ProSec Modeler: Embedded Security Compliance with BPMN*

Leo Poss[1,*], Linda Maria Kölbel[1] and Stefan Schönig[1]

[1]University of Regensburg, 93040 Regensburg, Germany

Abstract

IoT regulatory compliance requires verifiable alignment between operational processes and security standards. Especially in resource-constrained Small and Medium-sized Enterprises (SMEs), traditional workflows rely on decoupled spreadsheets manually cross-referenced with process models. This human-centric approach introduces critical vulnerabilities, including zero element-level traceability, compliance drift during model evolution, and an inability to contrast design intent against operational reality. To bridge this gap, we present *ProSec Modeler*, an embedded, web-based tool that integrates compliance tracking directly into a BPMN canvas. This unified environment replaces siloed spreadsheet checklists by mapping standard-agnostic relational attributes directly to unique process element identifiers.

Keywords

BPMN, security compliance, IoT, model-driven assessment, ISVS

1. Introduction

Regulatory compliance in industrial organizations using (Industrial) Internet of Things (IoT) architectures demands explicit traceability between security mandates and process-level interactions. Frameworks such as the *OWASP Internet of Things Security Verification Standard* (ISVS) [1] define controls spanning ecosystem design, device security, data protection, and operational integrity that can be mapped directly to the structural components (participants, activities, and message flows) of Business Process Model and Notation (BPMN) models [1, 2, 3]. Recently, unified catalogs have emerged to operationalize these requirements for industrial environments [4]. With global regulations tightening, demonstrating traceable, auditable compliance at the design stage has transitioned from a best practice to a critical competitive and operational requirement.

In contemporary enterprise environments, particularly within Small and Medium Enterprises (SMEs), compliance evaluations are frequently treated as an administrative burden. Due to limited resources, smaller businesses rarely have dedicated compliance officer roles; instead, evaluations are conducted ad hoc, using decentralized spreadsheets that are manually cross-referenced with visual design models. This manual practice introduces three critical operational vulnerabilities that hinder an organization's ability to maintain a robust security standing: *(i)* Traditional workflows rely on siloed checklist spreadsheets decoupled from actual process diagrams. There is no explicit, programmatic link between an absolute compliance response and the exact BPMN model component (e.g., a specific Task or Lane) responsible for fulfilling it. This makes auditing nearly impossible, as human auditors cannot easily verify where or how a requirement is implemented in the system.; *(ii)* As industrial process designs undergo structural iterations, security states might degrade slowly. A security requirement verified in one version can be completely invalidated or erased by a layout change in the next version, without triggering any structural or syntax warnings., and *(iii)* Organizations struggle to contrast their idealized design-time security intent (*To-Be*) against the actual runtime operational configurations (*As-Is*). Without a unified

Proceedings of the BPM 2026 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum (BPM 2026), Toronto, Canada, September 27th to October 2nd, 2026.

*This work is funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (SIREN, Grant No. DIK-2308-0012//DIK0511).

*Corresponding author.

leo.poss@ur.de (L. Poss); linda.koelbel@ur.de (L. M. Kölbel); stefan.schoenig@ur.de (S. Schönig)

0009-0009-9221-8704 (L. Poss); 0009-0006-6907-2784 (L. M. Kölbel); 0000-0002-7666-4482 (S. Schönig)

environment, checking the delta between planning and operational reality remains a challenging, manual task.

While model-driven security literature offers BPMN extensions for annotating security properties [5, 6], these approaches rely heavily on static, almost always visual, annotations that lack a decoupled persistence layer, comparative capabilities, and machine readability and thus executability. Literature reviews confirm a widening gap between modeled security intent and evidence of operational verification [7]. To bridge this gap, we introduce *Process Security & Compliance Modeler (ProSec Modeler)*, a web-based, database-backed BPMN modeling environment designed to formalize and execute embedded security compliance assessments. Unlike static security annotations or flat spreadsheets, ProSec Modeler dynamically contextualizes compliance checklists. When an element is selected, the tool surfaces only the attributes relevant to that specific element type (e.g., task vs. lane), embedding evaluation directly into the design workflow.

Our contributions for this tool demonstration are as follows: *(i)* an evolutionary traceability architecture that binds compliance metadata directly to unique BPMN element identifiers; *(ii)* a standard-agnostic relational schema separating standards, modular attributes, and multi-session assessments; and *(iii)* a multi-posture assessment management layer that allows analysts to preserve, switch, and evaluate differing operational postures (To-Be vs. As-Is) over an identical underlying process model.

