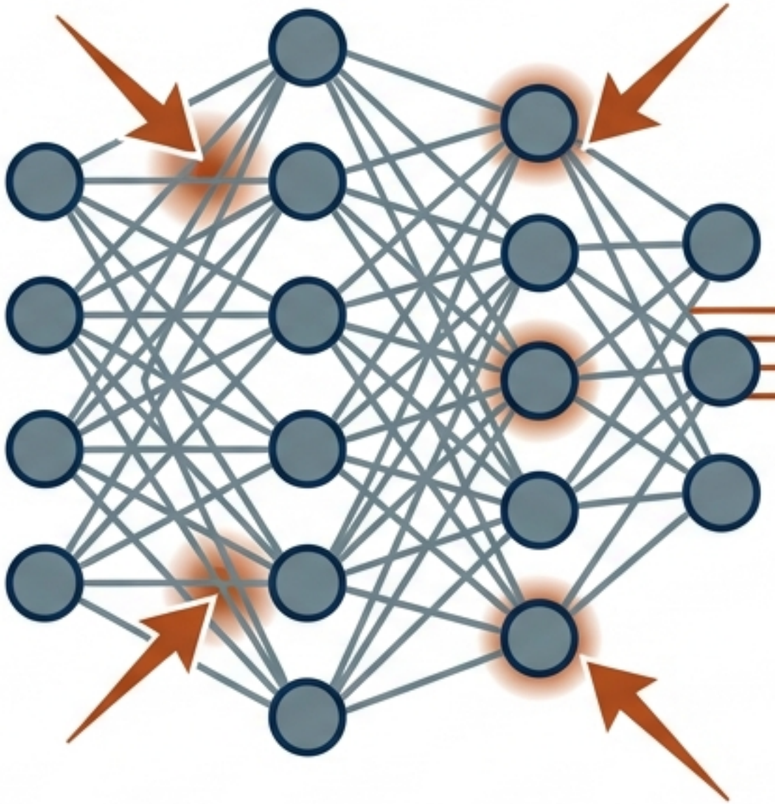
European Standard | December 2025

The Unique Attack Surface of Artificial Intelligence

Traditional Security



AI Security



Data Poisoning:
Malicious introduction of data to compromise performance (Clause 3.1)

Model Inversion:
Inferring sensitive training data from model outputs

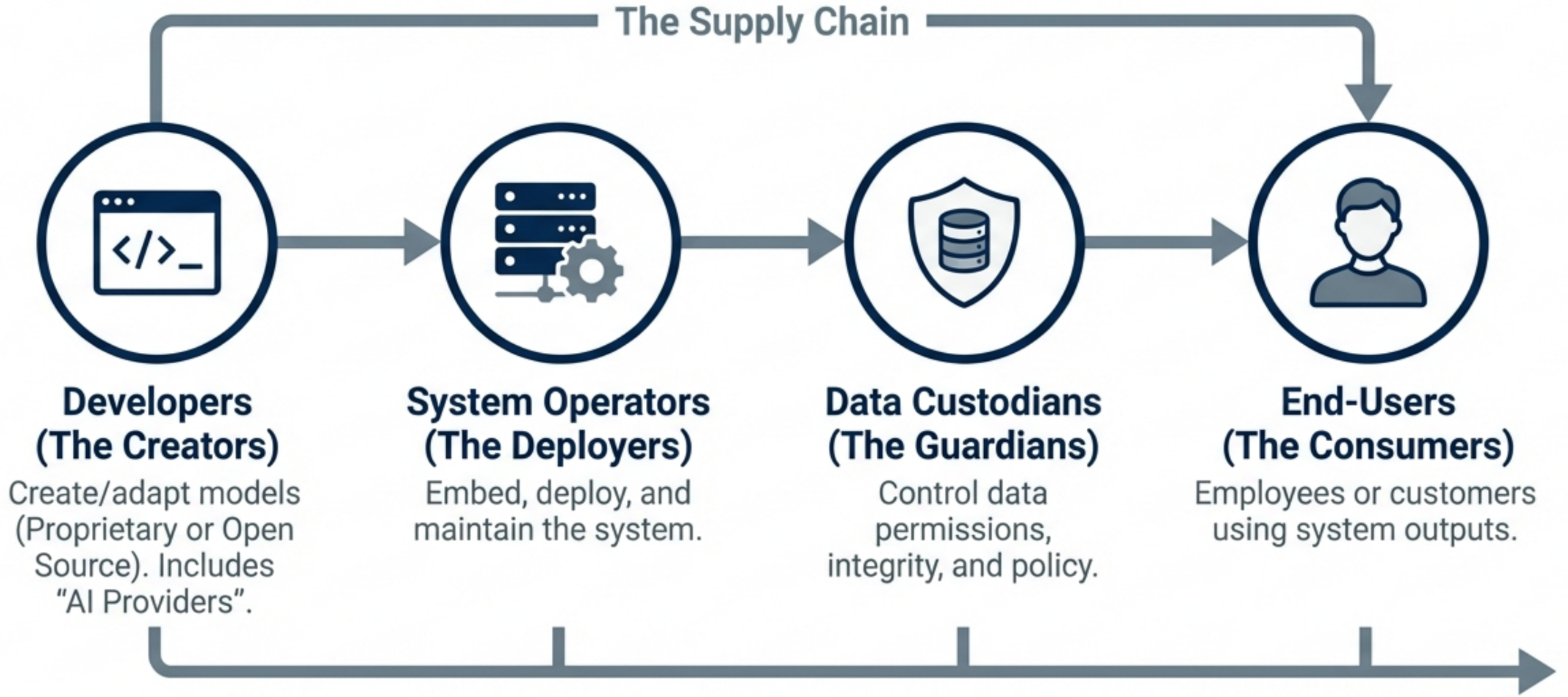
Indirect Prompt Injection:
Manipulation via instructions embedded in input data

