Vehicle Security Operations Center for Cooperative, Connected and Automated Mobility

Kevin Mayer, Tina Volkersdorfer, Jenny Hofbauer, Patrizia Heinl, Hans-Joachim Hof
CARISSMA Institute of Electric, Connected and Secure Mobility (C-ECOS)
Technical University Ingolstadt
Ingolstadt, Germany
e-mail: {kevin.mayer | tina.volkersdorfer | jeh7703 | patrizia.heinl | hof}@thi.de}

***Abstract*—Security Operations Centers (SOCs) are well established in the general IT domain. They provide IT security services, including collecting and correlating data, detecting and analyzing cybersecurity incidents, and applying dedicated reactions to such incidents. With the increasing digital capabilities of modern vehicles, appropriate reactions to cybersecurity incidents for vehicles and their ecosystem should be applied, too. Therefore, we propose a novel architecture for a Vehicle Security Operations Center (VSOC) in a Cooperative, Connected, and Automated Mobility (CCAM) environment. The VSOC implements different boxes addressing data storage, analysis capabilities, event-processing procedures, response options, digital forensics capabilities, and threat-hunting activities. The architecture allows the VSOC to communicate with third parties such as manufacturer backends or cybersecurity service providers (e.g., threat intelligence). Furthermore, we evaluate the proposed VSOC against fourteen metrics, which result from related work and our contribution. Examples are autonomy, data aggregation, coverage, inclusion of people, addressing physical assets, and supporting real-time safety.**

***Keywords-automotive; cyber security; security operations center; vehicle; vehicle security operations center; defensive; detection*.**

I. Introduction

Modern vehicles introduce a variety of different services and features. Examples include smartphone integration in modern infotainment systems, smart-home integration of vehicles, and vehicle-to-infrastructure communication. Those new services pose security risks to modern vehicles. The complexity and exposure of interfaces and the introduction of vehicle services increase by integrating vehicles into an ecosystem of connected entities. The European Parliament refers to the new ecosystem as Cooperative, Connected and Automated Mobility (CCAM) [1]. In this context, vehicles and ecosystem participants are connected, collaborate, and provide automated functionality. Participants in this environment aim to identify issues and join forces to mitigate them automatically. One key element in successfully implementing a CCAM is adapting security practices to mitigate security risks.

As a result, security practices must be evolved and adapted to keep up with these heterogeneous systems in CCAM environments. This adaption includes the area of detection and response tasks such as those implemented in Vehicle Security Operations Centers (VSOCs). In this regard, a VSOC should be capable of providing CCAM-specific services. Those include cooperative, connected, and automated features focusing on automotive-specific qualities such as safety, real-time, privacy, and legacy system characteristics. Based on these unique characteristics and the developments in the CCAM environment, we state the following three research questions:

***RQ1**: What data streams are relevant for a VSOC?*

***RQ2**: Which components of an VSOC are required in a CCAM environment?*

***RQ3**: Which information of a VSOC are beneficial to provide to CCAM participants?*

The research questions focus on implementing and evaluating a VSOC capable of handling a realistic amount of data. The data is within a connected environment (i.e., CCAM) while being heterogeneous and diverse. As a result, we present the following contributions:

- Identification of data streams (RQ1)
- Simplified and adaptable VSOC architecture (RQ2)
- Applicability on CCAM environments (RQ2)
- Identification of outgoing data (RQ3)

The remainder of the publication is structured as follows. Section II focuses on a literature survey showing work that aims to solve parts of our contributions. Next, Section III highlights critical aspects of a VSOCs followed by Section IV presenting our approach in implementing a VSOC for the CCAM environment. Section V evaluates the presented implementation. The publication concludes with suggestions for future research directions and a conclusion in Section VI.

II. Existing research and requirements for automotive security

First, we focus on research and requirements in automotive security. We specify the literature survey on works and regulations that highlight monitoring and defensive techniques in the vehicle ecosystem environment because those tasks are the primary responsibility of a VSOC.

Langer et al. establish an environment similar to a VSOC called "*Automotive Cyber Defense Center*" [2, pp. 98-122]. The authors present a theoretical implementation that aims to protect six layers: (1) Public mobility operation, (2) Original Equipment Manufacturer (OEM) mobility operation, (3) fleet operation, (4) vehicle operation, (5) vehicle network operation, and (6) Electronic Control Unit (ECU) operation. They further set requirements for the defense center by the ISO/SAE 21434 [3] and UN Regulation No. 155 [4] that lead to the following

metrics: (a) Reaction time, (b) criticality, (c) autonomy, (d) data aggregation, and (e) control-flow.

Hofbauer et al. identify metrics from IT-focused Security Operations Centers (SOCs) that apply for VSOCs too [5]. Those are (a) coverage of the VSOC, (b) people (including domain knowledge, analyst bias), (c) technical (including limitations, vulnerabilities, risks, safety implications, and incident), and (d) governance as well as compliance topics (e.g., regulations and identity/asset management).

Barletta et al. present a tool called "*V-SOC4AS*", a VSOC for improving automotive security in general [6]. The tool collects Controller Area Network (CAN) logs, converting them to a Syslog representation using the JSON format and sending them to the Security Information and Event Management (SIEM) (in their case, IBM Qradar). The focus of the VSOC implementation is on data from in-vehicle components.

