fortiss

AI Agents and Security Compliance in the Software Development Process

Florian Angermeir

fortiss, Blekinge Institute of Technology

SERL Conference – 19.11.2026



Two Perspectives

1

Agents as Analysts

How can we use agents to analyze process compliance?

2

Agents as Engineers

How does agentic development reshape our compliance evidence?

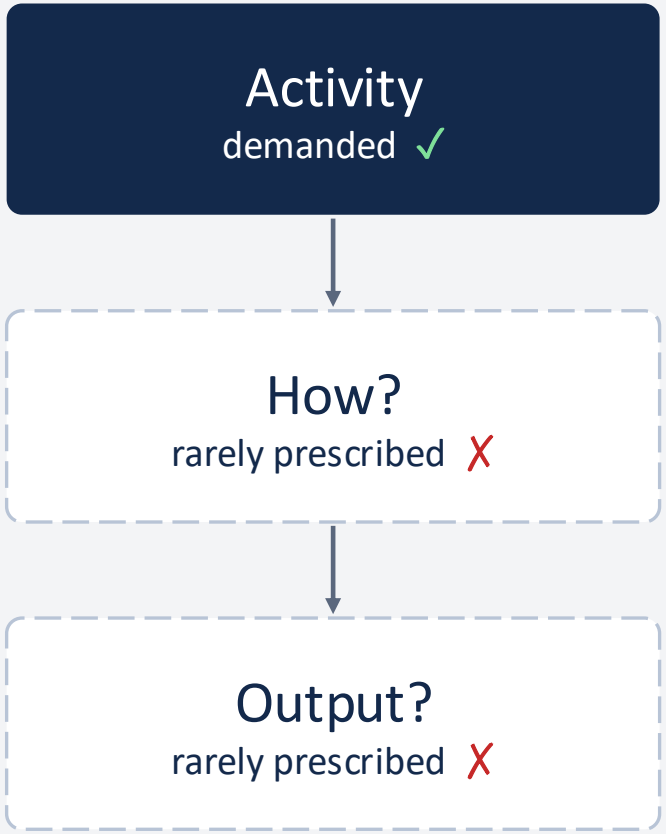
① uses agents ***for*** compliance, ② examines compliance ***under*** agents

Compliance Requirements

"

Manufacturers shall "apply **effective** and **regular** tests and reviews of the security of the product with digital elements"

Regulation (EU) 2024/2847 (CRA), Annex I, Part II



1 Hard to check

The activity must be inferred from the output

2 Open to interpretation

When engineers become agents, it is unclear what "performed" means

1

Agents as Analysts

How can we use agents to analyze process compliance?

Agents as Analysts: Motivation

Effort

Compliance activities are resource-intensive

- **Proactive** — self-assessment, audit preparation
- **Reactive** — audit findings, incidents