Model Obfuscation:
Hiding malicious logic within the 'black box'

Scope: Applies to DNNs, GenAI, and engineered systems. Excludes academic research not for deployment.

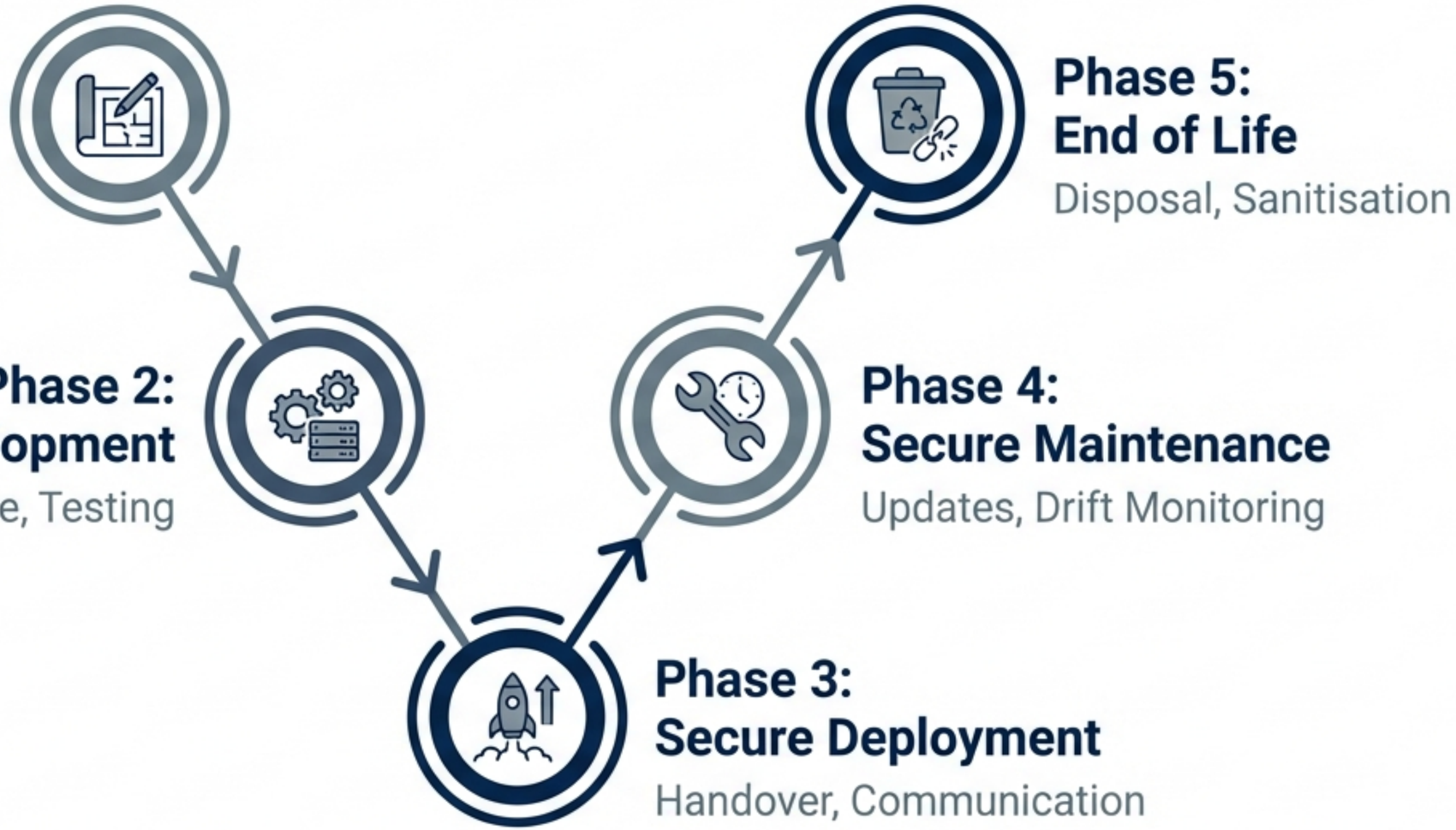
Strategic Goal: Establish a baseline for secure AI across the entire supply chain, from the AI Factory to the End-User.

