One Passport to Govern Them All: Bringing Order to IoT Security and Compliance*

Sara Nieves Matheu García and Antonio Skarmeta†

University of Murcia, Murcia, Spain
{saranieves.matheu, skarmeta}@um.es

Abstract

The increasing complexity of the Internet of Things ecosystems has exposed critical gaps in visibility, traceability, and governance of device security throughout the supply chain. Current practices rely on fragmented descriptors that, although valuable individually, lack a unified framework for integration, traceability, and lifecycle management. These limitations are particularly pressing in light of emerging regulatory requirements, most notably the EU Cyber Resilience Act (CRA), which demands structured, transparent, and auditable security documentation. To bridge this gap, we introduce the Device Security Passport (DSP), a structured, extensible, and lifecycle-aware model to consolidate and exchange security-related information about IoT devices. Built upon the Open Security Controls Assessment Language, the DSP integrates multiple security descriptors such as software and hardware bills of materials, vulnerability disclosures, and behavioral specifications, into a cohesive, hierarchical framework that evolves with the device from manufacturing to decommissioning. By allowing collaborative contributions from manufacturers, integrators, and operators, the DSP facilitates continuous security assurance, automated policy enforcement, and compliance with regulatory frameworks such as the CRA, thereby fostering greater transparency and accountability throughout the IoT supply chain.

Keywords: Cybersecurity · Transparency · Supply chain · Internet of Things · CRA · SBOM · OSCAL.

1 Introduction

The rapid proliferation of the Internet of Things (IoT) has transformed the digital landscape, embedding connected devices into nearly every domain of society—from critical infrastructure and industrial control systems to consumer electronics, vehicles, and smart cities. While these innovations bring significant benefits, they also introduce an expanded and increasingly opaque attack surface [7]. One of the most pressing challenges emerging from this evolution is the lack of visibility into the security posture of IoT devices across the entire supply and operational chain. Devices are often composed of numerous software and hardware components sourced from a variety of vendors, assembled and deployed by integrators, and operated in contexts far removed from their original design intent. At each of these stages, the available security-related information is typically fragmented, inconsistent, and difficult to interpret or verify.

The problem is further compounded by the absence of standardized mechanisms for documenting, sharing, and maintaining device-specific security information in a trustworthy and machine-readable form. In many cases, security attributes—such as known vulnerabilities,

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. S1, December 16-18, 2025, Sapporo , Japan.

†Corresponding author

cryptographic configurations, or software dependencies—are either undisclosed or only partially disclosed by manufacturers. Integrators and operators are thus forced to rely on limited documentation or perform their own assessments, leading to inconsistent evaluations and delayed remediation actions. Furthermore, when vulnerabilities are discovered post-deployment, the lack of traceability between a device and its underlying components makes coordinated response and patching both time-consuming and unreliable.

These issues are not only technical but also regulatory. With the advent of policies such as the European Union's Cyber Resilience Act (CRA) [2], manufacturers and distributors of digital products are now required to demonstrate secure-by-design practices, disclose software compositions, handle vulnerabilities transparently, and maintain documentation that spans the product lifecycle. The CRA and similar frameworks represent a shift from optional best practices to enforceable obligations, especially for connected devices. However, the reality on the ground is that most IoT vendors are ill-equipped to provide structured, traceable, and lifecycle-aware documentation that would enable compliance with such mandates. Existing efforts are often ad hoc, proprietary, or embedded in formats that are not designed for interoperability or automation.

In response to these challenges, we propose a comprehensive and standardized approach to document and share security-related information of IoT devices throughout their lifecycle. Central to this proposal is the concept of a Device Security Passport (DSP): a digitally signed, versioned, and extensible document that consolidates all relevant security descriptors associated with an IoT product. Inspired by the Open Security Controls Assessment Language (OSCAL) [4] developed by the U.S. National Institute of Standards and Technology (NIST), the DSP is designed to serve as both a communication interface and a persistent security profile for each device. It enables manufacturers to describe the software composition, hardware dependencies, network behavior, cryptographic configurations, known vulnerabilities, and conformance claims of a product in a structured and verifiable way. More importantly, it also allows integrators and operators to contribute new observations, contextual data, and assessment results, enabling a collaborative and continuously evolving security representation.

