SOLVEWORX

T-Flux Ultra Executive Security Brief

Enterprise AI security for boards, CISOs and risk leaders

Executive position

Large Language Models create a different security problem from conventional enterprise software. They interpret natural language, retrieve enterprise information, invoke tools and can increasingly initiate actions. T-Flux Ultra is designed on the assumption that prompts, retrieved content, model outputs and agent actions should not be trusted by default. Security is enforced by the platform around the model, not by relying on the model to behave correctly.

Private infrastructure + governed data + controlled models + policy guardrails + least-privilege agents + compiled execution + auditability

Why this matters to boards and CISOs

- Prompt injection can manipulate model behaviour, including through malicious instructions hidden inside external content.
- LLM output can be dangerous if it is passed directly into applications, databases or APIs without validation.
- Sensitive data can be disclosed when model access, retrieval or egress is insufficiently controlled.
- Agents can create material operational risk if they are given excessive authority to act.
- Third-party models, libraries and AI components introduce new supply-chain and provenance risks.
- AI-generated answers can be inaccurate or inappropriate, creating legal, compliance and decision risk if over-relied upon.

T-Flux Ultra control model

Infrastructure	On-premises, private cloud or customer-controlled environments keep enterprise AI workloads inside the required security and jurisdictional perimeter.
Identity & access	RBAC, ABAC, ACL-aware retrieval, tenant isolation and scoped permissions control who and what may access data, models, tools and actions.
Data governance	Data can be classified, version controlled, approved, superseded, retired, masked, retained and restricted at granular level.
Model governance	The Model Catalogue controls model provenance, version, approval state, deployment location, tenant assignment and lifecycle.
Guardrails	Policy controls apply before, during and after model execution to restrict data access, output, tool invocation and downstream actions.
Agents & tools	Least privilege constrains each agent to explicitly authorised capabilities, with human approval where required.
Audit & oversight	Requests, retrieved evidence, model versions, guardrail decisions, tool calls, outputs and downstream actions can be traced and audited.

SOLVEWORX

Key enterprise AI risks and T-Flux response

Prompt injection	Contain impact through permission boundaries, ACL-aware retrieval, model/tool restrictions and approval gates. A manipulated model does not inherit additional authority.
Insecure output	Treat model output as untrusted. Validate, sanitise and policy-check generated SQL, code, API calls and structured payloads before execution.
Sensitive disclosure	Keep data within the customer-controlled environment and apply classification, masking, egress restrictions, retrieval permissions and audit logging.
Insecure plugins/tools	Use authenticated connectors, scoped credentials, allow-listed APIs, validation and transaction-level permissions.
Excessive agency	Apply least privilege to each agent and differentiate read, create, update, approve, delete, transmit and execute authority.
Overreliance	Use grounded retrieval, citations, traceability, validation, multi-model comparison and human-in-the-loop controls for high-risk decisions.
Supply chain	Use controlled model registration, provenance, approval, versioning and lifecycle management; maintain controlled software dependencies and releases.

Why the compiled Java/C++ architecture matters

T-Flux Ultra is built around compiled Java and C++ software rather than a Python-script-driven execution model. The security proposition is not that one language is inherently secure and another insecure. The advantage is a more constrained and auditable runtime: approved compiled capabilities can be exposed to models and agents without providing a general-purpose scripting environment for arbitrary runtime execution.

- Reduces reliance on dynamically generated scripts, notebooks and runtime package installation.
- Constrains agents to predefined and authorised functions rather than unrestricted code execution.
- Supports signed builds, controlled releases, locked dependencies, vulnerability scanning and software bills of materials.
- Makes deployed capability, version and change history more deterministic and auditable.

Board-level assurance statement

T-Flux Ultra is designed so that a compromised prompt or unpredictable model response encounters independent controls over data, permissions, tools, actions and execution.

The objective is not to claim that LLM risk can be eliminated. It is to ensure AI operates inside an enterprise-controlled environment where the customer retains authority over the data, models, permissions, workflows, infrastructure and actions.