Shared Responsibility Across the AI Supply Chain



Note: A single entity can hold multiple stakeholder roles.

The 5-Phase Security Framework



Aligned with ISO/IEC 22989 Life Cycle Stages.

Phase 1: Secure Design

Foundational Principles

1

Principle 1: Raise Awareness

- **Mandate:** Training tailored to roles (e.g., secure coding for devs).
- **Requirement:** Regularly update for new AI threats.

2

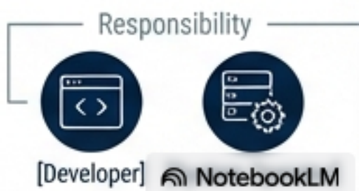
Principle 2: Design for Security

- **Mandate:** Define business requirements & risks *before* creation.
- **Action:** Design for resilience against adversarial attacks.
- **Role:** **Data Custodians** must assess data sensitivity.

3

Principle 3: Evaluate Threats

- **Mandate:** Conduct AI-specific Threat Modelling (e.g., membership inference).
- **Risk:** Address 'Superfluous Functionality' (unused modalities = attack surface).

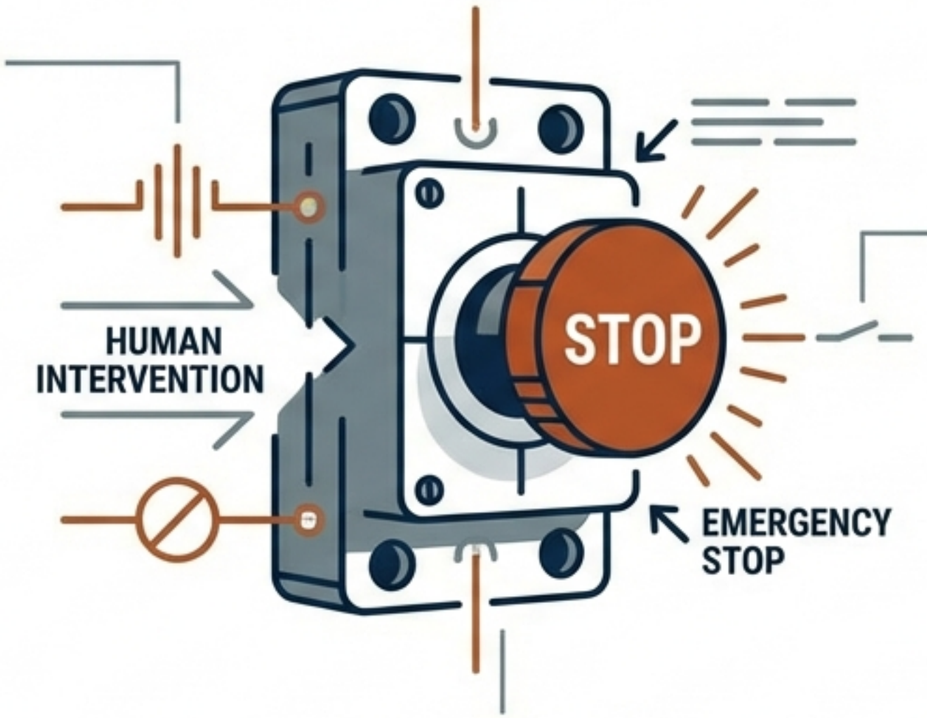


Phase 1: Secure Design (Human Responsibility)

Principle 4: Enable Human Responsibility

Human in the Loop

Systems must be designed to allow human intervention and oversight.

