DOI 10.5162/iCCC2026/P31

# A Cybersecurity Approach for Visual Data Flow in Digital Twins: A Real-World Case Study

Hamid Zargariasl, Aghaali Faraj, Christian Borck, Martin Behm, Annamria Arnouk, Christian Herglotz,
Brandenburg University of Technology, Cottbus, Germany
zargaria@b-tu.de

## Abstract

We considered a digital twin system designed to handle visual information through virtual reality (VR) and fixed cameras. This system is part of a digital twin framework that has been proposed and managed over several years, specifically related to the production line of a component of an airplane engine. Several relevant tools are integrated into this framework using technologies such as HTTP, gRPC and WebRTC. Our approach involves analysing the topology of the digital twin system and conducting a threat analysis using the STRIDE model. Based on this analysis, we propose mitigation strategies to enhance the resilience of the current configuration against potential cyber-attacks.

## 1 Introduction

Digital twins are generally defined as virtual representations of physical assets, systems, or processes that are continuously updated with real-time data, enabling analysis, prediction, and optimization across the asset's lifecycle. A digital twin integrates the physical and virtual spaces through data, models, and services, forming a closed loop of information that supports monitoring, simulation, and decision-making. The digital twin as a three-part structure composed of the physical entity, the virtual entity, and the data connections between them, emphasizing its role in lifecycle management and predictive analytics. Together, these definitions highlight the importance of real-time data integration and bidirectional communication as foundational characteristics of digital twin systems ([1],[2]).

Digital twins have gained significant attention in recent years as a key enabler for integrating physical systems with their virtual counterparts, enabling real-time monitoring, analysis, and optimization. Researchers increasingly emphasize the role of digital twins in improving system performance, enhancing predictive maintenance, and supporting data-driven decision-making across diverse domains such as manufacturing, transportation, and smart cities. These studies illustrate the expanding scientific focus on digital twins as an essential component of cyber-physical systems ([1],[2]).

Handling visual information within digital twin systems poses significant challenges due to the large volume, high dimensionality, and real-time processing requirements of such data. Beyond the inherent computational complexity, these visual pipelines introduce additional vulnerabilities that must be addressed through robust cybersecurity approaches. As a result, ensuring secure and efficient processing of visual information has become a critical aspect of maintaining the reliability and resilience of digital twin environments.

This paper investigates a real-world environment for implementing a digital twin system, incorporating video information, where multiple components interact using diverse communication protocols. Our objective is to analyze the network topology to identify its most vulnerable points and to propose suitable mitigation methods against potential cyberattacks.

Chapter 2 provides an overview of the digital twin scenario under study. Chapter 3 details the network analysis methodology and the rationale behind the proposed mitigation strategies. Finally, the conclusion summarizes the key findings and discusses future research directions.

## 2 A Real World Digital Twin

We utilize a camera-based topology during the assembly of an engine component, as previously reported [3]. The digital twin is implemented using the Asset Administration Shell (AAS), which provides a standardized framework for representing and managing the asset's data and functions. AAS is a standardized digital representation of an asset in Industry 4.0. It defines how data, functions, and services related to an asset are structured and accessed. The hardware topology used in this process is depicted in Figure 1.

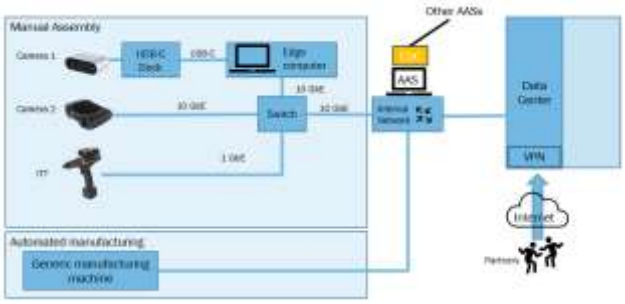



**Figure 1** The topology of connected hardware including digital twin (AAS node).

The visual system, comprising augmented reality (AR) and cameras, is capable of detecting any aspect of the work and automatically notifying the worker. This visual information is integral to the digital twin aspect of the system. Various tools, such as AFOS (Agent and Server), CTDS (Tacit Data Server), CTDS (Digital Assembly Master), Nucleus Server, NVIDIA 3D Asset Bridging, and others, are interconnected via HTTP, gRPC, and WebRTC. However,

DOI 10.5162/iCCC2026/P31

delve into those specific details here as our primary focus is on cybersecurity issues, which are addressed in Figure 2.