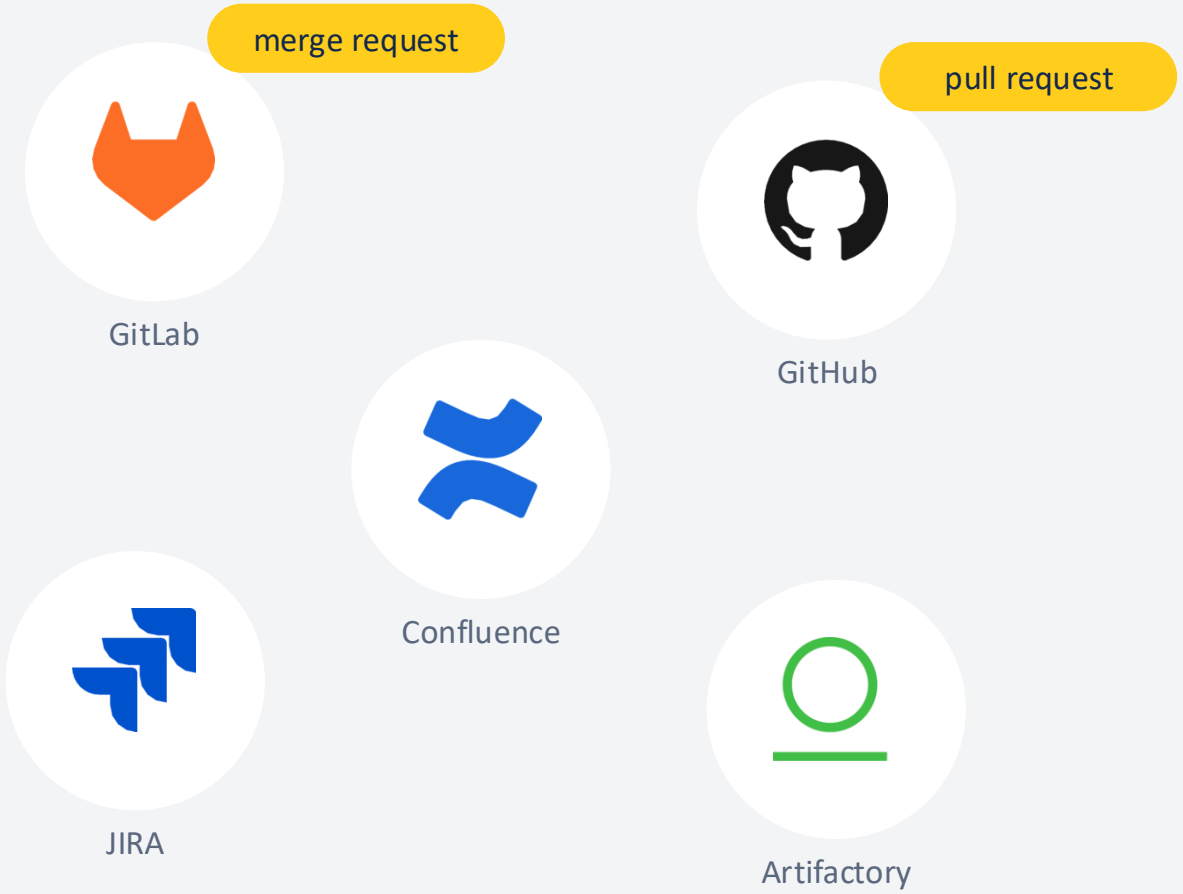
Scale

More changes, faster



Agents could help, but only if grounded in actual data: otherwise they are not reliable from run to run

Agents as Analysts: Problem



- Data is distributed across platforms
- Each platform uses its own terminology (**merge request** vs. **pull request**)
- Compliance statements require reasoning across all of them