The DSP is not conceived as a static compliance artifact, but rather as a living digital asset that evolves as the product moves from design and manufacturing to integration, deployment, and eventual decommissioning. Through the use of machine-readable formats, the DSP facilitates the automation of tasks such as vulnerability scanning, onboarding configuration, security policy enforcement, and incident response. It serves as a single source of truth, linking traditionally siloed activities and actors in the supply chain. In doing so, it allows for faster identification of risks, more accurate evaluations, and more coordinated remediation efforts.

This paper presents the conceptual design of the DSP, identifies its key components, and outlines its alignment with existing and emerging regulatory frameworks. By establishing a unified model for security documentation, the DSP aims to bridge the gap between security engineering, operational assurance, and regulatory compliance, paving the way for a more resilient and transparent IoT ecosystem.

2 A landscape of security descriptors

In the IoT ecosystem, transparency regarding device composition and security posture has become critical. To enhance visibility across the supply chain and towards end-users, several descriptors have emerged to describe distinct aspects of device security, such as software and hardware components, cryptographic configuration, network behavior, mitigations or vulnerability exposure, all of which are essential for managing security risks and demonstrating

compliance.

The most well-known among these descriptors are the bills of materials (BOMs), which provide detailed inventories of a device's components. The most established example is the Software Bill of Materials (SBOM) [9], which lists all software elements included in a product—third-party libraries, firmware modules, and related metadata such as version numbers, license terms, and source information. SBOMs help identify known vulnerabilities by referencing public databases such as the National Vulnerability Database (NVD) or Common Vulnerabilities and Exposures (CVE) and support software tracking and license management. The DSP integrates SBOMs using widely adopted formats such as SPDX or CycloneDX to ensure consistency and interoperability. In addition to software inventories, the Hardware Bill of Materials (HBOM) [7] captures details about the physical components of a device, such as integrated circuits, sensors, and wireless modules, along with information about their origin and specifications. HBOMs support supply chain traceability and can help detect unauthorized substitutions or inconsistencies in hardware sourcing. While HBOM standardization is still developing, emerging formats such as CycloneDX-HBOM offer a basis for structuring this type of data. The DSP supports their inclusion to enhance visibility into the device's hardware structure. Beyond software and hardware, the BOM approach has also been extended to cover other areas of security relevance. The Cryptographic Bill of Materials (CBOM) [5] lists the cryptographic elements present in the device, including algorithms, libraries, certificates, and key management systems. CBOMs help verify compliance with encryption policies and allow for the identification of weak or outdated cryptographic methods that may need to be replaced. As cryptographic requirements evolve, particularly in the context of post-quantum readiness, the CBOM plays an increasingly important role in enabling responsive and systematic updates. Within the DSP, CBOM data is handled alongside SBOMs and HBOMs, offering a more complete picture of the device's internal architecture.

Although these BOMs provide visibility into what the device contains, additional descriptors describe how it is expected to behave and how it manages risk. The Manufacturer Usage Description (MUD), defined by the IETF in RFC 8520 [3], allows manufacturers to specify the network communication patterns that the device should follow. These statements can be used to configure access controls or network policies that limit exposure to unauthorized traffic. The DSP also incorporates Threat MUD documents, which include blacklist based on threat intelligence feeds. These profiles can help prevent contact with known malicious infrastructure and can be updated as threats evolve.

For vulnerability handling, the DSP integrates two additional types of descriptors, the Vulnerability Disclosure Report (VDR) [1] and the Vulnerability Exploitability eXchange (VEX) [8]. A VDR outlines the known vulnerabilities that apply to a product, their severity, and the status of the corrective actions. A VEX, in contrast, clarifies whether a specific vulnerability actually affects the device in practice. For example, a vulnerability may appear in a software library listed in the SBOM but may not be exploitable due to device configuration or unused code paths. Including both VDRs and VEX documents helps reduce noise in vulnerability assessments and offers better insight to operators, auditors, and regulators. The DSP supports these using formats like CycloneDX VEX, CSAF, and OpenVEX.