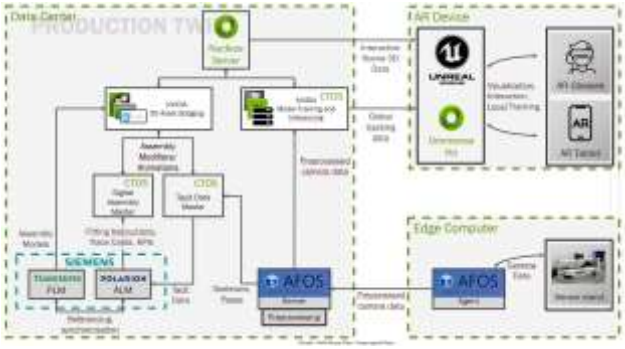



**Figure 2** The topology of connected tools.

# 3 Threat Analysis

The STRIDE threat analysis model is one of the most established frameworks for systematically identifying security threats in distributed and networked systems. Originally introduced by Microsoft, STRIDE categorizes threats into six classes—Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege—allowing researchers to perform structured evaluations of system vulnerabilities. Its applicability has been demonstrated across various cyber-physical systems, including IoT, cloud infrastructures, and digital twins ([4-6]). These references establish STRIDE as a robust and academically validated approach for threat identification in complex digital environments.

The STRIDE threat modeling framework is aligned with the requirements and terminology of ISO/IEC 27005 and ISO/IEC 15408, which provide standardized methodologies for information-security risk assessment and evaluation. We applied the STRIDE threat-modeling framework, and the resulting analysis is summarized in Table 1.

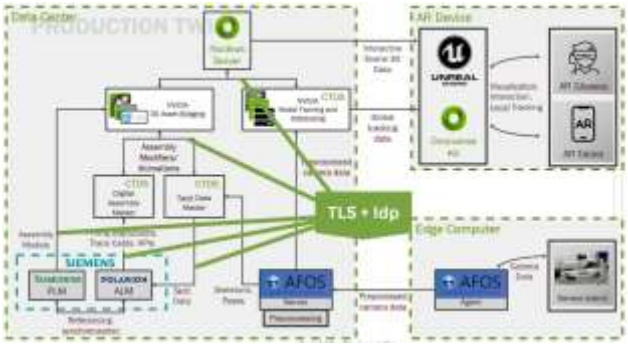
| STRIDE threat | Where it occurs | Mitigation in design |
|---|---|---|
| S-Spoofing | gRPC (AFOS aganet and Server) | mTLS with CA issued certificates |
| S-Spoofing | Nucleus, APIe, USD Connectors | idP issued JWT tokens +TLS |
| S-Spoofing | AR/WebRTC Signals | idP Authentication+DTLS |
| T-Tampering | gRPC Traffic | TLS integrity through mTTS |
| T-Tampering | HTTP (Nucleus /APIs/Connectors) | TLS 1.2 /1.3 |
| T-Tampering | WebRTC media /data | SRTP +DTLS |
| R- Repudiation | Backend Servics | Certificate Identity +logging |
| R- Repudiation | Nucleus and APIs | JWT tokens+server side logs |
| R- Repudiation | AR/WebRTC Signals | Authentication signalling logs |
| I- Information Disclosure | gRPC data | mTLS encryption |
| I- Information Disclosure | Nucleus/APIs/USD Connectors | TLS encryption |
| I- Information Disclosure | WebRTC AR Streams | DTLS +SRTP |
| D- Denial of Service | Backend, APIs/ Signals, WebRTC | Certificate and token checkreduce unauthorized traffic |
| E- Elevation of Privilage | Nucleus and APIs | idP role claims+ACL enforcement |
| E- Elevation of Privilage | Backend Ervices | mTLS identity binding |

