Safety OS™: Runtime Governance Infrastructure for the EU AI Act

Purpose: This document maps the Safety OS™ architecture directly to the enforcement requirements of the EU AI Act. It is designed for regulatory review, demonstrating how high-risk AI systems in healthcare can achieve demonstrable, deterministic compliance at runtime.

---

1. The Regulatory Gap: Why Logging is Insufficient

The EU AI Act mandates strict human oversight (Article 14) and robust logging capabilities (Article 12) for high-risk AI systems. However, the current paradigm of deploying AI in healthcare relies heavily on *post-execution logging*.

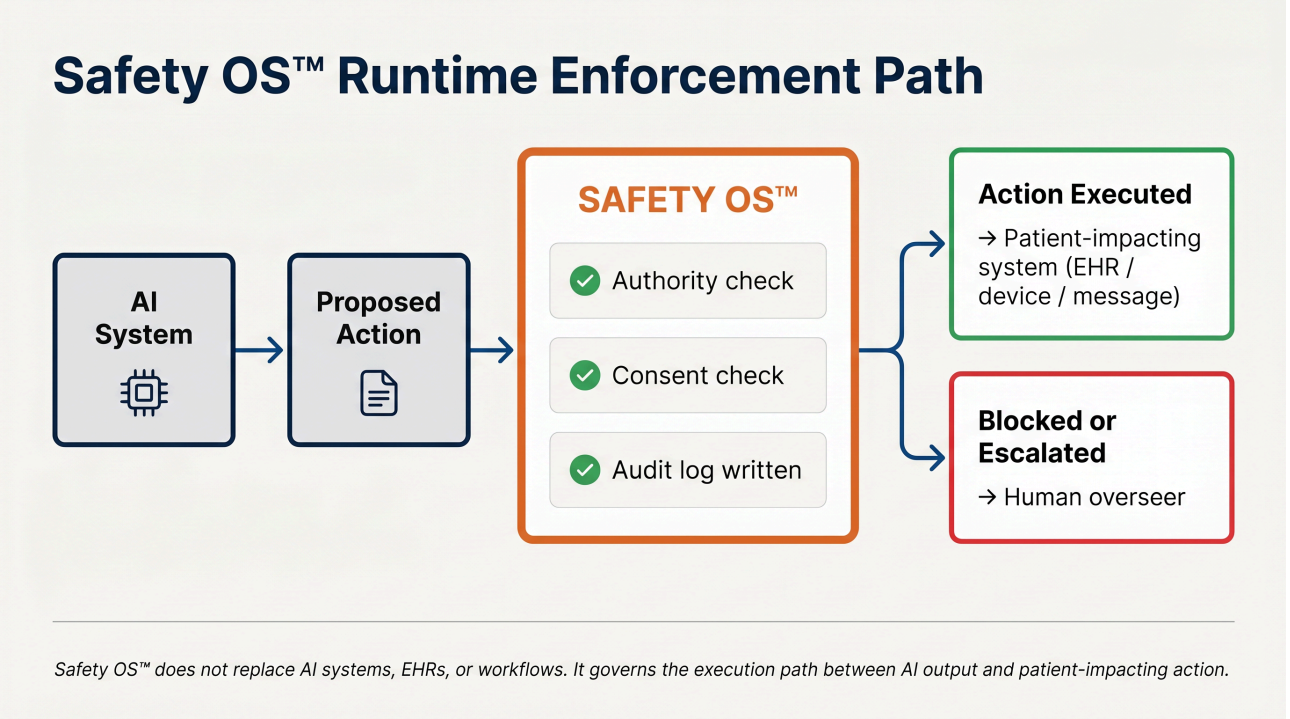
Logging records what happened, but it does not prevent a non-compliant or unsafe action from occurring. It is an auditing mechanism, not an enforcement mechanism.

To satisfy the intent of the EU AI Act - specifically the requirement that human oversight can intervene and stop the system - healthcare AI requires an orchestration layer that enforces governance *before* execution.

2. Safety OS™: The Control Plane for Runtime Governance

Safety OS™ is a Runtime Governance Infrastructure (RGI) layer. It operates as middleware - akin to Kubernetes for AI governance - sitting between the AI system's output and the patient-impacting action (e.g., an EHR write, a patient communication, or a clinical decision support alert).

