Reference Architecture of Cybersecurity Digital Twin

Taru Itäpelto[1 [0000−0001−7862−265X]], Mohammed Elhajj[1 [0000−0002−4022−9999]], Marten van Sinderen[1 [0000−0001−7118−1353]], and Maria Iacob[2 [0000−0002−4004−0117]]

[1] Semantics, Cybersecurity & Services (EEMCS-SCS)
https://www.utwente.nl/en/eemcs/scs/
[2] Section Industrial Engineering and Business Information Systems (BMS-HBE-IEBIS)
https://www.utwente.nl/en/bms/iebis/
University of Twente, PO BOX 217, 7500 AE Enschede, The Netherlands
{t.m.itapelto, m.elhajj, m.j.vansideren, m.e.iacob}@utwente.nl

Abstract. The transformation of previously isolated Critical Infrastructures (CIs) into intricate Systems-of-Systems has rendered them vulnerable to various threats. CIs are characterized by long life cycles and high availability requirements, which pose significant challenges in maintaining cybersecurity throughout their operational life cycle. Existing testing methodologies prove inadequate and may compromise the CI's operational continuity. This paper proposes to shift testing activities to a Digital Twin (DT) connected to the CI. The DT provides a digital counterpart of the real system, enabling cost-effective testing without compromising operational integrity. For this approach, we present an enterprise architecture called the cybersecurity DT reference architecture. Through a camera surveillance system use case, we demonstrate the feasibility of this reference architecture, focusing on what-if testing using DT-enabled attack simulations. We show how to enhance decision-making when evaluating system configurations and how to deploy optimized configurations to the real system.

Keywords: Digital Twin · Critical Infrastructure · cybersecurity · Enterprise Architecture · DT.

1 Introduction

In recent years, concerns regarding the cybersecurity of Critical Infrastructures (CIs) have emerged as a focal point of research and societal attention. Key sectors such as energy, water, communication, and transportation systems form the backbone of modern societies, playing a pivotal role in sustaining economic activities and ensuring the health and safety of citizens [1, 15].

Historically, CIs were characterized by physical and digital isolation [15]. However, by introducing increased connectivity and integrating advanced functionalities, such as Internet of Things (IoT) and cloud-based solutions, CIs have

ushered in a new era. IoT applications facilitate remote monitoring and control, along with intelligent analysis of big data using Artificial Intelligence/Machine Learning (AI/ML). Nevertheless, these advantages are accompanied by drawbacks, including new security threats [10,15]. Ensuring CIs' security throughout its Life Cycle (LC) requires continuous testing. Albeit, existing testing methods for CIs prove inadequate and may compromise operational continuity [15].

The Digital Twin (DT), a virtual replica of a Real System (RS), is regarded as a promising solution for enhancing the cybersecurity of a CI throughout its LC [6,7] if the challenges it introduces, such as DT's cybersecurity, are properly addressed [6]. Even though definitions of DT vary in the literature [3, 9], we adopt the one from [9]. This definition distinguishes three main components of the DT concept: the actual system of interest, called the Real System (RS); the virtual counterpart that maintains a digital copy of the RS, called the DT; and the bi-directional communication, known as twinning, that synchronizes the RS and DT.

This paper addresses these cybersecurity challenges by proposing to shift testing activities to a DT connected to the CI, focusing on enhancing both the CI's and the DT's cybersecurity. We present an Enterprise Architecture for this approach, which can function as a reference architecture (RA) for cybersecurity Digital Twins in CI domains. With the Surveillance System (SS) use case, we exemplify how the RA can support three DT-enabled smart security services: what-if testing, decision support, and optimization. DT-enabled smart services, which can be dynamically added, removed, or modified, are motivated by evolving security concerns throughout the CI's LC. SSs, widely used in CIs such as nuclear power plants, can accidentally serve as entry points for attackers due to IoT vulnerabilities, underscoring the necessity for robust security throughout their LC. For instance, unauthorized access to nuclear power plant management systems could have severe consequences. DT-enabled what-if testing allows for efficient and effective virtual testing of various configurations and responses without disrupting CI operations. This supports informed decision-making and optimization through a feedback loop between DT and RS. Our example system is relevant to CIs and illustrates how our RA can be applied to a real-world CI component, an IoT-enabled SS integrated into a nuclear power plant. To our knowledge, this is the first time a generalised RA has been introduced to enhance the security of CIs using DT.