Beyond individual descriptors such as SBOM, MUD, VEX, or VDR, several standardized frameworks have been developed to provide a common structure for documenting and exchanging security information. The Security Content Automation Protocol (SCAP) [10], maintained by NIST, has been widely adopted for automating vulnerability management, security measurement, and compliance evaluation. SCAP bundles a set of specifications, such as CVE, Common Platform Enumeration (CPE), and Extensible Configuration Checklist Description

Format (XCCDF), enabling automated scanning and reporting of configuration and vulnerability data. However, SCAP's focus is primarily on technical checks and assessment automation, offering limited flexibility to capture broader lifecycle and governance information. More recently, NIST introduced the Open Security Controls Assessment Language (OSCAL) [4], a machine-readable set of models designed to represent system security plans, assessment procedures, test results, and remediation actions in a consistent and extensible way. Unlike SCAP, OSCAL is intended to serve as a general-purpose framework for documenting and exchanging security control information across diverse contexts. In the proposed DSP, OSCAL is not used in its full scope; instead, it has been adapted and simplified to suit IoT-specific requirements and the transparency obligations of the EU CRA. This adaptation allows the DSP to embed or reference other security descriptors such as SBOMs, MUD profiles, and VEX reports within a unified, lifecycle-aware model, enabling them to co-exist under a consistent schema while preserving their original formats and update cycles.

3 The Device Security Passport

The DSP is conceived as a structured model to capture and exchange device security information in a consistent, interoperable, and extensible way throughout the device lifecycle. In the following subsections, we describe the DSP lifecycle, its layered structure, and how OSCAL models have been adapted to represent both general and instance-specific security information.

3.1 The DSP lifecycle

The DSP is designed as a dynamic and layered model that evolves alongside the IoT device it represents, adapting to its lifecycle stages while maintaining a unified structure. Contrary to rigid and monolithic documentation approaches, the DSP introduces a dynamic and incremental model organized into three interoperable layers: the generic DSP (genDSP), the instance-level DSP (IDSP1), and the operational instance-level DSP (IDSP2). These layers are not different models but constitute a single adaptable structure, where each layer builds upon the previous one, refining and contextualizing the security profile of the device over time, as shown in Figure 1. This layered view reflects a progression rather than a separation. As the device transitions from manufacturing through integration to operational deployment, its security requirements, ownership context, and operating environment naturally evolve. The DSP is designed to follow this path, ensuring that security information remains traceable, relevant, and consistent throughout the supply chain.

The **genDSP** is created by the manufacturer during the design and development phase of the device. It serves as a baseline that contains static and general security information about the product. This includes SBOM, HBOM, CBOM, manufacturer and vulnerability disclosure details, expected behavior and policy recommendations (MUD), known vulnerabilities (VEX and VDR), and EU declaration of conformity. Each genDSP is cryptographically signed to ensure authenticity and can include both embedded content and external links to documents maintained on the manufacturer's infrastructure. The genDSP is version-controlled, allowing manufacturers to maintain historical records of changes and to distribute updates when new vulnerabilities are discovered or patches are made available. When third parties or users identify vulnerabilities, they can report them to the manufacturer, who can update the genDSP accordingly. Only modified sections are updated, and new versions retain backward references to prior versions, ensuring traceability and continuity.