2. Tool Architecture & Relational Knowledge Model

ProSec Modeler[1] is a data-driven, single-page web application using `bpmn-js`[2] for canvas rendering, `Tanstack Router`[3] for state management, and an embedded `SQLite` database via `drizzle`[4]. Instead of hardcoding compliance frameworks, the UI sidebar dynamically generates widgets based on metadata queried from the relational layer. To decouple compliance and process lifecycles, the data model isolates the framework topology from the process topology across three core relational domains (see Figure 1):

Compliance Domain (*Norm, Requirement, Attribute*): This hierarchy ingests arbitrary frameworks as database rows. The *Attribute* entity defines UI rendering via the `input_type` property and routes structural tracking obligations to specific BPMN types (Tasks, Events, Pools, Lanes) using the `applicable_for` column.

Process Topology Domain (*Process_Model, Process_Element*): Extricated from static text annotations inside the BPMN XML, individual process elements are mapped directly to unique relational identifiers upon ingestion.

Evaluation Intersection (*Compliance_Assessment, Assessment_Value*): This domain utilizes a ternary relation to map independent operational states. Let A be the set of Compliance Assessments, E be the set of Process Elements, and V be the set of structural Attributes. The valuation mapping is defined inline as $f : A \times E \times V \rightarrow \text{Value}$. Resolving this relationship allows different process states, such as design-time (*To-Be*) and runtime operational reality (*As-Is*), to coexist on an identical process diagram topology without data duplication.

2.1. Implementation & Frontend-Backend Synchronisation

To enable real-time updates without hindering canvas operations, *ProSec Modeler* is based on a decoupled, asynchronous state-sync pattern: When users interact with graphical components on the canvas workspace, we fetch the explicit metadata map from the server. The client application tracks changes locally before writing blocks into the database instance. Input mutations executed within the UI (e.g., changing a dropdown selection or editing free-text reference paths) immediately dispatch database updates. Upon receiving confirmation from the server database, the frontend uses the internal

[1] A short, guided introduction can be found at https://youtu.be/Gvv64dwVVEg. The source code can be found at https://github.com/LeoPoss/ProSecModeler/ with a running version found at http://pcinf00057.ur.de/modeler

[2] https://bpmn.io/toolkit/bpmn-js/

[3] https://tanstack.com/router/latest

[4] https://orm.drizzle.team/

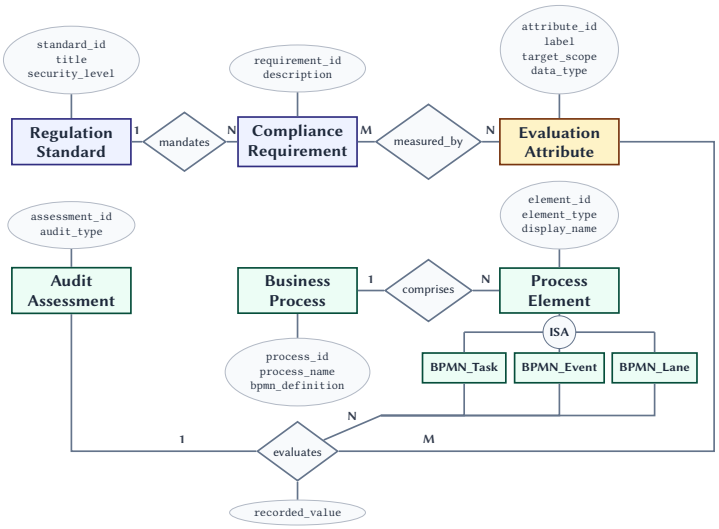

Figure 1: Relational Knowledge Model Architecture.