The rest of the paper is organized as follows: We present the rationale behind the proposed RA and other background information in Section 2 and related work in Section 3. Following these, we propose the DT-CI system's RA in Section 4, illustrate its overview in Section 4.1, and detailed layered perspectives of the RS in Section 4.2 and the DT in Section 4.3. Furthermore, we integrate the example use case-specific changes to these detailed models. Additionally, we introduce the architecture of the SS-specific DT-enabled smart services, i.e. what-if testing, decision-making support, and optimization in Section 4.4. Subsequently, we address the limitations of our approach, along with potential future research avenues in Section 5. Finally, we summarize our paper in Section 6.

2 Background

CIs are essential to society's functioning, economy, and security [1, 15]. These systems are often cyber-physical systems (CPS), which consist of critical physical parts controlled and monitored by cyber parts and a network connecting these two parts [14]. Examples of physical part assets are IoT sensors, actuators, and embedded systems. However, cyber part components, such as management applications, could also be considered critical assets needing protection. The digitization of previously isolated CIs has resulted in them forming complex System-of-Systems with various (inter)dependencies with other systems and CIs. For simplicity, with the term *RS*, we refer to CI as an IoT-empowered CPS and System-of-Systems in the rest of the paper. We also consider that DT is used to monitor, analyze, and improve the security of such a system's critical assets, including the IT systems used to manage the physical part.

Ensuring the CI's cybersecurity throughout its life cycle (LC) necessitates continuous security testing and maintaining appropriate security measures. On the other hand, the unique characteristics of CIs impose specific demands on testing procedures. Firstly, the testing process must not disrupt the operation of CIs. Secondly, CIs have a long life cycle (lasting 30-40 years [4]), so it is essential to maintain and test the cybersecurity measures considering evolving threats, requirements, systems, and operating contexts throughout the entire life cycle of the CI [3, 15].

However, with current security testing methods, it is difficult to test these complex System-of-Systems and mitigate all possible current threats; it might even be unobtainable [2]. One example of such a method is pen and paper testing, which is not an adequate tool for testing even the current systems [15]. On the other hand, penetration testing might endanger the availability of the RS [15]. Using traditional, isolated testbeds is not viable since keeping these up-to-date is difficult and costly. Furthermore, testbeds often have different settings and functionalities, which makes them different from the RS, and test results are possibly inaccurate [15].

One potential suggested solution to address these challenges is DT, an evolving mirror of RS empowering a thorough understanding of the changing RS, its dependencies, and operating context [6]. It is essential to remain attentive concerning emerging threats from evolving operating conditions and the complex network of systems and dependencies. On the other hand, our proposed DT-CI integration facilitates the selection and deployment of effective and efficient countermeasures. Effective countermeasures address all relevant threats, while efficient ones have the least redundancy and minimal impact on the CI's operation.

Because of the pivotal role of CIs, it is important to get a grip on the proper alignment between stakeholders' concerns and goals on the one hand and the technical design and implementation of the CI on the other. Enterprise Architecture [11] is a proper conceptual tool for this. An Enterprise Architecture provides an integrated view of an enterprise system in an enterprise context, which helps to identify and maintain the mentioned alignment during the LC

of the system. A widely accepted framework and modelling language for Enterprise Architecture is ArchiMate[3]. ArchiMate also provides useful structures and patterns, like security overlay [13], which we utilize to model the security requirements of the proposed architecture of DT-enhanced CI discussed in Section 4.5.