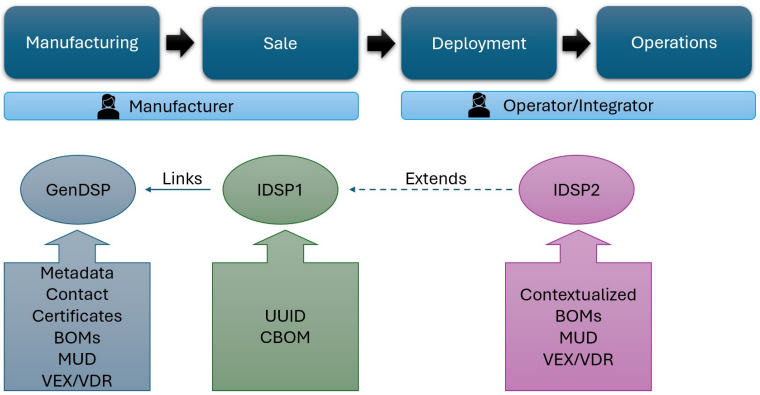

Figure 1: DSP lifecycle

As the device moves through the supply chain, **IDSP1** comes into play, binding the generic security profile to a specific physical unit of the device. It is generated by the manufacturer either at the end of production or at the point of sale and incorporates unique device identifiers such as IDs and public key material in the form of a CBOM. Although it links to the corresponding genDSP, it does not duplicate its content. Instead, it references it using pointers (e.g., URLs) so that users and systems can access the most current version of the underlying general information. This structure allows IDSP1 to act as a bridge between the manufacturer's baseline information and the operational context in which the device will be placed. Manufacturers can apply granular access controls to both genDSP and IDSP1, defining precisely which stakeholders can view particular sections, enabling controlled delegation of access to integrators or operators. In practice, this means that an integrator who buys a device can request access to the associated IDSP1.

At this stage, integrators and operators can enrich the IDSP1 with information that emerges after installation, such as updated SBOMs if new software libraries are added, revised HBOMs if hardware modifications are made, adapted MUD profiles to reflect custom configurations, or VEX and VDR records for vulnerabilities detected during operation. All this information is used to create the **IDSP2**, which is indeed, a local extension of IDSP1. The IDSP2 is used and maintained throughout the device's operational life, reflecting the actual conditions and security state of a specific unit in its deployment environment. Like the genDSP, it is inherently updatable, but while genDSP changes are manufacturer-driven and apply broadly to all devices of the same model, IDSP2 changes are instance-specific and typically initiated by integrators or operators. Maintained locally, the IDSP2 supports active security enforcement, such as applying network policies derived from its MUD profile or supplying contextual vulnerability information to monitoring and detections systems. As before, if the operator considers that a security finding may be relevant to the generic device, they can send this information to the manufacturer to request an update of the generic DSP. The manufacturer will then determine whether the information provided is accurate and if it should be incorporated into the generic DSP. Additionally, the manufacturer can provide extra configurations or patches to address or resolve the vulnerability.

Finally, as devices approach end-of-life, the DSP provides a valuable historical record of all security-related actions taken throughout the device's lifecycle. This record can inlcudes secure decommissioning procedures such as revoking cryptographic credentials and security certifi-

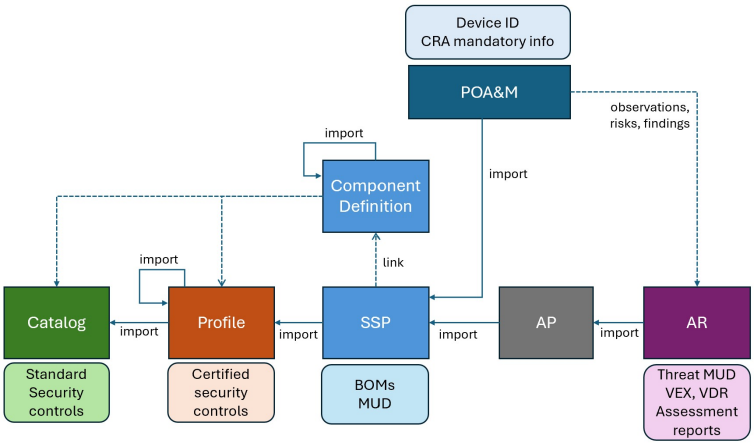

Figure 2: OSCAL mapping to security descriptors

cates but also support post-incident investigations, or guide procurement decisions for future deployments. In addition, it is designed to record the manufacturer's official support period and update availability, providing a clear reference for maintenance and security obligations beyond active use.

Throughout its lifecycle, the DSP works as a living record, progressively accumulating layers of detail without losing coherence. The genDSP provides a baseline, the IDSP1 binds it to physical instances, and the IDSP2 captures the aspects of the operational context. By unifying these layers into a single model, the DSP supports regulatory compliance, strengthens vulnerability management, and embeds security considerations into every stage of a device's lifecycle, fostering transparency and trust across the supply chain.