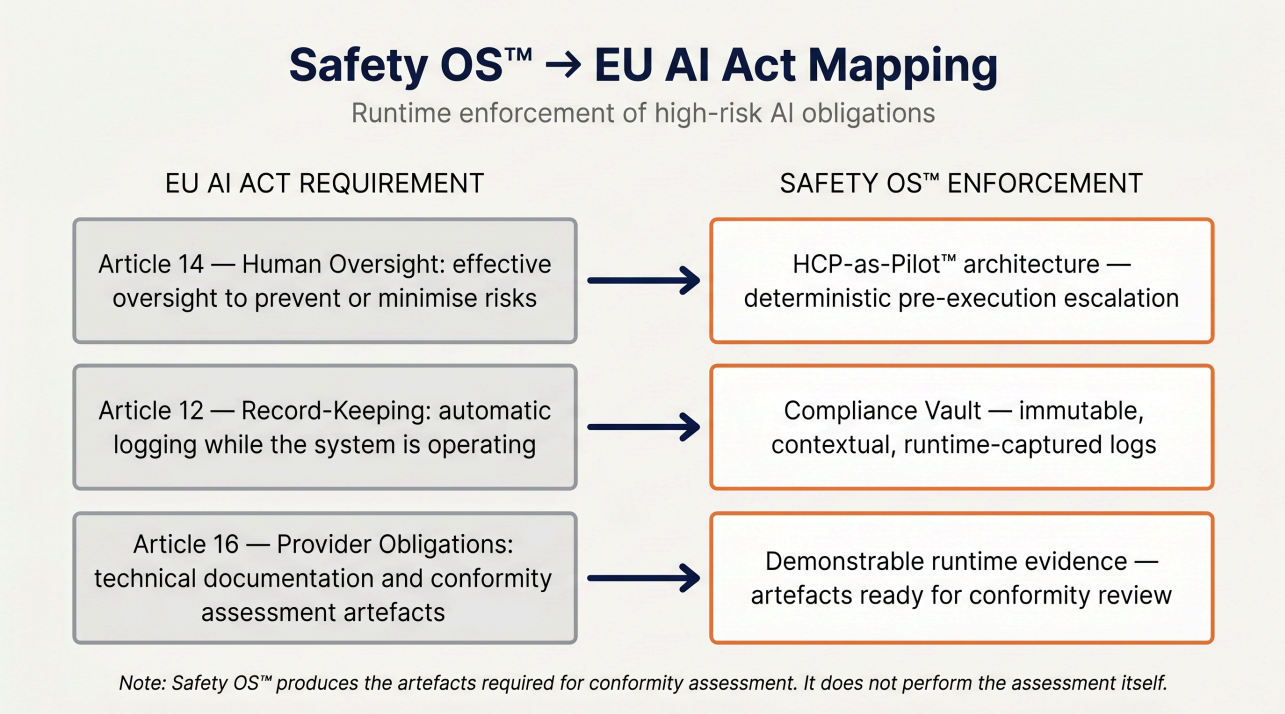
Safety OS™ does not replace the AI system or the EHR. It governs the execution path between them.



2.1 Deterministic Enforcement

Safety OS™ replaces probabilistic AI behaviour with deterministic rules. Before any AI-generated action is executed, Safety OS™ evaluates the request against a strict, policy-configurable rules engine. If the action violates clinical authority boundaries, patient consent, or predefined safety parameters, Safety OS™ blocks the execution and escalates to a human overseer.

3. Mapping Safety OS™ to the EU AI Act



3.1 Article 14: Human Oversight

Requirement: High-risk AI systems must be designed to allow effective human oversight to prevent or minimize risks to health, safety, or fundamental rights.

Safety OS™ Enforcement:

- **HCP-as-Pilot™ Architecture:** Safety OS™ enforces the HCP-as-Pilot™ model, ensuring the AI system operates strictly as a subordinate agent to the human clinician.
- **Pre-Execution Escalation:** Actions that exceed the AI's bounded autonomy are deterministically blocked and escalated to the human overseer for review and approval. The human remains the ultimate authority.

3.2 Article 12: Record-Keeping (Logging)

Requirement: High-risk AI systems must automatically record events ('logs') while the system is operating to ensure traceability and facilitate post-market monitoring.

Safety OS™ Enforcement:

- **The Compliance Vault (Truth Layer):** Every evaluation, decision, escalation, and execution passing through Safety OS™ is immutably recorded in the Compliance Vault.
- **Contextual Traceability:** Logs are not merely system errors; they capture the full context of the AI's request, the specific rule that evaluated it, the human's decision (if escalated), and the final outcome. This provides a complete, defensible audit trail.

3.3 Article 16: Obligations of Providers of High-Risk AI Systems

Requirement: Providers must draw up technical documentation, keep logs, and ensure the system undergoes the relevant conformity assessment procedure.

Safety OS™ Support:

- **Demonstrable Runtime Evidence:** Safety OS™ produces the exact technical documentation and runtime logging required to support a conformity assessment.
- *Note: Safety OS™ produces the artefacts required for conformity assessment. It does not perform the assessment itself.*

4. Conclusion

Safety OS™ shifts healthcare AI compliance from a design-time promise to a runtime guarantee. By enforcing human authority and deterministic rules at the point of action, it provides the essential Runtime Governance Infrastructure required to safely deploy high-risk AI systems under the EU AI Act.

Design-time governance defines accountability. Safety OS™ proves it at runtime.