TARTU ÜLIKOOL
1632

Towards Secure AI-Driven Security Assessment

Ijeoma Faustina Ekeh
Vjatšeslav Antipenko
Raimundas Matulevicius

CHESS
Cyber-security Excellence Hub in Estonia and South Moravia

Institute of Computer Science
University of Tartu

34th European Conference on Information Systems 15.06.2026

Business scenario

“Well, usually our suppliers just send the invoices as PDFs to our general office email. Once I see them, I download the file and put it into a folder on our shared office drive so the accountant can look at it later.”

Security flags

Secure storing

Secure commu-nication

"Well, usually our suppliers just send the invoices as PDFs to our general office email. Once I see them, I download the file and put it into a folder on our shared office drive so the accountant can look at it later."

Access control

Information manipulation

Prototype development streams

Course project in applied artificial intelligence
Information Security ChatBot

Interview Logic (Perception module)

Bsc Thesis
Design and Implementation of AI-agent for Cybersecurity Risk Assessment

Risk Assessment Logic (Reasoning module)

Kirill Mitjurin https://thesis.cs.ut.ee/9537516b-f766-48a5-ab37-9943b5f24b83

Digital Innovation Industry Project (DIIP)
AI-Powered Compliance Assessment Tool

UI/UX

Agentic AI architectural deployment patterns

Deployment and Security (Infrastructure)

Research questions

RQ1: *What architectural patterns and technological building blocks are currently used to construct LLM-based assistant systems?*

RQ2: *To what extent do these architectural patterns satisfy the predefined functional and security requirements for AI-supported assessment assistants?*

Requirements

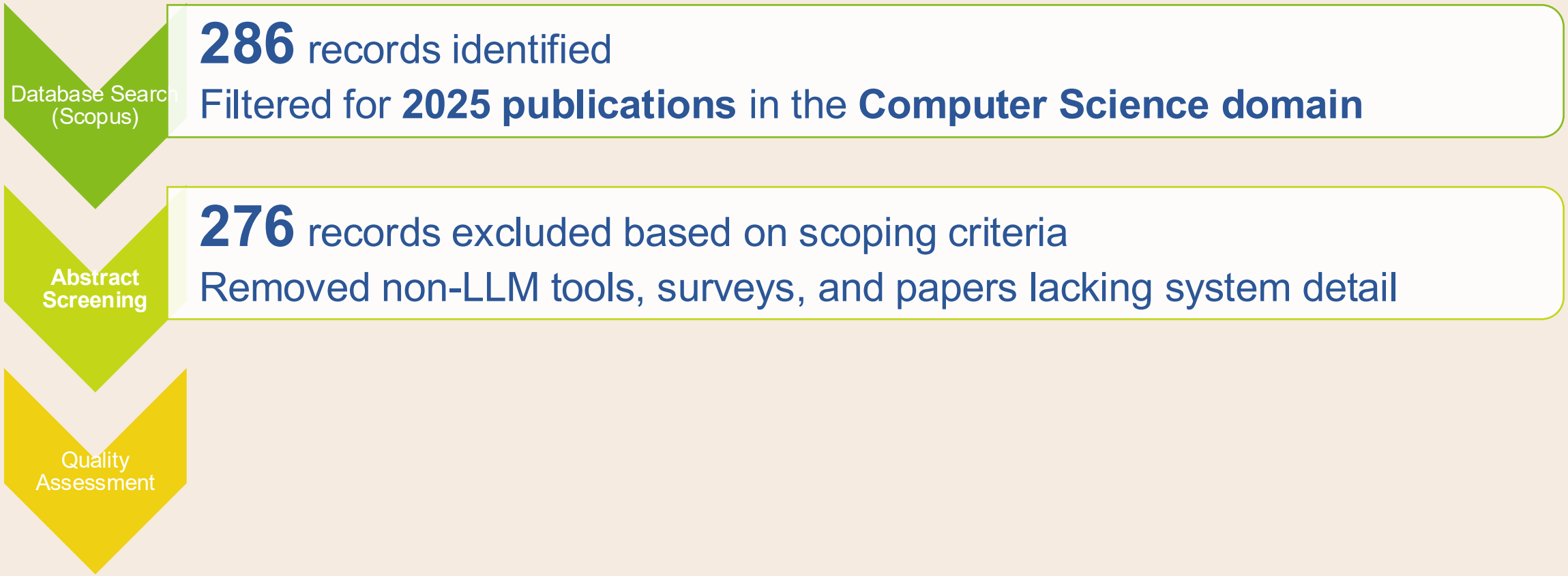
Selection from the full functional and security requirements tables

Functional requirements	
REQ1.2	Train the model
REQ1.4	Continuously tune or update the model
REQ2.3	Deploy the ML model
REQ4.2	Validate input data before or during processing

Security requirements	
SRQ1	Maintain and monitor training data and the ML model before data preprocessing
SRQ4	Encrypt and protect ML model parameters from unauthorized access
SRQ10	Perform contextual analysis on user prompts or input data to detect malicious intent.
SRQ13	Verify AI artefacts (models, configurations, dependencies) before usage.

Ekeh, I. F., Matulevičius, R., & Karagjaur, M. (2025). Systematic technical report for model-driven elicitation of functional and security requirements in AI-supported systems [Technical report]. Zenodo. https://zenodo.org/records/17878604

Research Method



Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering (EBSE Technical Report EBSE-2007-01). Keele University and Durham University Joint Report.

Functional & Security Coverage

Across 9 papers: functional patterns are visible, security controls are mostly missing

Functional requirements

Mostly partial coverage

2–3 F 6–7 P 3 N

- Modular / multi-agent designs
- Layered or microservice patterns

Security requirements

0 **papers fully address any SRQ**

2–3 P 11–12 N

- Privacy, audit & compliance missing
- Robustness & drift monitoring weak

Bottom line: functional maturity is emerging; security remains the decisive gap.

Observed Architectural Patterns

Software as a service (SaaS) LLM API (Cloud-Only Inference)
Minimal deployment effort, but requires total 3rd-party trust

Software as a service (SaaS) + Retrieval-Augmented Generation (RAG)
High context-awareness, but introduces data exposure risks

Hybrid (Local Processing + Cloud Inference)
Enhanced auditability, but high orchestration complexity logic with cloud inference

Fully Local / Private Cloud LLMs
Full data control and isolation, but heavily resource-intensive

Conclusions & Work in Progress

The study points to a control gap — not a capability gap.

Functional maturity ≠ security readiness

01	**Functional capability exists**	Modular pipelines and input handling are common.
02	**Lifecycle governance lags**	Training, testing, versioning and drift management are weak.
03	**Security must be designed in**	Access control, sanitisation, verification and robustness are missing.
04	**Next step: realisation framework**	Map architecture choices to requirements, then validate through a prototype.

Bottom line: trustworthy assistants require architecture-level security, not post-hoc wrappers.

MOVE BEYOND WRAPPERS

Toward hybrid / private-cloud designs with explicit control

verify • govern • monitor

TARTU ÜLIKOOL
1632

CHESS

Cyber-security Excellence Hub in Estonia and South Moravia