Previous works highlighted the UN Regulation No. 155 [4] that states requirements regarding a Cyber Security Management System (CSMS). Vehicle OEMs must provide monitoring capabilities for their products (i.e., vehicles and backend services). However, the standard does not explicitly highlight the need for a VSOC.

In a whitepaper by NTT DATA, the authors introduce an Intrusion Detection System (IDS) for the in-vehicle CAN bus [7]. The collected data and identified anomalies are transmitted to the NTT DATA VSOC. The whitepaper highlights no additional implementation details. However, the authors indicate that IDS-related data can be valuable for a VSOC.

Menges et al. publish their General Data Protection Regulation (GDPR) compliant SIEM called "*DINGfest*" [8]. The implementation complies with legal requirements for pseudonymization while maintaining detectability. They defined boundaries for GDPR compliant architectures. The protectable data, regarding privacy aspects, are stored in a central repository.

Compared to existing related work, we provide a VSOC architecture that is suitable for diverse CCAM environments and addresses automotive-specific requirements such as moving endpoints (i.e., cars) and the use of proprietary technologies. Existing architectures can not fulfill the requirements one faces in a CCAM environment. Hence, our implementation focuses on the CCAM environment that utilizes in-vehicle data combined with vehicle ecosystem data.

III. Analysis and Design

An effective VSOC should follow principles proven by classical enterprise IT SOCs. Hofbauer et al. present for SOC metrics that should be adopted by effective VSOCs [5]. Those metrics are (a) coverage metrics on how many assets are monitored by a VSOC. (b) People metrics focus on analyst domain knowledge and analyst bias. (c) Technical metrics focus on limitations, vulnerabilities, risk and safety, and incident handling. (d) Governance and compliance metrics focus on compliance, identity, and asset management. In addition, as focused by Menges et al., (e) Data privacy concern metric is relevant for an effective VSOC [8]. Vehicles generate vast amounts of data, including sensitive information about occupants and behaviors. Protecting this data from unauthorized access and ensuring compliance with data privacy regulations (such as GDPR or CCPA) is critical to automotive SOC operations.

TABLE I
Identified metrics for Vehicle Security Operations Centers.

Metric	Source
Reaction time	Langer et al. [2]
Criticality	Langer et al. [2]
Autonomy	Langer et al. [2]
Data aggregation	Langer et al. [2]
Control-flow	Langer et al. [2]
Coverage	Hofbauer et al. [5]
People	Hofbauer et al. [5]
Technical	Hofbauer et al. [5]
Governance and compliance	Hofbauer et al. [5]
Data privacy concern	Menges et al. [8]
Physical assets	Our contribution
Real-time safety	Our contribution
Complex supply chain	Our contribution
Attack vectors	Our contribution

We further extend the metrics for an effective VSOC with the following: (f) Physical assets metric because automotive ecosystems involve physical assets such as vehicles, sensors, and infrastructure, unlike classical IT environments, which predominantly deal with virtual assets like servers and databases. It means the threats an automotive SOC faces include physical tampering, theft, sabotage, and digital attacks. (g) The real-time safety concerns metric is that security breaches can directly impact safety in automotive environments, leading to potentially life-threatening situations. Therefore, VSOCs must not only focus on data breaches and system compromises but also on ensuring the vehicles' and their occupants' safety and integrity. (h) Complex supply chain metric since the automotive industry involves a complex ecosystem of suppliers, manufacturers, and service providers, leading to a broader attack surface than classical IT environments. SOCs in automotive ecosystems must consider the security implications of the entire supply chain, including third-party components and software. (i) Attack vector metric since automotive systems are susceptible to unique attack vectors such as remote hacking of vehicle electronics, GPS spoofing, and manipulation of connected infrastructure (e.g., traffic lights). SOCs in automotive environments must be equipped to detect and respond to these unconventional threats.

As a result, the metrics from Hofbauer et al., Menges et al., Langer et al., and our extension lead to a foundation for effective VSOCs in a CCAM environment. Hence, instead of collecting in-vehicle data (e.g., from an IDS) only, we suggest extending the coverage of the VSOC to the vehicle ecosystem. All metrics are summarized in Table I.

IV. Implementation

The implementation focuses on fulfilling the proposed metrics. Furthermore, we aim to allow organizations such as

OEMs to adapt the architecture. As a result, the concept must be technology-independent, expandable, modular, and follow the KISS principle to "*keep it simple {and} stupid*" [9, p. 21].

The following sections will highlight the outside and inside views of the proposed VSOC implementation.

A. Outside view

The main interface to exchange information with participants and the VSOC is the VSOC API. It allows the introduction of technology-independent interfaces for communication between vehicle ecosystem participants and the monitoring entity (i.e., VSOC). We use HTTP REST for the proposed CCAM VSOC. HTTP(S) REST is well-documented and allows us to follow best practices. Various APIs from internet services use it. As a result, the VSOC provides three main services to the outside CCAM world:

- VSOC API as a communication interface based on the HTTP REST architecture style.
- Collecting predefined input data through the API based on defined and documented communication channels.
- Providing services for CCAM participants that are communicated through the API.