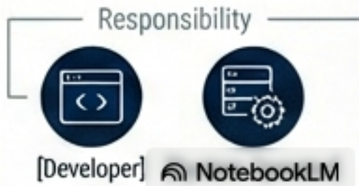


Interpretability

Outputs must be explainable. If a human cannot understand *why*, they cannot be responsible.

Prohibited Use

Developers must explicitly define and communicate prohibited use cases to End-Users.

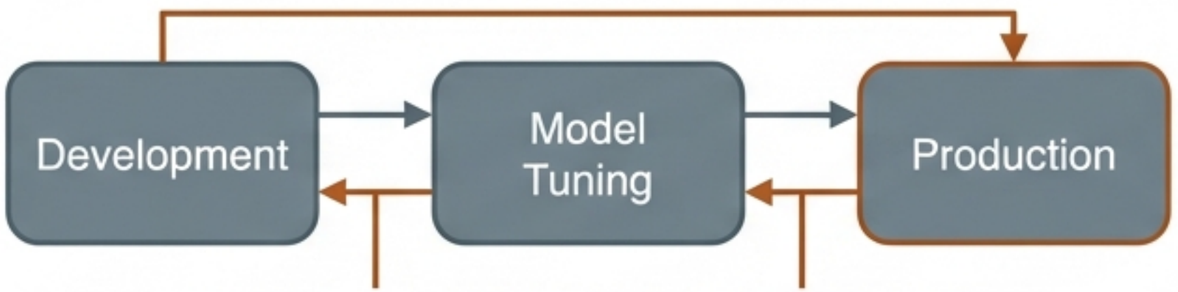


Phase 2: Secure Development (Assets & Infrastructure)

Principle 5: Identify, Track, and Protect Assets

- **Mandate:** Maintain a comprehensive inventory including data dependencies.
- **Mandate:** Maintain a comprehensive inventory including data dependencies.
- **Disaster Recovery:** Plans must account for restoring a ‘known good state’ after model poisoning.

Principle 6: Secure the Infrastructure



Strict separation of environments. Training data resides only in training environments.

- **API Security:** Implement rate limits to prevent reverse-engineering or rapid poisoning.
- **Vulnerability Disclosure:** A clear policy must be published.

Responsibility
[Developer] [Sys

Phase 2: Secure Development (The Supply Chain)

Principle 7: Secure the Supply Chain

The Weakest Link Mandates

Process: Follow secure software supply chain processes (generate SBOMs).

Verification: Re-run evaluations on released models before use.

Undocumented Components

If using a model that is not well-documented, the System Operator *must* justify this decision in writing and implement mitigating controls.

Transparency

Communicate intention to update models to End-Users *before* the update occurs.

Responsibility [Developer]

Responsibility [System Operator]

Technical Editorial

Phase 2: Secure Development (Validation)

Principle 8: Documentation

- **Data Lineage:** Document sources of training data. For scraped data, record URL and capture date/time.
- **Why?:** To trace Data Poisoning attacks discovered post-training.
- **Hashing:** Release cryptographic hashes for model components.

Principle 9: Testing

- **Red Teaming:** Use independent security testers with specific AI skills.
- **Anti-Reverse Engineering:** Ensure outputs don't leak training data.
- **Influence Check:** Verify outputs don't give users unintended influence over system behavior.

Responsibility: [Developer] [System Operator]

Phase 3: Secure Deployment

Principle 10: Communication with End-Users

Transparency Mandates

- **Data Usage**: Operators must state if user data is used for model retraining or reviewed by humans.
- **Limitations**: Provide accessible guidance on failure modes and what the system *cannot* do.
- **Updates**: Proactively inform users of security-relevant updates.
- **Incident Response**: Support agreements must be in place to help End-Users contain impacts.