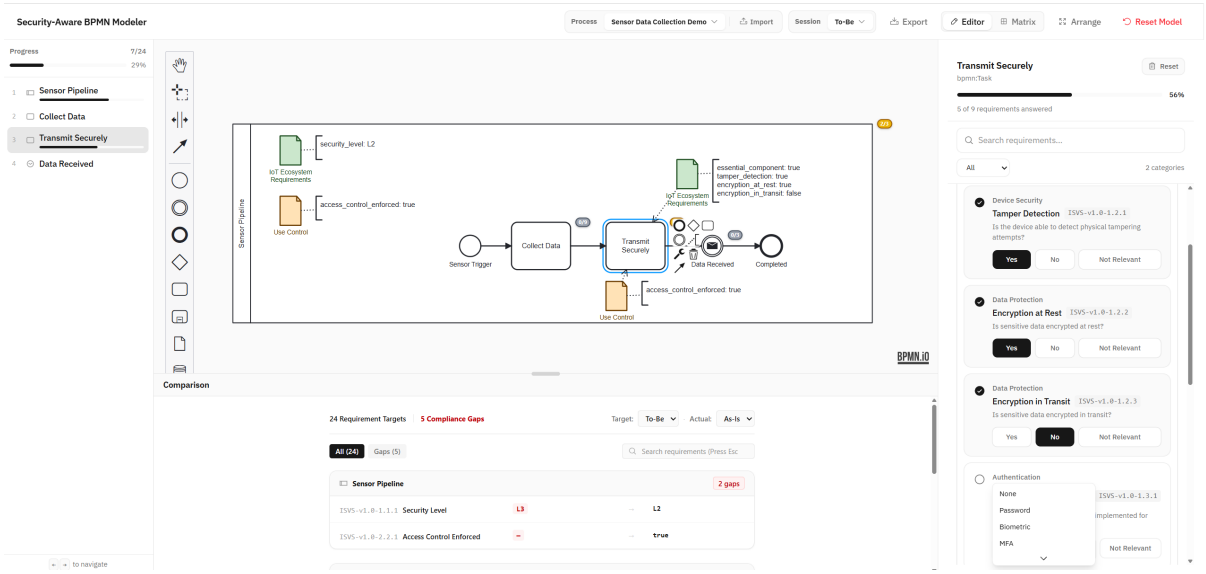

Figure 2: ProSec Modeler interface execution view: Interactive BPMN canvas highlighting embedded element compliance badges alongside context-aware sidebar attributes for the ISVS [1].

`ElementRegistry` API of `bpmn-js` to programmatically append or shift visual overlays. This dynamic layout injection ensures that form schemas can expand inside the backend tables without breaking UI compatibility.

2.2. Attribute Model and Structural Coverage

The tool operationalizes the theoretical mappings from the ISVS framework by linking specific security attribute categories across four standard process modeling blocks [1, 4]. Table 1 defines the specific structural alignment criteria embedded inside the current database configuration. For this demonstration, we focus on a small subpart of the norm with full support in an upcoming publication. Of the active attributes parsed during compliance runs, seven currently render as binary Boolean choices, two generate finite, discrete selection lists that match standard verification scopes [1], and one accepts unrestricted character arrays to log external threat-model reference strings or audit URLs.

Table 1
Supported ISVS attribute categories with element-type coverage.

Requirement Attribute	ISVS ID	Task	Msg. Event	Pool / Lane
IoT Ecosystem Requirements				
Security Level	1.1.1	○	○	●
Essential Component	1.1.2	●	●	○
Threat Model Reference	1.1.3	○	○	●
Tamper Detection	1.2.1	●	○	○
Encryption at Rest	1.2.2	●	○	○
Encryption in Transit	1.2.3	●	●	○
Authentication Mechanism	1.3.1	●	○	○
Authorization Enforced	1.3.2	●	○	○
Signed Updates	1.4.1	●	○	○
Rollback Support	1.4.2	●	○	○
Use Control				
Access Control Enforced	2.2.1	●	●	●

3. Demonstration Scenario

To demonstrate the execution of *ProSec Modeler*, we present an evaluation walkthrough based on an Industrial IoT sensor data collection process: Imagine an SME systems analyst tasked with ensuring that an upcoming IoT deployment complies with strict industrial security rules. Operating without a dedicated security expert, the analyst risks missing element-specific flaws (such as unauthenticated data transmission) when forced to rely on traditional, disconnected spreadsheet checklists.

ProSec Modeler guides the analyst through a localized, context-aware workflow that helps reduce common manual oversight errors, automatically maps data object configurations, and provides immediate visual feedback on the system's compliance health. The only prerequisite is that a standard BPMN model of the process under consideration exists. First, the analyst imports the XML file of the chosen BPMN process into the canvas environment. In this case, the included "*Sensor Data Collection Demo*" is used. At this stage, the process does not differ from a typical BPMN process model. The canvas shows grey badges on every pool, task, and message event where questions are to be answered. These are initialized to baseline coverage targets (0/3 for pool-level attributes and 0/9 for task-level attributes) directly on the elements, showing the total number of applicable requirements from the norm. Next, the analyst can extend the process with relevant security information. To do this, the analyst must walk through the process model step by step and respond to potentially important questions. When the analyst selects a lane, task, or message event in a process, the relevant questions for the selected BPMN element appear in the sidebar menu on the right side.