As highlighted, the communication method is the API. We utilize two specific communication methods while following the KISS principle: (a) CCAM participants implement their own HTTP REST client or server. It allows them to communicate with the VSOC and subscribe to relevant endpoints. (b) CCAM participants install SIEM-specific tooling to exchange information with the VSOC. One example would be the *Splunk Universal Forwarder* if Splunk is used as a SIEM. Figure 1 illustrates the outside view based on the presented technologies.

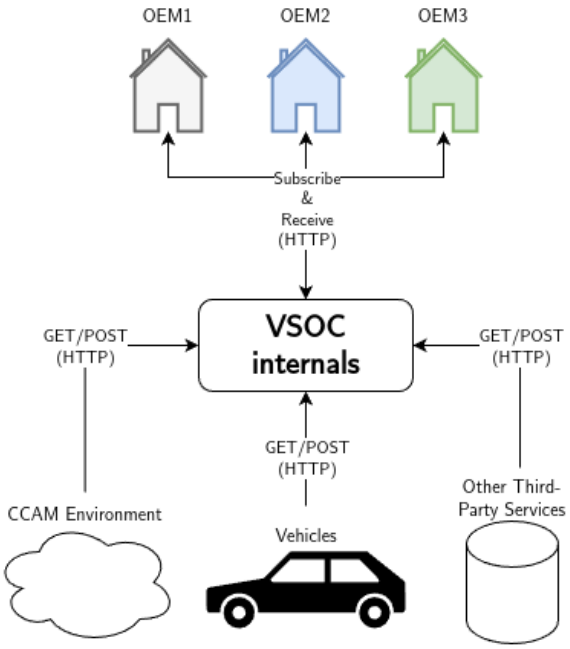

Figure 1. Outside view of the Vehicle Security Operations Center.

B. Inside view

Next, we introduce the inside view of the VSOC. Here, we follow a similar structure from IDSs. Vehicle IDS have been shown as a suitable method to manage cybersecurity events in automotive systems [10, p. 2774-2779] [11, p. 117-123][12, p. 185489-185502][13, p. 2531-2533][14][15, p. 1-9]. Hence, we follow a similar architectural structure in the VSOC.

Figure 2 illustrates the complete inside view and data streams.

Bidou took a similar approach [16]. However, we extend the pure IDS architecture with CCAM specifics such as Digital Forensics (DF) aspects for court-ready (requires an extensive amount of documentation and attributes such as reproducibility of the investigation results) event reconstruction and reporting capabilities. Due to vehicles' safety implications, this aspect is relevant for CCAM environments. As a result, we argue that cybersecurity incidents tend to lead to legal actions. Another extension is the Threat-informed Management System (TiMS) component. This component provides a knowledge foundation to facilitate defense and identification services. The diversity of components in CCAM environments enables the occurrence of complex cyberattacks that can evade typical IDS. Therefore, assisting threat hunting is significant to ensure security in a CCAM ecosystem. Based on the classical IDS structure and an adaption to CCAM, we propose the following components:

- D-box: data repository separated in (raw) data storage and a repository for knowledge in a dedicated format (e.g., ontologies or knowledge graphs).
- A-box: implements analysis capabilities.
- E-box: main system after the events are received. Distribute them accordingly.
- R-box: submits responses to external systems based on A-box results and using D-box data.
- F-box: provides forensic capabilities for investigations.
- TiMS-box: complements the A-box by implementing threat-analysis capabilities based on design information for threat hunting activities.

1) D-box: The D-box, or data box, serves as a repository within the system architecture, encompassing raw data storage and a dedicated repository for structured knowledge organized in formats like knowledge graphs. The primary requirement for the D-box lies in its ability to store vast amounts of data while also providing mechanisms for structuring and organizing this data into meaningful formats. It is the foundational element upon which other components of the VSOC internal architecture rely, necessitating robust storage capabilities and data retrieval mechanisms. Additionally, the D-box must facilitate seamless integration with other boxes, enabling easy access to raw data and structured knowledge for downstream processes.

The D-box's needs revolve around scalability, flexibility, and interoperability. It must be capable of accommodating diverse data types and formats, ranging from structured to unstructured