3 Related work

To the best of our knowledge, only a few studies have been conducted integrating DT and Enterprise Architecture to improve a system's cybersecurity. Sellitto and Masi et al. [12,17] introduce a cybersecurity perspective, an extension viewpoint integrated into the existing architecture of a RS. This viewpoint was employed to identify the necessary countermeasures for elevating the modelled RS's security level to a predetermined standard. In their initial study [17], they proposed utilizing Enterprise Architecture models, while their subsequent work [12] allowed the incorporation of any architecture models. By iterating the process of attack simulation and incorporating feedback into architecture models, they successfully devised a RS architecture with a cost-effective set of countermeasures capable of mitigating the specified security threats to an acceptable degree.

Masi et al. [12] referred to their solution as a Digital Shadow due to its ability to run simulations and modify architecture models, but lacking automatic data exchange between the Operational Remote System (ORS) and its DT. In contrast, we consider a DT-enhanced approach, incorporating bidirectional, real-time twinning between the operational RS and its DT. The DT utilizes real-time data collected from the RS to realise its services. The feedback loop in the other direction enables DT to deploy the chosen countermeasures to RS to optimize it and improve its cybersecurity.

4 Proposed Solution

This section will present our proposed RA for a DT-CI system and exemplify its application to a nuclear power plant's SS and three DT-enabled smart security services.

We start with the assumption that DT's fidelity is sufficient, i.e., that DT accurately represents the RS and its behaviour, and that continuous monitoring and testing of the DT are conducted to maintain this accuracy.

Within the scope of this RA, we made some modelling choices, such as integrating the general reference and the specific application example architectures, modelling each component only once, modelling only publish-subscribe paradigm-based communication, and omitting separate, additional sensors and actuators possibly deployed to RS to enable DT-based specific smart services. These choices allow us to simplify our models and avoid redundancy.

[3] https://www.opengroup.org/archimate-forum/archimate-overview

In all of our models, we have used the darkness of the colours to distinguish elements belonging only to the general reference (darkest green), both the general reference and the example SS (lighter green/blue/yellow), or only to the example SS (lightest green/blue/yellow) architectures. For example, in *On-site* tier of *RS* in Figure 2, the lighest colour components (e.g. *Camers(s)*) belong only to the SS, the darkest green colour components (e.g. *Actuators*) only to the general reference and others (e.g. *Log(s)*) to both architectures.

The following models[4] will be discussed in detail in subsequent sections: Overview of the DT-CI system's RA in Section 4.1, *RS* model in Section 4.2, *DT* model in Section 4.3, *Smart Services* model in Section 4.4, and Security model in Section 4.5.

4.1 Overview of the Reference Architecture (RA)

In this section, we will provide an overview of our proposed DT-CI system's RA, comprising three core components: the *RS*, *DT*, and the use case specific *Smart Services* as illustrated in Figure 1. The overview architecture encompasses three ArchiMate layers: the business layer with yellow, the application with blue and the technology layer with green elements.

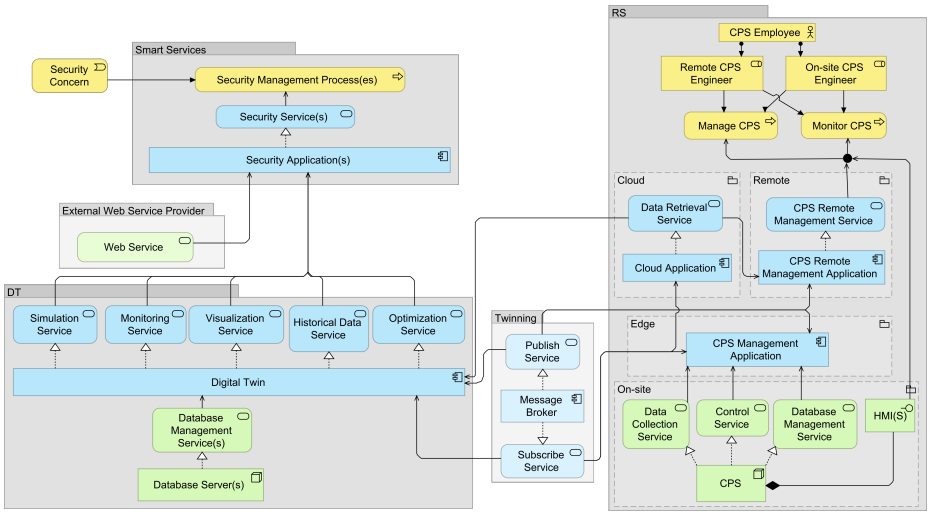

Fig. 1. Total view of the proposed solution

The *RS* embraces the *On-site*, *Edge*, *Cloud* and *Remote* tiers. It can be monitored and controlled on-site or remotely by *On-site/Remote CPS Engineer*s. We discuss the detailed model of this cloud-edge-based *RS* in Section 4.2.