3.2 Structure

The DSP model is built upon OSCAL, an open and machine-readable framework developed under the leadership of NIST and supported in Europe through initiatives such as EUROSCAL[1] and the MEDINA[2] project. OSCAL was selected as the basis for the DSP because of its flexibility and expressiveness, its capacity to represent a wide variety of security-related information, and its inherent extensibility. Its adoption by key international actors such as IBM and European projects such as COBALT[3] or DOSS[4] ensures long-term stability, active community support, and a clear path for interoperability with future compliance tools and frameworks.

OSCAL is organized into several interconnected models, each describing a different facet of security information. These models are designed to work together, forming a structured ecosystem in which information can flow between related documents. At the top of this hierarchy is the Plan of Action and Milestones (POA&M) model, which serves as a root container capable of linking to other models and providing an overarching structure for risk tracking, mitigation planning, and associated security documentation. Given its role as the central node in OSCAL's architecture, the POA&M was chosen as the structural foundation for the DSP.

[1] https://euroscal.eu/
[2] https://medina-project.eu/
[3] https://horizon-cobalt.eu/
[4] https://dossproject.eu/

Each external descriptor, such as SBOMs, HBOMs, CBOMs, MUD profiles, Threat MUD lists, VEX and VDR, has been integrated into the OSCAL framework by mapping it to the most appropriate existing model or by extending the model with additional fields where necessary, as shown in Figure 2. A key architectural decision in our DSP model is linking external descriptors such as MUD files, SBOMs, and other essential security documents rather than embedding them directly within OSCAL. By doing so, each descriptor can evolve independently, with updates applied to a single source without forcing a regeneration of the entire DSP. Moreover, by maintaining external references, responsibility and authority over these files clearly remain with their original creators, ensuring accountability and preserving the integrity and accuracy of each descriptor. This approach of decoupling the security information from the device information is also recommended by CycloneDX, as while the composition of a device remains relatively stable, security information is much more dynamic and subject to frequent changes. The mapping also ensures that duplicated information is avoided, since details already covered in an external descriptor are removed from the original OSCAL structure. At the same time, new DSP-specific fields have been introduced to capture CRA-required transparency data that are not natively represented in any of the existing OSCAL models.

While a complete view of the DSP structure can be found here[5], we briefly describe the main building blocks:

- The DSP UUID and metadata form the entry point of the model. The UUID, following the notation described in RFC 4122 [6], uniquely identifies the DSP, while the metadata section provides essential contextual information about the document itself. This includes its title, version, revision history, date of last modification, roles and responsible parties, and references to supporting documentation such as EU conformity statements. In the case of IDSP documents, the metadata must also include a reference link to the corresponding genDSP, ensuring traceability across layers.

- The System ID block contains the unique identifier of the physical device, also compliant with RFC 4122. It is mandatory in IDSP.

- The Local definitions block consolidates several types of information that describe both the device and its security posture. It allows the DSP to specify components, assessment platforms, and assessment results, as well as to capture observations and risks identified during evaluation or operation. Within this block, the components subfield details each element's type, purpose, operational status, and description, with links to external descriptors such as SBOM, HBOM, CBOM, and MUD files. The CBOM is mandatory at the IDSP1 level, since it provides the cryptographic identity and material that uniquely bind a digital passport to a specific device instance. To avoid redundancy, certain OSCAL subfields, such as *protocols*, have been removed because their content is already covered in MUD profiles. The assessment-assets subfield is used to record tools, platforms, or additional components used in testing. Alongside this, the block includes an observations section, which documents findings from assessments or operations, specifying their type, collection method, supporting evidence, validity period, and links to detailed reports where available. Complementing this, the risks section catalogues identified vulnerabilities or threats, describing their status, associated CVSS vectors, remediation measures (including actions derived from Threat MUD), the responsible party for mitigation, and their relationship with recorded observations.

[5] https://github.com/saranieves92/DSP/tree/main

- The Imports block is an optional element included in the DSP to enhance flexibility. Its role is to link an external OSCAL models whenever such documentation is available. This allows security test outcomes, evidence, evaluation and system details to be referenced directly in their native OSCAL format, avoiding the need to duplicate them within the DSP structure. In cases where no OSCAL model is imported, equivalent information can instead be captured within the Local definitions block.