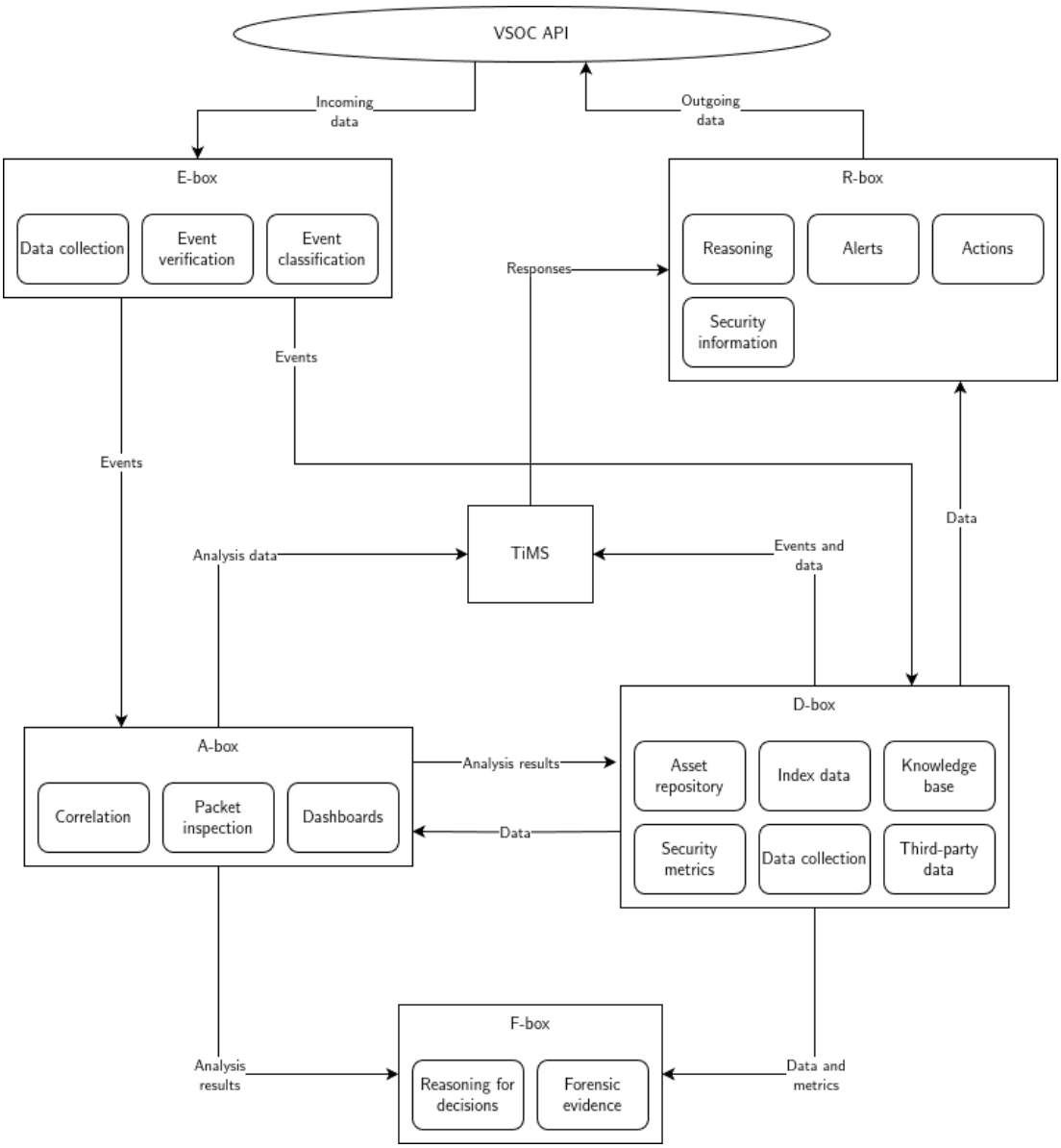

Figure 2. Inside view of the Vehicle Security Operations Center.

data, and scaling seamlessly to handle growing data volumes. Furthermore, the D-box should support interoperability with various data sources and formats, enabling seamless integration with external systems and data streams. Ensuring data quality, security, and privacy is also paramount, necessitating robust data validation, access control, and encryption mechanisms. Privacy principles are adapted by following guidelines from Menges et al. [8][17].

The D-box's capabilities include data storage and retrieval, support for structured knowledge representation, and seamless integration with other system components. It enables semantic querying and reasoning over stored data by leveraging knowledge graphs, facilitating advanced analytics and decision-making processes. Moreover, it is a centralized repository for shared knowledge within the system, enabling consistent interpretation and understanding of data across different components.

The D-box's limitations primarily revolve around scalability challenges, potential performance bottlenecks, and complexities of managing diverse data types and formats. As data volumes grow, the D-box may face scalability limitations, requiring careful design considerations and optimization strategies to ensure optimal performance. Also, managing heterogeneous data sources and formats can introduce integration and interoperability complexities, potentially leading to inconsistencies or data quality issues.

2) A-box: The A-box, or analysis box, constitutes a critical component within the system architecture that implements advanced analytics capabilities. Essential requirements for the A-box include the ability to provide complex data analysis tasks,

such as statistical analysis, machine learning, and predictive modeling, on the data received from the D-box. Furthermore, in the CCAM environment, it propagates a trust score that addresses the trustworthiness of components. It must also support real-time or near-real-time processing to enable timely insights and decision-making.

The primary need of the A-box lies in its capability to derive actionable insights and intelligence from the vast amounts of data ingested from the D-box. It offers advanced analytical algorithms, models, and techniques tailored to the specific domain or application context. Furthermore, the A-box must handle diverse data types and formats, ranging from structured to unstructured data and support scalability to accommodate growing data volumes and computational requirements. However, the D-box will achieve the normalization of data, and the A-box will implement its capabilities based on the data the D-box provides.

The A-box's capabilities include advanced analytics, machine learning, predictive modeling, anomaly detection, and pattern recognition. By leveraging sophisticated algorithms and techniques, the A-box enables the extraction of valuable insights and patterns from complex datasets, empowering decision-makers with actionable intelligence. Moreover, it supports iterative model training and refinement, enabling continuous improvement and adaptation to changing data dynamics.

The A-box's limitations primarily revolve around computational complexity, resource constraints, and the need for domain-specific expertise, specifically in the automotive domain. Performing advanced analytics tasks on large-scale datasets can be computationally intensive, requiring significant computational resources and infrastructure. Additionally, designing and implementing effective analytical models often necessitate expertise in data science, statistics, and domain knowledge, which may pose challenges in resource-constrained environments.