[4] High-resolution images of the models available in: https://gitlab.utwente.nl/itapelto/whattwin

The *DT* has *Simulation*, *Monitoring*, *Visualization*, *Historical Data* and *Optimization* services, which the example use case specific *Smart Services*, i.e. *Security Application(s)* build on top of DT's services and offering *Security Services* to *Security Management Process(es)*. We will elaborate on the *DT* in Section 4.3 and the example use case specific *Smart services* in Section 4.4.

One of the key functionalities of the proposed DT-CI system is the bidirectional communication between *DT* and *RS*, i.e. *Twinning*. The twins' ability to publish and subscribe messages to/from the other twin facilitates seamless mapping and synchronization between digital and physical realms. For clarity, we chose this widely used communication paradigm in IoT and cloud-based applications. This choice does not affect the modelling of RA but should be replaced with the DT-CI-specific communication architecture when applying our RA to a specific RS. *DT* subscribes to data messages published by *RS* and *RS* subscribes to control/optimization messages published by *DT*.

Following the introduction of the high-level architecture model, we will provide in-depth, layered insights into the two main components of the DT-CI system and the example use case specific smart security services in the subsequent sections 4.2-4.4. Additionally, since DT increases the attack surface [6], paying attention to the fundamental communication-related security features is essential when applying our DT-CI system's RA in a practical system. To simplify our models, we elaborate security functionalities in a separate security model in Section 4.5 instead of including them in this overview and the detailed *DT*, *RS* and example application use case specific *Smart services* models.

4.2 Real System (RS)

As discussed, we have modelled the *RS* as an IoT and cloud-based system, with *On-site*, *Edge*, *Cloud* and *Remote* tiers illustrated in Figure 2. We emphasize that depending on the specific RS and desired DT-enabled smart security services and their requirements, collecting the required data and applying the changes based on *DT*'s feedback might require integration of additional physical and/or cyber components and instrumenting the RSs' cyber components to support these functionalities. As mentioned in Section 4, we decided not to model any such additional components. Still, to highlight the importance of such components, we modelled one sensor relevant to our example application use case, i.e. SS camera's *Angle Sensor(s)*.

To exemplify the application of our RA to a specific CI, we have illustrated *Nuclear Power Plant*'s SS-specific architecture components with light colours (yellow/blue/green/grey) in Figure 2. Even when a SS is integrated as a security measure, its components must be secured throughout their LC [7, 16]. Although the SS might be secure against current threats, ongoing updates, additions, or removals of physical, cyber, or hybrid components could introduce new vulnerabilities exploitable by unknown threats.

As a model of a CPS, the *RS*'s *On-site* component consists of physical and cyber (application) layers. In the physical layer (depicted in green in Figure 2), each *Actuator*, *Sensor* and *Embedded System* provide a low-level *Data Collection*