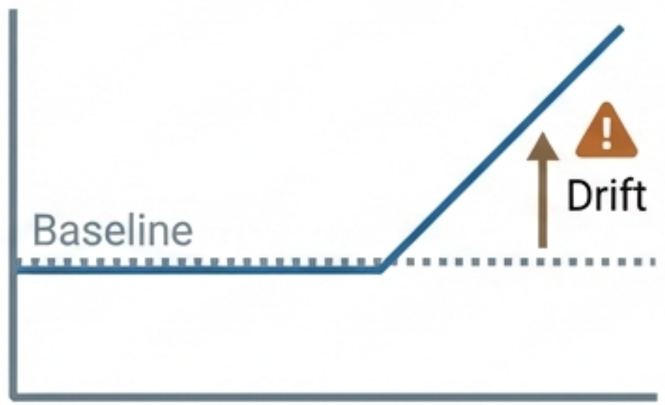
Responsibility: [Developer] [System Operator]

Technical Editorial

Phase 4: Secure Maintenance

Principle 11: Regular Updates & Patches

- **Versioning:** Major system updates = New Version. Requires a fresh security testing cycle.
- **Contingency**: Have a plan for when updates are not possible.



Principle 12: Monitor System Behaviour

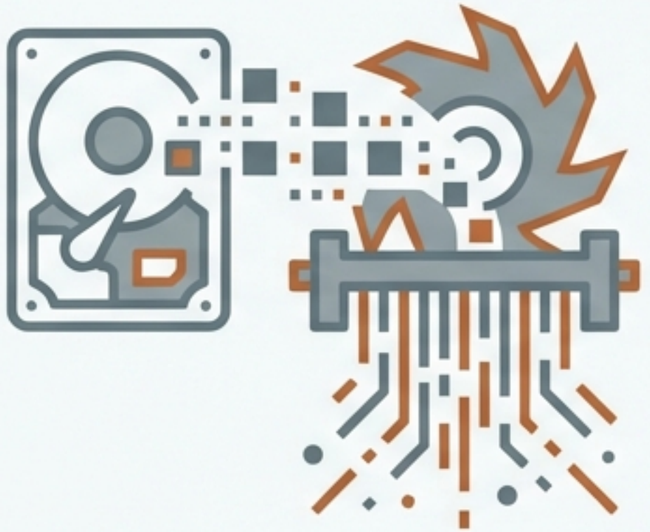
- **Beyond Performance:** Don’t just check if it works; check for anomalies.
- **Drift Detection:** Monitor for gradual changes indicating data drift or slow-roll poisoning.
- **Logging:** Log user actions to support forensic investigation.

Responsibility: [Developer] [Syst

Technical Editorial

Phase 5: Secure End of Life

Principle 13: Data and Model Disposal



The Clean Exit Mandates

- **Transfer**: If transferring a model, Data Custodians must ensure permissions travel with the asset.
- **Decommissioning**: Securely delete data and configuration details.
- **Sanitisation**: Scrub hardware and storage to prevent ‘Model Extraction’ from discarded physical assets.

Responsibility: [Developer] [System Operator]

Technical Editorial

The 13 Principles at a Glance

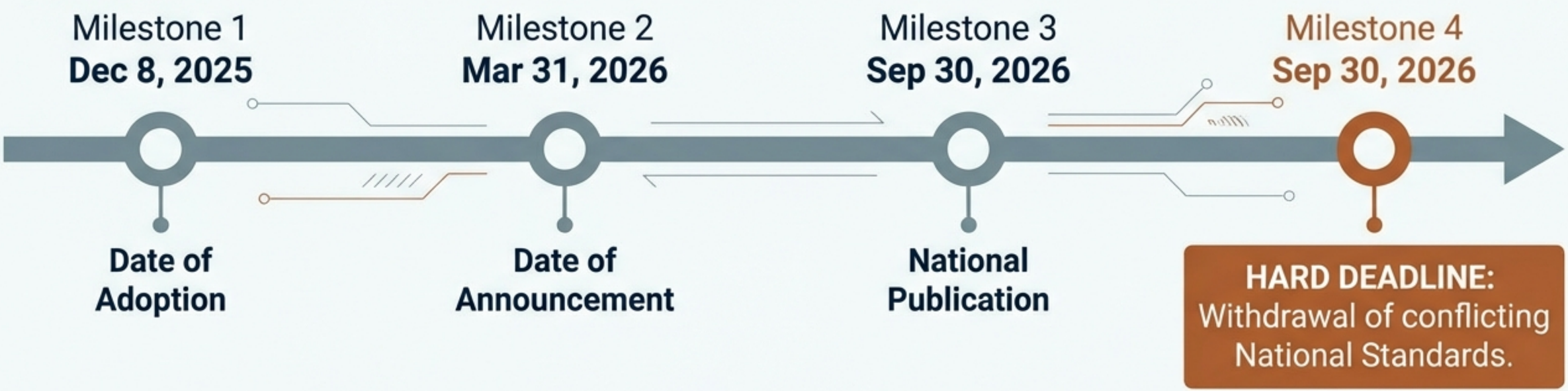
Design	Development	Deployment	Maintenance	End of Life
1. Raise Awareness	5. Identify & Track Assets	10. End-User Communication	11. Updates & Patches	13. Secure Disposal
2. Design for Security	6. Secure Infrastructure		12. Monitor Behaviour	
3. Evaluate Threats	7. Secure Supply Chain			
4. Human Responsibility	8. Document Data/Models			
	9. Conduct Testing			