3) E-box: The E-box, or event box, is the primary system after receiving events. Essential requirements for the E-box include event processing, routing, and distribution functionalities. It must be capable of receiving events from external sources, processing them in real-time or near-real-time, and distributing them to downstream components or subsystems. In the case of the proposed CCAM VSOC, the E-box receives events from the HTTP REST. The current HTTP REST interface does not fulfill real-time requirements. However, we argue that the current implementation does not require real-time since decisions and actions are verified by humans regardless. API and distributes them to other boxes.

The primary need for the E-box revolves around its ability to effectively handle incoming events and ensure timely processing and distribution within the system. It implements robust event processing capabilities, fault tolerance, and scalability to accommodate varying event volumes and processing requirements. Furthermore, the E-box must support event routing and filtering based on predefined criteria or rules, enabling targeted distribution to relevant components.

Capabilities of the E-box include event ingestion, processing, routing, and distribution. By leveraging event-driven architecture and real-time processing capabilities, the E-box enables rapid response to incoming events, facilitating timely decision-making and action. Moreover, it supports seamless integration with external systems and data sources, enabling interoperability and data exchange across disparate systems. The seamless integration is realized through the usage of HTTP REST. As highlighted in Section IV-A, participating entities implement HTTP REST capabilities.

The E-box's limitations include scalability challenges, potential performance bottlenecks, and event processing and routing complexities. As event volumes grow, the E-box may face scalability limitations, requiring careful design considerations and optimization strategies to ensure optimal performance. Additionally, managing event streams from diverse sources and ensuring reliability and fault tolerance can introduce system design and implementation complexities.

4) R-box: The R-box, or response box, is responsible for submitting responses to external systems based on the results generated by the A-box and utilizing data from the D-box. Essential requirements for the R-box include response generation, integration with external systems, and data retrieval from the D-box for contextual information. Again, the HTTP REST API is used as a communication interface.

The primary need for the R-box lies in its ability to effectively translate insights and intelligence derived from the A-box into actionable responses for external systems. It provides seamless integration with external interfaces and protocols and data retrieval mechanisms from the D-box to enrich responses with contextual information. Furthermore, the R-box must support adaptability and reconfigurability to tailor responses based on specific requirements or preferences.

The R-box's capabilities include response generation, integration with external systems, and data retrieval from the D-box. By leveraging insights and intelligence generated by the A-box and utilizing contextual information from the D-box, the R-box enables the generation of timely and relevant responses to external stimuli. Moreover, it supports interoperability with diverse external systems, enabling seamless data exchange and communication.

The R-box's limitations primarily revolve around integration challenges, scalability constraints, and the complexity of response generation. Integrating diverse external systems and protocols can be challenging, requiring extensive customization and adaptation to ensure compatibility and seamless communication. Additionally, as response complexity and data volumes increase, the R-box may face scalability limitations, necessitating careful design considerations and optimization strategies to ensure operation.

5) F-box: The F-box, or forensics box, is particularly relevant in the context of the CCAM environment. Its primary purpose is to provide forensic capabilities tailored for event reconstruction within CCAM ecosystems. Essential requirements for the F-box include data preservation, traceability, and analysis functionalities specific to the unique characteristics

of CCAM environments, such as vehicular communication networks and autonomous vehicle operations.

The need for the F-box stems from the inherent complexity and dynamic nature of CCAM environments, where interactions between connected vehicles, infrastructure, and other elements create a vast and constantly evolving data landscape. In such environments, incidents or anomalies may occur, necessitating detailed forensic analysis to reconstruct events, identify root causes, and facilitate corrective actions. The F-box must, therefore, support the preservation and collection of relevant data traces, including vehicle sensor data, communication logs, and environmental context, to enable comprehensive event reconstruction.

Capabilities of the F-box encompass a range of forensic techniques and tools tailored for CCAM environments. These include data acquisition and preservation mechanisms, data correlation and analysis algorithms, and visualization techniques for presenting reconstructed events. By leveraging advanced forensic methodologies (e.g., data normalization of vehicle data and correlation of functional as well as non-functional in-vehicle data), the F-box enables investigators to reconstruct complex sequences of events, analyze causality relationships and identify contributing factors, ultimately supporting effective incident response and mitigation efforts within CCAM ecosystems.

The relevance of the F-box in CCAM environments lies in its ability to address specific challenges inherent to vehicular communication networks and autonomous vehicle operations. In these environments, incidents or anomalies may have far-reaching implications, affecting safety, security, and operational efficiency. The F-box provides essential capabilities for reconstructing events within this context, enabling stakeholders to gain insights into the underlying causes of incidents, identify potential vulnerabilities, and enhance the resilience and robustness of CCAM systems.

We further argue that the need for a dedicated F-box lies in the relevance of CCAM environments. Investigations involving cars (i.e., touching safety aspects of participants and occupants) must be investigated with more focus on the quality of a Automotive Digital Forensics (ADF) investigation. Hence, the results of an ADF must be usable in front of a court.