As shown in Figure 2, selecting the task ("*Transmit securely*") shows nine potentially relevant questions in the sidebar menu. For the other task, the same eight questions appear. Selecting the lane reveals two questions that may be relevant to the entire machine. Additionally, selecting the communication task displays two questions about data transmission (both based on the current mapping for this publication, which we will extend in the future). The completion status of the questions for all elements can be viewed in the overview on the left side. For any question, the analyst can determine that it is "Not relevant" to the process; in this case, no adaptations are made in the model.

Currently, each requirement's answer can be a Boolean ("yes" or "no"), a choice of predefined values (e.g., Authentication Mechanism: Password, Biometric, MFA, Certificate), a collection of arbitrary specific values, or a single value (e.g., port number). The answered questions are then mapped onto the process model as shown in Figure 2. Every question has an associated attribute that appears in a `TextAnnotation` with the corresponding answer. The `TextAnnotation` is connected to a `DataObject` named after the topic (e.g., "*IoT Ecosystem Requirements*"). This `DataObject`, in turn, is connected to the process element to which the question was answered. To prevent visual overlap of the generated

attributes in the process model, use the ”Arrange” button in the right-hand corner. This function places the `DataObjects` and `TextAnnotations` in a structured, systematic way to avoid overlaps and collisions among BPMN elements in the process model. To test compliance drift directly within the workspace, the analyst creates at least one separate operational session (”As-Is”) using the interface header controls. In the *Comparison* section below the modeler, the differences between the ”To-Be” and ”As-Is” processes are shown for every BPMN element, with gaps highlighted in red.

4. Discussion & Future Roadmap

The architecture replaces decoupled, external checklists with a unified, context-aware modeling canvas. Running on a centralized server, the system supports concurrent multi-user collaboration out of the box through a shared database. This avoids any changes to the underlying entity-relationship schema. SMEs frequently operate in a compliance vacuum where ”nobody truly knows what needs to be done, who is responsible for doing it, or why certain security steps are mandatory”. By lowering the technical barrier to entry, *ProSec Modeler* acts as an interactive guide for non-specialist personnel. It translates abstract compliance catalogs (like OWASP ISVS [1]) into localized, element-level questions, effectively compensating for the lack of specialized enterprise security roles and providing clear, auditable documentation out of the box.

While this work focuses on tool architecture and execution feasibility, formal empirical user studies and performance evaluations with industrial partners remain key directions for our future work. Our roadmap follows two distinct paths: *(i)* building automated reporting bridges to dump session validation states directly into structured JSON/CSV auditing outputs and *(ii)* replacing the localized greedy layout routine with global layouting to completely avoid annotation clipping on the canvas.

5. Conclusion

ProSec Modeler bridges the gap between process design and compliance engineering [7]. By binding dynamic schema fields directly to structural element keys, the system eliminates tracking isolation errors and provides visual and structural mechanisms to detect compliance drift during deep model iterations. The underlying framework is highly extensible: adding standard models such as ISO 27001 requires only basic data insertions into schema rows (cf. [4]), bypassing frontend code changes entirely.

Declaration on Generative AI

During the preparation of this work, the author(s) used Google Gemini (Gemini 3.1 Pro, Gemini 3.5 Flash) for grammar and spelling checks. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

[1] OWASP Foundation, IoT security verification standard (ISVS) v1.0, 2024. URL: https://owasp.org/www-project-iot-security-verification-standard/.

[2] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, 2nd ed., Springer, 2018.

[3] L. M. Kölbel, M. Hornsteiner, S. Schönig, Process-oriented security compliance for industrial iot systems: Formal modeling and integration, in: Proceedings of the 21st International Conference on Evaluation of Novel Approaches to Software Engineering, SciTePress, 2026, pp. 757–768.

[4] L. M. Kölbel, L. Poss, S. Schönig, A unified security requirements catalog for the industrial iot, in: Research Challenges in Information Science (RCIS), Springer Nature Switzerland, 2026, pp. 525–541.

[5] D. Basin, M. Clavel, M. Egea, A decade of model-driven security, in: Proceedings of the 16th ACM Symposium on Access Control Models and Technologies (SACMAT), 2011, pp. 1–10.

[6] A. Rodríguez, E. Fernández-Medina, M. Piattini, A bpmn extension for the modeling of security requirements in business processes, IEICE Transactions on Information and Systems E90-D (2007) 745–752.
[7] M. Leitner, S. Rinderle-Ma, A systematic review on security in process-aware information systems, Information and Software Technology 104 (2019) 1–13.