Responsibility: [Developer] [System Operator]

Technical Editorial

Implementation Timeline & Compliance

Timeline for harmonizing AI security protocols and standards.

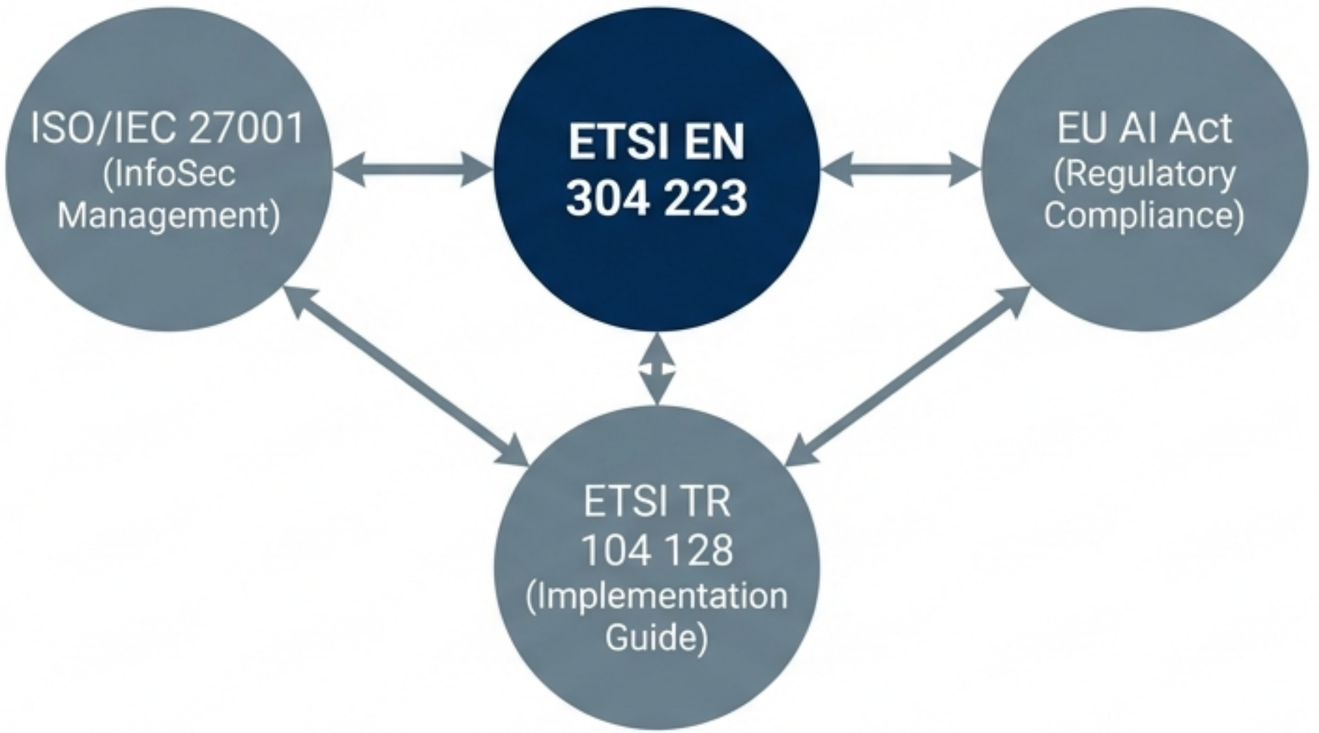


Organizations must align their AI security protocols before the **September 2026** harmonization.

Responsibility: [Developer] [System Operator]

Technical Editorial

The Broader Security Ecosystem



"Security is not a product, but a process. From design to disposal, secure AI is a shared responsibility."

Reference: ETSI EN 304 223 V2.1.1 (2025-12)