6) TiMS-box: The TiMS-box is a feature in the VSOC that complements the A-box capabilities. The knowledge repository, utilized by the TiMS, collects different data sources, such as design, architectural, threat, and attack data. This data helps throughout threat hunting activities. It allows analysts to check for relevant aspects, such as critical endpoints or crown jewels of the overall architecture. Essential requirements for the TiMS-box include analysis and mapping functionalities of threat, attack, and design information within the CCAM ecosystem and its iterative application during threat-hunting activities, including the tractability of the design and threat information.

The need for the TiMS based on the dynamic and diverse CCAM ecosystem. This environment leads to complex, unique, and fast-changing threats that have to be considered to ensure safety. Instead of only waiting for alerts about incidents, active threat hunting activities complement the mainly passive and reactive activities of SOC analysts, e.g., determining the impact a compromised supplier has on systems and infrastructure. Therefore, a structured approach supports threat hunting activities within the CCAM environment. The generated, prioritized, and design-based attack paths, consisting of single steps, assist threat hunters in a guided way to detect, track, and disrupt threats as early as possible and throughout a complex threat.

The TiMS-box's capabilities include analysis algorithms, modeling, data mapping, graph theory, and knowledge representation techniques. Mapping of relevant threat (including attack and adversary) information (like Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege (STRIDE), Tactics, Techniques, and Procedures (TTP)s from MITRE ATT&CK) with related design and asset information within the CCAM represents the modeled knowledge foundation [18]. Using the VSOC API and E-box allows the integration and storage of the individual sources (e.g., form threat intelligence providers) into the D-box. Coming from the D-box (via VSOC API and E-box), an analyst requests iteratively relevant attack paths for a specific adversary group on this knowledge base. As a result, the R-box submits the generated and prioritized attack paths of design information back to the analyst.

The TiMS-box's limitations primarily include mapping adversary, threat, asset, and design information. This information has heterogeneous data sources, formats, and diverse abstraction levels. The A-box assists the TiMS-box in necessary analysis and prepossessing tasks, e.g., due to different formats of sources. However, input data integration, mapping, and representation face complexity challenges, including the complexity of attack path generation. Moreover, mapping attack and design information often necessitate expertise in threat hunting, security architecture, graph theory, and domain knowledge, which may pose challenges in resource-constrained environments. Finally, the quality of the generated attack paths and their abstraction level depend on the quality and abstraction of the available design and attack information as input for the TiMS.

C. Technical realization

The technical implementation is realized using Docker. Figure 3 illustrates the different docker-compose components. We further published the full VSOC implementation on GitHub [19]. Additional and in-depth implementation details can be found in the referenced GitHub repository.

The VSOC API uses HTTP REST as a communication protocol. Each tool and participant of the CCAM environment has their REST endpoint and required methods. We further use OpenTelemetry to generate metadata for the HTTP REST requests. It is an open-source and widely used tool that suits our requirements within the VSOC. It identifies user agents and other telemetry data that the A-box can use for anomaly detection. The OpenTelemtry logs are gathered using OTL Collector (an OpenTelemtry utility), which transmits the logs to an APM server. The APM server stores the data in the SIEM of the VSOC. In case of this implementation, we use

the ELK Stack as a SIEM. The ELK Stack is open-source, widely used in SIEMs, and well documented, which makes it a suitable technology for the VSOC. We use Elasticsearch as a search engine and Kibana for data visualization. Both are integrated in the ELK Stack out of the box. In the case of the Docker implementation, each part of the ELK Stack holds certificates for secure communication and the data in their corresponding Docker volumes. This approach separates the data from each system and makes the VSOC tool-agnostic to change the SIEM solution or other technical parts.

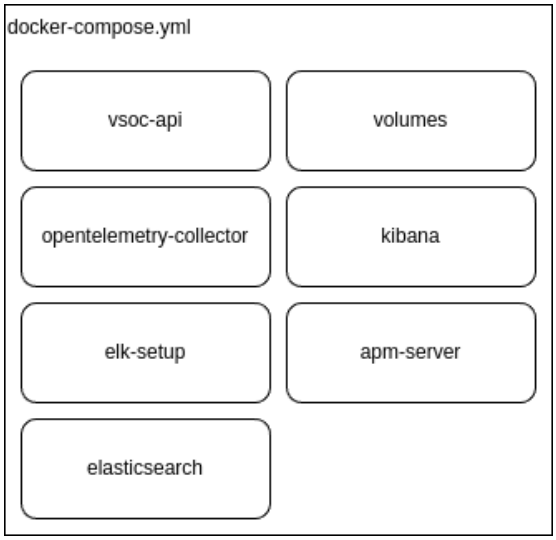

Figure 3. Docker compose components.

The different boxes (E-box, R-box, A-box, D-box, and F-box) are realized within the ELK Stack. For example, the Docker volumes used by the ELK Stack realize the D-box and F-box as storage units. The A-box is implemented within Kibana and Elastic search. Analysts can utilize both tools to evaluate events, visualize logs, and perform actions. The E-box and R-box are part of the VSOC API, while the ELK Stack realizes tasks such as event classification and verification.

We argue that, nevertheless, which SIEM technology is used (e.g., Splunk, ELK Stack, etc.), our proposed methodology can be adapted. In the case of this publication, we utilize the ELK Stack and highlight the implementation of our methodology. Other SIEM solutions might need additional adaptation to certain boxes.