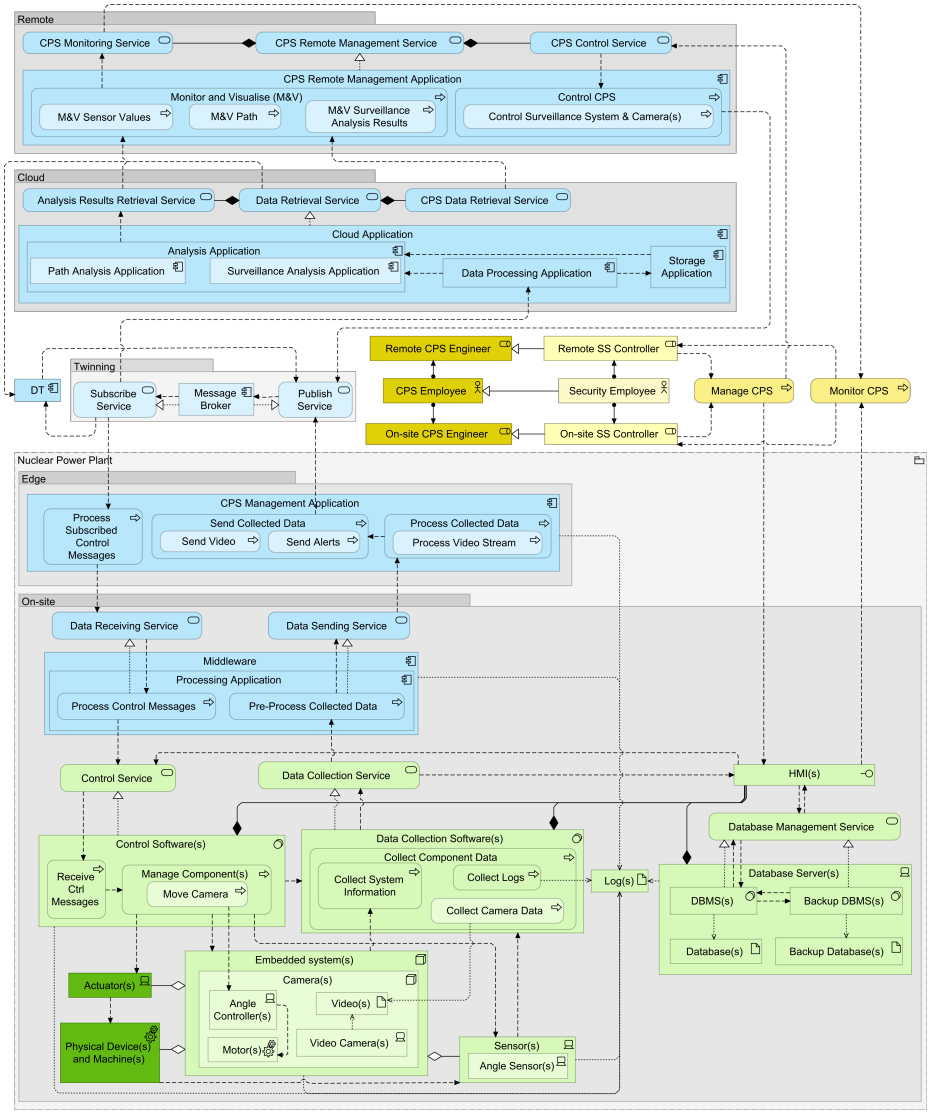

Fig. 2. Layered View of the Real System

Software to retrieve/collect their data and logs, and *Control Software* to control them. *Sensor(s)* monitor the *Physical Device(s) and Machine(s)* and their operational context, while *Actuator(s)* facilitate actions on these. *Embedded systems* serve as integrated platforms for managing all these components.

The control options could include a possibility to *Manage Component(s)*, i.e. to re-configure, re-calibrate, update the component or install new software to them. Collected data could include *Logs* from all *RS* components and *System*

information, such as file system, system inventory and general system information acquired, for example, by periodically executing system commands.

Within the realm of data management, *Databases* accessible through *Database Management Service(s)* realized by database management systems, i.e. *DBMS(s)* play a crucial role. They store collected data, encompassing a wide range of information such as sensor readings, system details, and behavioural patterns. These databases are not only repositories but also essential for *RS* recovery. *Backup databases* are utilized to retrieve data and restore the *RS* with recovered data. As SS-specific components (lightest green elements in Figure 2), we have modelled the *Camera(s)* as embedded systems consisting of *Motor(s)*, their *Angle Controller(s)* and *Video Camera(s)* capable to shooting surveillance *Video(s)* and storing them locally, e.g. to a memory card. Each *Camera*'s *Control Software* allows changing the camera angle (*Move Camera*) via *Motor(s)*. *Data Collection Software* can collect (*Collect Camera Data*) the stored *Video(s)* and store to the on-site database using *Database Management Service*. An *Angle Sensor(s)* responsible for monitoring the surveillance camera's angle were integrated as evidence of the RA supporting additional components required to implement DT-enabled smart services. *DT* or *Data Collection Software* could be instrumented to collect its data to detect attacks targeting SS.