**Table 1** STRIDE threats and mitigations.

We focused more on the communication links between the software components in our system and identified three principal communication methods as a use case: gRPC, HTTP, and WebRTC. These technologies were selected due to their widespread adoption and strong scientific foundation, each of which is well documented in existing literature. Specifically, recent studies demonstrate the efficiency and scalability of gRPC in distributed systems, the robustness and interoperability of HTTP in heterogeneous environments, and the low-latency, real-time capability of WebRTC for peer-to-peer communication [7-10]. Together, these publications provide a solid scientific basis for understanding and evaluating the communication mechanisms used in our environment.

We systematically designed mitigation measures adapted to the security characteristics of each individual communication protocol:

DOI 10.5162/iCCC2026/P31




A**. mTLS for gRPC**

Mutual TLS (mTLS) is an extension of standard Transport Layer Security in which both the client and the server authenticate each other using X.509 certificates. This bidirectional authentication ensures that only trusted components can join a distributed system, which is particularly important in microservice-based architectures using gRPC. Because gRPC is built on HTTP/2 and is commonly used for high-performance inter-service communication, securing these channels with mTLS provides confidentiality, integrity, and strong identity binding for each endpoint. The use of mTLS is widely recommended in cloud-native environments due to its proven security properties and compatibility with service meshes such as Envoy or Istio. The security foundation of mTLS is defined in the TLS 1.3 standard (RFC 8446) [7]. Its usage in our network is illustrated in Figure 3.

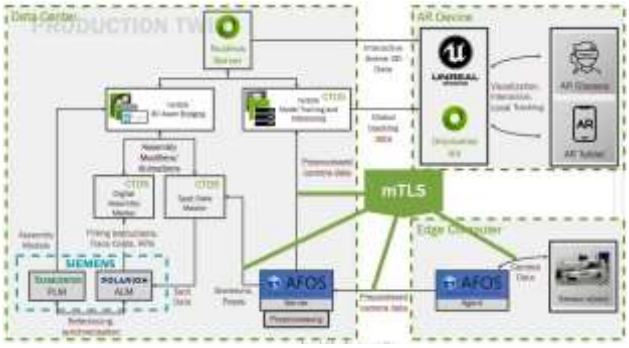



**Figure 3** mTLS for gRPC.

B. **TLS + Identity Provider (IdP) for HTTP**

For HTTP-based communication, a widely adopted security pattern combines TLS encryption with authentication and authorization managed by an external Identity Provider (IdP). TLS provides transport-layer confidentiality and integrity for data exchanged between clients and web services, whereas the IdP—commonly implementing OAuth 2.0 or OpenID Connect—issues verifiable tokens representing user or service identities. This separation of concerns enables secure scalability across distributed digital-twin platforms and enterprise service architectures. TLS 1.3 (RFC 8446) establishes the cryptographic foundation for secure transmission [7], while the identity-federation mechanisms are specified in "The OAuth 2.0 Authorization Framework" (RFC 6749) [8]. The integration of these technologies offers robust authentication, flexible authorization, and encrypted transport without requiring modifications to the HTTP application logic. The usage of this components in our network is illustrated in Figure 4.

**Figure 4** TLS + Identity Provider (IdP) for HTTP.

C. **DTLS + SRTP for WebRTC**

WebRTC real-time communication relies on Datagram Transport Layer Security (DTLS) and Secure Real-time Transport Protocol (SRTP) to protect media and data streams over peer-to-peer connections. DTLS provides TLS-equivalent security over UDP, enabling certificate-based authentication and encryption during the handshake phase. Following key exchange, SRTP encrypts and authenticates audio and video packets efficiently, minimizing latency—an essential requirement for AR/VR and interactive real-time systems. DTLS is formally specified in RFC 9147 [9], while SRTP is standardized in RFC 3711 [10]. Together, DTLS and SRTP constitute the mandatory security architecture of WebRTC, ensuring confidentiality, integrity, replay protection, and secure key negotiation for real-time multimedia communication. Figure 5 illustrates their usage in our network.