V. Evaluation

We address the highlighted criteria from Langer et al. [2], Hofbauer et al. [5], Menges et al. [8], and our own to evaluate the highlighted VSOC for the CCAM environment. Table II presents the metrics and their fulfillment. The checkmark (✓) illustrates their complete fulfillment, and the bullet (•) is their partial fulfillment.

Our VSOC partially fulfills the **reaction time** metric by implementing alerting and automation within internal boxes. These features facilitate prompt identification and response to potential security threats, enhancing responsiveness. However, there is room for improvement in optimizing reaction times under real-world conditions.

TABLE II
Used criteria as evaluation metrics.

Fulfillment	Metric	Source
•	Reaction time	Langer et al. [2]
✓	Criticality	Langer et al. [2]
✓	Autonomy	Langer et al. [2]
✓	Data aggregation	Langer et al. [2]
✓	Control-flow	Langer et al. [2]
✓	Coverage	Hofbauer et al. [5]
•	People	Hofbauer et al. [5]
✓	Technical	Hofbauer et al. [5]
✓	Governance and compliance	Hofbauer et al. [5]
•	Data privacy	Menges et al. [8]
✓	Physical assets	Our contribution
•	Real-time safety	Our contribution
•	Complex supply chain	Our contribution
•	Attack vectors	Our contribution

The VSOC fully meets the **criticality** requirement by effectively flagging events with criticality tags in Kibana (ELK stack). This capability ensures that incidents are prioritized based on severity, allowing for efficient resource allocation and swift resolution of critical issues.

Autonomy is another area where our VSOC excels. The system incorporates a high level of automation within its internal processes, minimizing the need for constant human intervention. Human operators oversee critical aspects and events, ensuring essential decisions are scrutinized appropriately.

Our **data aggregation** capabilities are robust. We leverage the VSOC HTTP REST API alongside OpenTelemetry and the ELK stack. This combination enables comprehensive monitoring and analysis of security events across various data sources, fully meeting the data aggregation metric. We further tested the load using the Locust.

The VSOC's **control-flow** capabilities are robust. The implemented VSOC HTTP REST API, as well as the internal boxes, enables a simple way to communicate and control data between the diverse CCAM participants, independent of the endpoint, e.g., OEM or a vehicle-component.

The VSOC API is designed to collect data comprehensively from vehicles and the CCAM ecosystem. In addition, IT metadata is gathered using OpenTelemetry, providing extensive **coverage** and insight into the security landscape. This thorough approach ensures that we meet the coverage requirement.

The **people** metric is partially fulfilled due to the limited capacity to test the VSOC in real-life settings with an entire team of VSOC analysts. Despite this, integrating the ELK stack demonstrates the system's ability to augment human analysts' capabilities, enhancing overall performance.

We fully meet the **technical** metric by successfully collecting and analyzing technical information from various sources, including ECUs, wireless and wired communications, consumer electronics, vehicle components, and sensors. The VSOC does not perform automated blocking or reactions; instead, it provides suggestions to tools, maintaining a human-in-the-loop setup for critical decision-making.

Our VSOC adheres to UNECE R155 standards and incorporates identity and asset management through ontologies as a knowledge repository. This ensures that our operations comply with regulatory requirements and follow best governance practices, fully meeting the governance and **compliance** metrics.

Implementing the different boxes with their dedicated responsibilities partially fulfills the **data privacy** metric. Depending on the box, e.g., the D-box, the recommended requirements by Menges et al. to ensure legal compliance of the VSOC can be applied individually. However, the evaluation of detection performance and necessary near-real-time data in combination with, e.g., pseudonymization still needs to be concluded.

Collecting data from the ecosystem, including **physical attributes**, is another area where our VSOC excels. Tools like safety monitoring and collaboration tools (e.g., to propagate information using car-to-car messages) exemplify our ability to gather comprehensive data from various physical assets, fully satisfying the physical assets metric.

Our human-in-the-loop approach partially fulfills **real-time safety** by minimizing the risk of unintended consequences from automated reactions. While this conservative approach ensures safety, it limits the system's real-time safety impact.

Our use of ontologies to store and manage the complex relationships within the **supply chain** partially meets this metric. The effectiveness of this approach depends on the availability and accuracy of knowledge within the system.

The VSOC partially addresses various **attack vectors** by utilizing the TiMS. This feature considers diverse threat information about design information of CCAM technologies and generates prioritized attack paths. The provided knowledge informs and guides analysts about how CCAM-based attack vectors (e.g., GPS spoofing in a platooning) can occur. A step-by-step path and the related context information facilitate analysts' detection and response to these threats. However, the effectiveness of these features depends on the availability and quality of threat and design data. The evaluation of effectiveness and associated tests still need to be concluded.

In summary, our VSOC demonstrates strong performance across multiple metrics, with particular strengths in criticality, autonomy, data aggregation, control flow, coverage, technical capabilities, governance and compliance, and physical assets. Areas for improvement include reaction time, real-time safety, people, data privacy concerns, the management of complex supply chain relationships, and attack vector management.

VI. Conclusion and Further Work

We presented a VSOC architecture for the CCAM environment. It consists of six boxes to provide event processing, analysis, data storage, forensic, response propagation, and threat hunting capabilities. We implemented a proof-of-concept using Docker, the ELK Stack, and OpenTelemetry.