It is worth noting that DSP files contain information that may be used to configure a device or determine its security status, making them inherently sensitive. To ensure their integrity and prevent tampering, DSP files are digitally signed. Following the MUD approach, the signature is linked in the metadata block. Prior to processing the rest of a DSP, the DSP signature file must be retrieved and validated. In any case, the owner of the DSP is the one in charge of signing the file (e.g., manufacturer or integrator).

4 Alignement with CRA

The European Union's Cyber Resilience Act (CRA) [2] marks a pivotal shift in the regulatory landscape, establishing stringent, lifecycle-spanning cybersecurity obligations for manufacturers of digital products. Its core mandate is to enforce a security-by-design and security-by-default approach, requiring transparent documentation, proactive risk management, and continuous support. The DSP's OSCAL-based structure provides a coherent model to support the CRA's requirements systematically. Rather than creating a disjointed set of documents, it unifies all necessary compliance evidence into a single, yet layered, digital asset. The following breakdown illustrates how the DSP's components directly fulfill specific CRA mandates:

- Manufacturer Transparency and Vulnerability Disclosure: The CRA emphasizes the need for a clear identification of the economic operator and accessible channels to report vulnerabilities. The DSP embeds this information within its *Metadata/Parties* section, detailing the manufacturer's identity and contact points. Crucially, it uses *Metadata/Links* to provide direct references to the manufacturer's coordinated vulnerability disclosure (CVD) policy, ensuring a clear and unambiguous path for security researchers and users to report issues.

- Product Identification and Traceability: To ensure accountability throughout the supply chain, the CRA requires unique product identification. The DSP addresses this mandate through two key elements: the *System ID* block, which provides a unique identifier for the physical device instance (in IDSP), and the *Local definitions/Component* section, which records the product name, type, and model. This dual approach ensures traceability at both the product family and individual device levels.

- Documentation of Functionality and Security Properties: Manufacturers must document the intended purpose, essential functions, and security properties of their products. While high-level descriptions can be captured within *Local definitions/Component*, the DSP's extensible design allows for *Metadata/Links* to point to comprehensive external technical documentation. This ensures that the DSP itself remains lightweight and manageable while providing full transparency into the device's capabilities and inherent security features.

- Risk Assessment and Communication: A cornerstone of the CRA is the obligation to assess and transparently communicate foreseeable cybersecurity risks. The DSP operationalizes

this requirement through its *Local Definitions/Risks* section. Here, manufacturers can provide structured descriptions of identified risks, including potential misuse scenarios. Furthermore, this section can link directly to VEX or VDR reports, offering stakeholders nuanced context on whether a known vulnerability is actually exploitable in the specific product, thereby demonstrating proactive and informed risk management.

- Ongoing Support and Lifecycle Management: The regulation requires clear communication on the duration of security support, update mechanisms, and procedures for secure decommissioning. The DSP captures these elements within Metadata/Links, which can reference policies on update types, support timelines, and end-of-life procedures. For secure operational practices, *Local Definitions/Components/Links* reference MUD files to guide secure network configuration, while *Local Definitions/Risks/Remediations/Links* can point to Threat MUD files, providing dynamic guidance for mitigating emerging threats.

- Software and Hardware Transparency: The CRA's requirement for an SBOM, is directly embedded in the DSP's architecture. *The Local Definitions/Components/Links* section provides dedicated references to SBOMs, HBOMs, and CBOMs. This offers verifiable insight into the software composition and hardware supply chain, which is critical to assessing integrity and responding to component-level vulnerabilities.

- Declaration of Conformity: Finally, to complete the compliance picture, the DSP's *Metadata/Links* section includes a reference to the EU Declaration of Conformity. This provides the necessary auditable evidence that the product meets the essential requirements of the CRA and other applicable regulations.

Through this alignment, the DSP functions as far more than a static compliance checklist. It acts as a dynamic compliance enabler, significantly reducing the burden on manufacturers and integrators. By offering a standardized, machine-readable, and lifecycle-aware format, it streamlines the process of demonstrating adherence to the CRA. It transforms regulatory obligations from a bureaucratic exercise into an integral part of the security engineering and operational workflow, fostering both technical interoperability and legal compliance.