Agents as Analysts: Approach

OTKG: Ontology-driven Temporal Knowledge Graphs

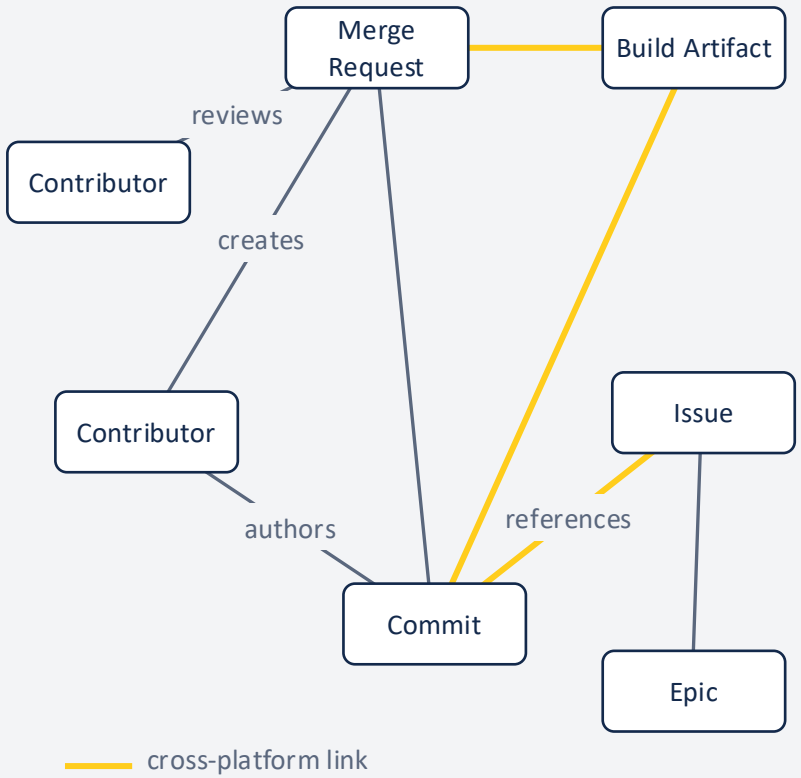
01 SOURCES

GitLab
- Merge Request
- Contributor
- Commit

JIRA
- Issue
- Epic

Artifactory
- Build Artifact
- Dependency

02 GRAPH



03 ANALYSIS

Security Posture Analysis

"Does the development process follow our requirements?"

Runs as

Query | Pipeline | Agent

→ always with **evidence**

Agents as Analysts: Example

IEC 62443-4-1

SI-1: Security Implementation Review

- Identify security requirements not adequately addressed by the implementation
- Static code analysis (SCA) on all source code changes

Heuristics

1. Change requests must be accompanied by a review
2. Reviews must discuss security requirements in substance
3. Change requests must contain an SCA report
4. Change requests must use the SCA results

Agents as Analysts: Example (Heuristics → Pipelines)