Our implementation highlights the relevant data streams for a VSOC (RQ1) in between the presented boxes. Furthermore, the architecture introduced the TiMS and F-box to address CCAM-specific requirements (RQ2). Finally, the API implementation and description of the boxes highlight relevant information that should be shared with other CCAM participants (RQ3).

Our present work does not yet consider Machine Learning (ML) approaches to assist VSOC analysts their decision-making. As various existing publications suggest applying ML algorithms for the detection, analysis, and automatic response to incidents in the IT environment, future work can investigate to which extent machine or deep learning can increase the performance of our proposed VSOC architecture. For that, the data that serves as input and output to the different VSOC components conceptualized in our work need to be processed and prepared for the respective learning task. Hence, further research has to be performed on feature extraction and selection. As it is crucial for the SOC analyst to understand the proposals of the ML model, further work should also consider the explainability and transparency of possible solutions.

Acknowledgment

This project has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101069748.

References

[1] European Partnership, *Ccam - connected, cooperative, and automated mobility*, url: https:www.ccam.eu, Accessed: 2024-09-19, Sep. 2024.

[2] F. Langer, F. Schüppel, and L. Stahlbock, "Establishing an Automotive Cyber Defense Center", in *17th Escar Europe: Embedded Security in Cars (Konferenzveröffentlichung)*, Oct. 2019. DOI: 10.13154/294-6652.

[3] International Organization for Standardization, "ISO/SAE 21434:2021 - Road vehicles - Cybersecurity engineering", International Organization for Standardization, Tech. Rep., Aug. 2021.

[4] UNECE, "UN Regulation No. 155 - Cyber security and cyber security management system", United Nations Economic Commission for Europe, Tech. Rep., 2020.

[5] J. Hofbauer, K. K. Gomez Buquerin, and H.-J. Hof, "From SOC to VSOC", in *21th Escar Europe : The World's Leading Automotive Cyber Security*, Ruhr-Universität Bochum, Oct. 2023. DOI: 10.13154/294-10389.

[6] V. S. Barletta *et al.*, "V-SOC4AS: A Vehicle-SOC for Improving Automotive Security", *Algorithms*, vol. 16, no. 2, p. 112, Feb. 2023, Collecting CAN logs, converting them to JSON (syslog representation), and send it to the SIEM (IBM Qradar). VSOC. DOI: 10.3390/a16020112.

[7] R. Bader, R. Katyal, and F. Capocasale, "Automotive Cybersecurity - An End-to-End Automotive Cybersecurity Solution Combining NTT DATA's Intrusion Detection System for CAN Bus with its State-of-the-Art Vehicle-Security Operation Center", NTT DATA Deutschland GmbH, Tech. Rep., 2021.

[8] F. Menges *et al.*, "Towards GDPR-compliant data processing in modern SIEM systems", *Computers & Security*, vol. 103, p. 102 165, Apr. 2021, ISSN: 0167-4048. DOI: 10.1016/j.cose.2020.102165.

[9] R. B. Misra, "Global IT Outsourcing: Metrics for Success of All Parties", in *Journal of Information Technology Cases and Applications*, p. 21, vol. 6, 2004.

[10] L. Yang and A. Shami, "A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles", in *ICC 2022 - IEEE International Conference on Communications*, IEEE, May 2022. DOI: 10.1109/icc45855.2022.9838780.

[11] I. Ahmed, G. Jeon, and A. Ahmad, "Deep Learning-Based Intrusion Detection System for Internet of Vehicles", *IEEE Consumer Electronics Magazine*, vol. 12, no. 1, pp. 117–123, Jan. 2023, ISSN: 2162-2256. DOI: 10.1109/mce.2021.3139170.

[12] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-Based Intrusion Detection System for In-Vehicle Can Bus Communications", *IEEE Access*, vol. 8, pp. 185 489–185 502, 2020, ISSN: 2169-3536. DOI: 10.1109/access.2020.3029307.

[13] H. Li *et al.*, "POSTER: Intrusion Detection System for In-vehicle Networks using Sensor Correlation and Integration", in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17, ACM, Oct. 2017. DOI: 10.1145/3133956.3138843.

[14] A. Leandros, "A Novel Distributed Intrusion Detection System for Vehicular Ad Hoc Networks", *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 4, 2015, ISSN: 2158-107X. DOI: 10.14569/ijacsa.2015.060414.

[15] D. Kosmanos *et al.*, "Intrusion Detection System for Platooning Connected Autonomous Vehicles", in *2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, IEEE, Sep. 2019. DOI: 10.1109/seeda-cecnsm.2019.8908528.

[16] R. Bidou, "Security Operation Center Concepts & Implementation", Semantic Scholar, 2005.

[17] F. Menges *et al.*, "Introducing DINGfest: An architecture for next generation SIEM systems", 2018, Initial architecture of DINGfest (SIEM)
Privacy aware SIEM. DOI: 10.18420/SICHERHEIT2018_21.

[18] T. Volkersdorfer and H.-J. Hof, "A Concept of an Attack Model for a Model-Based Security Testing Framework", in *SECURWARE 2020, The Fourteenth International Conference on Emerging Security Information, Systems and Technologies*, 2020.

[19] Security in Mobility, *SELFY VSOC*, https://github.com/securityinmobility/selfy-vsoc-api, Sep. 2024.