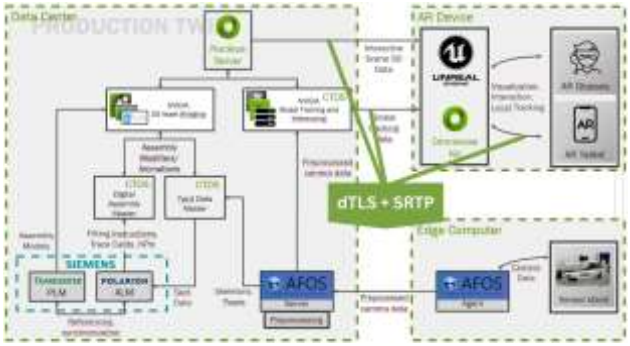



**Figure 5** DTLS + SRTP for WebRTC.

# 4 Conclusion

In this study, we examined a real-world implementation of a digital twin, particularly focusing on the component handling video information. Our analysis included the evaluation of all software tools used and their connection protocols, employing the STRIDE threat analysis methodology. From this, we compiled a comprehensive list of necessary mitigation measures applicable to this environment. The digital twin in question is actively utilized for visual information management within the production line of specific types of airplane engines.

Our findings underscore the significance and complexity of cybersecurity challenges in this domain, exacerbated by the intricate nature of the software and communication protocols involved. As these complexities continue to escalate, we assert that cybersecurity design methods require heightened attention to effectively anticipate and mitigate future threats.

Future research could expand on this work by investigating larger and more complex implementations of digital twins,

further contributing to the development of robust cybersecurity strategies in increasingly sophisticated technological environments.

## Acknowledgement

The project DIREKT and this publication were funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag (funding reference: 20L2108C1).

## 5 Reference

[1] M. Grieves and J. Vickers, “Digital Twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in Transdisciplinary Perspectives on Complex Systems, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, pp. 85–113.

[2] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” IEEE Trans. Ind. Informatics, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[3] H. Zargariasl, S. R. Khan, C. Borck, M. Behm, and C. Herglotz, “Threat analysis in real-world computer networks possessing a digital twin: Graph visualization and Markov clustering,” in Proc. IEEE Smart World Congress (SWC), Calgary, Canada, 2025.

[4] O. Saßnick, T. Rosenstatter, C. Schäfer, and S. Huber, “STRIDE-based methodologies for threat modeling of industrial control systems: A review,” in Proc. IEEE Int. Conf. on Industrial Cyber-Physical Systems (ICPS), 2024, pp. 1–8, doi: 10.1109/ICPS59941.2024.10639949.

[5] P. Das et al., “STRIDE-based cybersecurity threat modeling, risk assessment and treatment of an in-vehicle infotainment system,” Vehicles, vol. 6, no. 3, pp. 1140–1163, 2024.

[6] M. da Silva et al., “Automated ICS template for STRIDE Microsoft Threat Modeling Tool,” in Proc. 18th Int. Conf. on Availability, Reliability and Security (ARES), 2023.

[7] E. Rescorla, “The Transport Layer Security (TLS) protocol version 1.3,” RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8446.

[8] D. Hardt, “The OAuth 2.0 authorization framework,” RFC 6749, Oct. 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6749

[9] E. Rescorla, N. Modadugu, and J. Hodges, “Datagram Transport Layer Security (DTLS) version 1.3,” RFC 9147, Apr. 2022. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9147

[10] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, “The Secure Real-time Transport Protocol (SRTP),” RFC 3711, Mar. 2004. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3711