The components illustrated with the darkest green colour in Figure 2 represent components commonly used in *CPS*s but which are not used in the SS.

The nature of the collected data is highly adaptable and tailored to the specific needs of the *RS* and its distinct use cases. This data includes details about sensor readings, system states, components details, topology, hardware and software specifics, processes, configurations, network flows, control commands, logs, etc. It could also include information collected by possible additional DT-enabled smart service-specific components integrated into the *RS*.

Our RA allows *CPS Employee* playing the role of *Remote/On-site CPS Engineer* to *Manage/Monitor CPS*, its the *On-site* infrastructure directly through *HMI(s)* or using *RS*'s internal communication channels. For simplicity, we have also utilized the *Twinning* for this purpose. When applying the RA to a specific *RS*, its internal communication channels should be specified.

The *Middleware* application acts as a central control hub to manage the collected *RS* data and *Process Control Messages* from *CPS Management Application*. It is responsible for *Pre-Process Collect Data* process, including functionalities such as formatting and timestamping the collected *On-site* real-time and historical data before sending it into the *CPS Management Application* on the *Edge*.

The *Process Collected Data* process of *CPS Management Application* at the *Edge* is in charge of processing data before the *Send Collected Data* process sends it to the *Cloud* and *DT*. Some examples of processing functionalities could include data filtering, analyzing, aggregating, and transformation. We opted for edge-enhanced data processing to reduce network and computation load on the *DT* and *Cloud* as proposed by [5]. Specific data, like detailed *RS* state and behaviour information, is exclusively targeted for the *DT*'s high-fidelity simula-

tions. Albeit real-time data could directly reach the *CPS Remote Management Application*, we introduced a *Cloud* intermediary. This *Cloud* stores, processes, and analyzes the data before the *CPS Remote Management Application* accesses it. Although this approach might introduce some delay due to the additional layer, it avoids redundant services in the *CPS Remote Management Application*. In time-sensitive scenarios, direct end-to-end communication channels between the *Middleware* (or even *Data Collection Software*) and the *CPS Remote Management* applications could be an alternative solution, considering the impact of each layer on data transfer throughput. The SS-specific components in the *Edge* consist of *Process Video Stream* and *Send Video/Alerts* processes. The former is capable of quick, less resource-demanding but coarser anomaly detection than the *Cloud*, and the latter sends video and possible alerts of detected anomalies to the *Cloud*.

The *Cloud* is a versatile data management and analysis platform. Its *Data Processing Application* filters, cleans, transforms, and aggregates the subscribed data before *Storage Application* stores it in cloud storage and *Analysis Application* analyzes it. Additionally, the Cloud's *Analysis Application* employs advanced techniques, such as ML/AI, and Big Data, spatial, temporal, and statistical analysis methods to analyze the received data. In our example use case, the intruder's physical path could be analyzed by *Path Analysis Application*, and *Surveillance Analysis Application* could detect potential anomalies, such as intrusions. In essence, the *Cloud* stores data and provides sophisticated analytical capabilities, making it a powerful tool for processing real-time and historical data.

The final tier, *Remote* in our *RS* model includes the *CPS Remote Management Application*, empowering engineers to *Monitor and Visualize (M&V)* and *Control CPS* remotely. Leveraging data processed, stored, analyzed, and served by the *Cloud*, this application enhances the detection of significant events, such as alarms, through visualization. Examples of SS's monitored and visualized data and analysis results include *Surveillance Analysis Results*, *Sensor Values*, and intruder's physical *Path*. The *Remote SS Controller* and *On-site SS Controllers*, the general reference architecture *Remote CPS Engineer* and *On-site CPS Engineer* roles' specializations, may utilize the *Control Surveillance System & Camera(s)* process through the *CPS Control Service* to manage the SS or to move cameras, i.e. change the camera angles.

