EMILIA

EU AI ACT COMPLIANCE MAPPING · FINANCIAL SERVICES

Every high-risk action gets an owner. Here is the article-by-article proof.

Regulation (EU) 2024/1689 Annex III high-risk obligations — provisionally deferred from August 2, 2026 to **December 2, 2027** by the Digital Omnibus agreement (May 7, 2026; formal adoption pending) — carry penalties up to EUR 35M or 7% of global turnover. The deferral moved the clock, not the requirements: Article 14 still requires *technical measures* enabling human oversight — a design requirement, not a policy binder. This mapping shows how the EMILIA Protocol's receipt architecture satisfies each relevant article with evidence your assessor can re-verify independently.

`Annex III(5)(b) — creditworthiness & essential private services; fraud-exposed payment operations`

An AI agent that can release a wire, change a beneficiary, or amend payment instructions is a high-risk system the moment its output affects access to funds. The canonical attack — "the vendor updated their bank details" — succeeds precisely because approval is not bound to the beneficiary.

Article-by-article mapping

ARTICLE	REQUIREMENT	EMILIA PROTOCOL CAPABILITY	EVIDENCE ARTIFACT
Art. 9	Risk management system across the lifecycle	Policy engine evaluates every governed action to a ternary decision: allow, require_signoff, or deny — risk controls are executable, not documentary.	Policy definitions; decision recorded in each receipt
Art. 12	Automatic record-keeping enabling traceability	Every governed action emits a signed, Merkle-anchored Trust Receipt at execution time.	Receipt JSON — verifiable offline, years later
Art. 13	Transparency sufficient to interpret the system's output	The action's full context (what, against which target, initiated by which agent, approved by whom) is canonically serialized into the signed evidence.	Context fields inside the receipt
Art. 14	Human oversight: technical measures enabling intervention and override	Pre-execution hold + device-bound signoff by a named human (WebAuthn, user-verified) bound to the exact action parameters. Not a policy document — an enforced gate.	Class-A signoff assertion (ECDSA P-256)
Art. 15	Accuracy, robustness, cybersecurity	Reject-before-mutation invariant; Ed25519 / P-256 signatures; protocol core formally verified — 26 TLA+ theorems and 35 Alloy facts checked in CI.	Public conformance vectors; proofs in the open repo
Art. 72	Post-market monitoring	Receipt stream ingests into Splunk / Datadog / Elastic; a governed action without a matching receipt is itself the alarm.	SIEM field mapping (emiliaprotocol.ai/auditors)
Annex VII	Conformity assessment: demonstrable evidence	Open specification + open-source verifier + cross-language conformance suite: an assessor re-verifies the evidence with no access to our systems or the deployer's.	emiliaprotocol.ai/verify; npm @emilia-protocol/verify

Governed actions in this sector

PROTECTED ACTION	WHAT IT COVERS	CONTROL APPLIED
`payment_release`	Wire / ACH execution above policy threshold	Dual authorization; initiator ≠ approver (separation of duties)
`beneficiary_creation`	New or changed payee details	Class-A signoff bound to account + routing — a swapped beneficiary invalidates the approval
`payment_instruction_change`	Standing instruction amendments	Time-boxed approval; single use; full receipt

FinGuard: FinGuard ships policy templates for dual authorization, amount thresholds, beneficiary verification, and separation of duties — mapped to SOX-ready evidence.

How the control works (three steps)

1
Gate
The agent's call to a protected action is held pre-execution. Policy returns allow, require_signoff, or deny.

2
Signoff
A named human approves the exact action on their own device (Face ID / Touch ID / passkey). The approval is bound to the action's parameters — change one field and it is invalid.

3
Receipt
Execution releases only against the consumed signoff. A signed, Merkle-anchored Trust Receipt is the permanent, offline-verifiable evidence.

Why this evidence is different

Audit logs *assert*; receipts *prove*. Your assessor verifies every receipt with an open-source verifier — in the browser or fully offline — with no access to our systems or yours required. Neither a compromised agent, nor the deploying operator, nor EMILIA itself can forge the approval after the fact.

Verify every claim in this document yourself

→ In-browser verifier (nothing uploads): `emiliaprotocol.ai/verify`

→ Offline CLI: `npx @emilia-protocol/verify` (Apache-2.0, on npm)

→ Spec, formal proofs (26 TLA+ theorems, 35 Alloy facts in CI), conformance vectors: `github.com/emiliaprotocol/emilia-protocol`

→ IETF specification: `draft-schrock-ep-authorization-receipts`

→ Experience the ceremony first-hand: `emiliaprotocol.ai/try` · Auditor guide: `emiliaprotocol.ai/auditors`