5 Conclusions

The rapid growth of the Internet of Things has exposed the limitations of fragmented and static approaches to security documentation. In response to this challenge, this paper has introduced the concept of the DSP, a structured and extensible framework designed to provide transparency, traceability, and governance throughout the entire lifecycle of IoT devices. Built on the flexible and machine-readable foundation of OSCAL, the DSP consolidates diverse descriptors—such as SBOMs, HBOMs, CBOMs, vulnerability disclosures, and behavioral profiles—into a single, coherent model. Its layered architecture, composed of the genDSP, IDSP1, and IDSP2, allows security information to evolve in line with a device's journey from design, through integration, to its operational environment.

Unlike static and fragmented documentation approaches, the DSP is conceived as a dynamic, version-controlled, and machine-readable framework that can evolve alongside a device. This design allows it to support operational processes—such as onboarding, configuration management, and vulnerability handling—while at the same time meeting regulatory requirements for compliance demonstration and auditability. By maintaining a digitally signed, versioned, and

auditable record, the DSP provides manufacturers, integrators, and operators with a shared reference point for collaboration and accountability. This capability is particularly relevant in the context of the EU CRA, where obligations for transparency, vulnerability handling, secure configuration, and conformity evidence are directly addressed within its structure. In this way, the DSP not only streamlines day-to-day security management but also acts as a bridge between engineering practice and regulatory compliance, transforming the principles of security-by-design and compliance-by-design into actionable and verifiable processes.

Looking ahead, several research directions remain open. Tooling support will be critical to automate the generation, validation, and update of DSPs, ranging from libraries for DevSecOps pipelines to interactive dashboards for operators. Interoperability challenges will also need to be addressed, particularly in distributed or federated settings where multiple DSPs coexist and share information. Finally, the structured nature of the DSP creates opportunities for machine-driven security management, enabling applications such as compliance-as-code, automated policy enforcement, and risk scoring.

In conclusion, the DSP moves IoT security from static and fragmented reporting towards a model of continuous, collaborative, and transparent assurance. By embedding security and compliance information into every phase of the device lifecycle, it lays the foundation for a more resilient, trustworthy, and accountable IoT ecosystem.

6 Acknowledgments

The work has been produced with the support of a 2024 Leonardo Grant for Scientific Research and Cultural Creation, BBVA Foundation. The Foundation takes no responsibility for the opinions, statements and contents of this project, which are entirely the responsibility of its authors. Research also partially supported by the EU through the Horizon research and innovation program DOSS (grant agreement no. 101120270) and COBALT (grant agreement no. 101119602). The publication is also part of the VINCI project PDC2023-145924-I00, funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR.

References

[1] Jon Boyens, Angela Smith, Nadya Bartol, Kris Winkler, Alex Holbrook, and Matthew Fallon. Cybersecurity supply chain risk management for systems and organizations. 2022.

[2] Europen Commission. Cyber resilience act, 2025. [Online; accessed Apr. 2025].

[3] Eliot Lear, Dan Romascanu, and Ralph Droms. Manufacturer Usage Description Specification (RFC 8520), 2019.

[4] National Institute of Standards and Technology (NIST). Oscal model documentation.

[5] OWASP. Authoritative guide to cbom, 2024. [Online; accessed 2025-09-05].

[6] RFC4122. A universally unique identifier (UUID) URN namespace, 2005.

[7] Trevor Stalnaker, Nathan Wintersgill, Oscar Chaparro, Massimiliano Di Penta, Daniel M German, and Denys Poshyvanyk. Boms away! inside the minds of stakeholders: A comprehensive study of bills of materials for software systems. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA, 2024. Association for Computing Machinery.

[8] National Telecommunications and Information Administration (NTIA). Vulnerability-exploitability exchange (vex) – an overview, 2021. [Online; accessed 2025-09-05].

[9] U.S. Department of Commerce, NTIA SBOM Minimum Elements Working Group. Minimum Elements For a Software Bill of Materials (SBOM): Preliminary Report. Technical report, U.S.

Department of Commerce, National Telecommunications and Information Administration (NTIA), February 2023. Prepared by the NTIA SBOM Minimum Elements Working Group.

[10] David Waltermire, Stephen Quinn, Harold Booth, Karen Scarfone, and Dragos Prisaca. The technical specification for the security content automation protocol (scap) version 1.3. 2018.