HEURISTIC

Reviews must discuss security requirements in substance

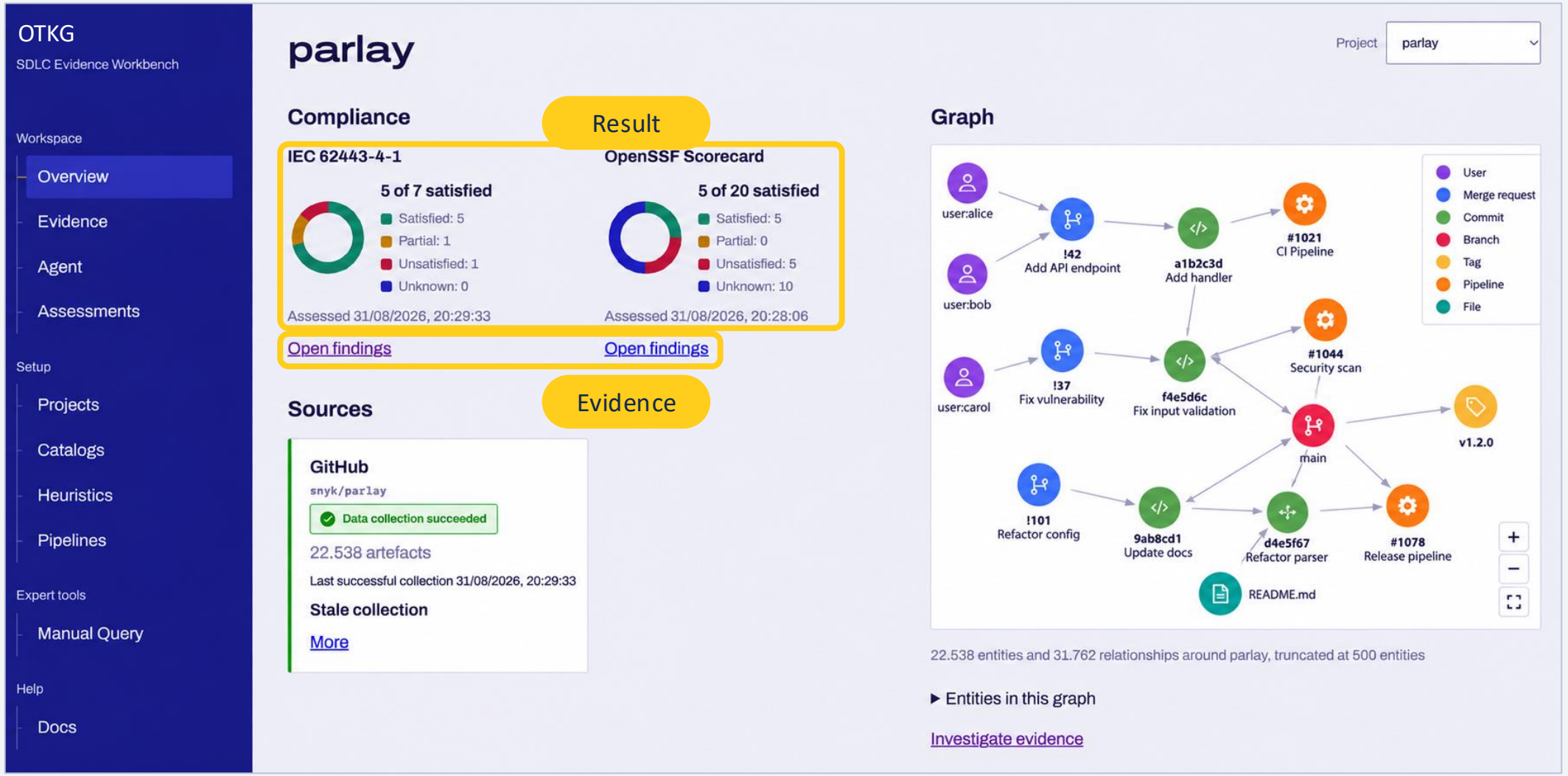
Query deterministic

LLM judgment

1. Collect change requests — Query
2. Is security-relevant code changed? — LLM
3. Does the review have security substance? — LLM
4. Gather security requirements — Query
5. Do the reviews cover the requirements? — LLM

→ pass / fail + evidence

Agents as Analysts: Example (Result)



Agents as Analysts: Status

Evaluation ongoing

OTKG platform in place
Focus

- IEC 62443-4-1 assessment
- Open Source Analysis

2

Agents as Engineers

How does agentic development reshape our compliance evidence?

Agents as Engineers: Shift

Human-driven	Agent-assisted	Agentic
1	2	3
Human acts, tools assist	Human acts, agent suggests	Agent acts, human oversees
IDE, CI, SAST	Code completion, chat	Plans, codes, tests, fixes
Our compliance evidence was designed for this		We’re heading here

Agents as Engineers: Motivation

1

Speed

Agentic development is arriving faster than standards and audit practice can adapt

2

Scale

Agent-produced changes grow while human review capacity stays constant

3

Regulation

Regulations expect evidence.

IEC 62443-4-1 SM-4 requires security expertise. Whose?

Agents as Engineers: Problem

1. Controls implicitly assume a human performs the activity

2. Evidence assumes human-produced traces (review comments, approvals, reports)

Merge request !128

Add input validation to login form

? Author — Human? — Agent?

? Reviewer — Human? — Agent?

Review comment: *“LGTM, security requirements are addressed.”*

Approved ✓

Agents as Engineers: Example (Threat Modeling)

IEC 62443-4-1 SR-2: Perform threat modeling for the product

Agent does	Produces a threat model
Evidence shows	Complete, well-structured document ✓
What's missing	Whether threat modeling was actually ***performed***, or the document was only ***generated***

Agents as Engineers: Example (Static Code Analysis)

IEC 62443-4-1 SI-1: SCA on all code changes, using a tool if one is available

Agent does	Reviews the code for security coding errors itself, without a separate SCA tool
Evidence shows	No SCA report ✗ the analysis exists only in the agent's **trace**
What's missing	Recognized evidence: the trace is not a typical audit artifact

In agentic development, the evidence probably lies in the trace, not in the outcome

Agents as Engineers: Research Directions

RQ1

Activity

When does an agent’s trace show that a demanded activity was performed, not just its output generated?

RQ2

Evidence

Can agent traces count as compliance evidence, and what does the trace need for that (completeness, integrity, readability)?

RQ3

Oversight

What human oversight of agentic development is enough, and how is it evidenced?

Agents as Engineers: Opportunities for Collaboration

1

Pilot teams

Teams already developing agentically: their workflows as case studies

2

Trace data

Access to agent traces and logs, to examine RQ1–RQ3 on real data

Contact



Florian Angermeir

angermeir@fortiss.org