4.3 Digital Twin (DT)

Figure 3 illustrates the *DT*'s architectural structure, encompassing both application and technology layers. Applying this part of the RA to the example SS does not introduce any changes.

For *DT* to function effectively, it necessitates data and models to create a high-fidelity simulation environment and support the functionalities outlined in the model. The *DT* as a virtual replica of *RS* consists of various models of *RS*, its behaviour, and context. Moreover, *DT* functionalities might necessitate creating and maintaining other models, like ML/AI and attack models used by *Predict*,

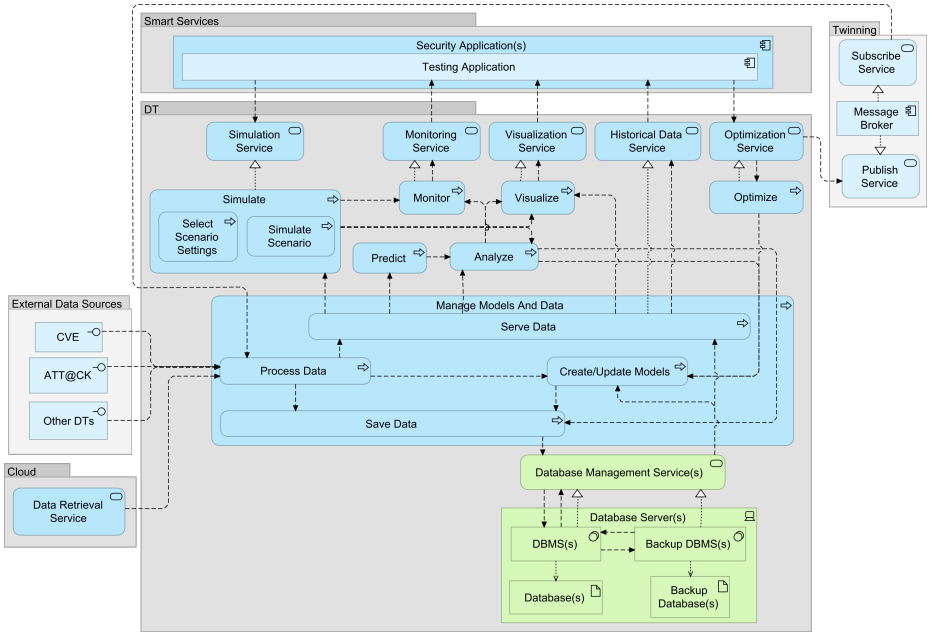

Fig. 3. Layered view for DT

Simulate and *Analyze* processes. The necessary data can be generated by the various *RS* tiers, *Smart Services* or *External Data Sources*. We discussed the highly adaptable *RS* data in Section 4.2 obtained through *Twinning* or retrieved from *Cloud*. External data sources may include databases like Mitre ATT@CK[5] that offer attack tactics and techniques and Mitre CVE[6] providing information on common vulnerabilities. Furthermore, DT might incorporate information from other DTs within (inter)connected and (inter)dependent systems, enriching its data pool.

In our example, the third data source, SS-specific *Testing Application* (see Section 4.4) as an example *Smart Service*, includes data related to test scenarios, such as simulation and scenario options and deployed optimizations.

DT provides five services to other applications, like to *Smart Services*: *Simulate*, *Monitor*, *Visualize*, *Optimize*, and *Historical Data Service*. These services are enabled by the corresponding and underlying *Analyze*, *Predict*, and *Manage models and data* processes.

The *Manage Models And Data* process involves various sub-processes, including *Process, Save* and *Serve Data*, and *Create/Update Models*. Initially, received data, *DT* models, and analysis results are processed, stored, and managed in databases in a similar way as in *RS*'s *Cloud* and *On-site*. This data can be accessed by the *Create/Update Models* and *Serve Data* processes. The

[5] https://attack.mitre.org/resources/working-with-attack/
[6] https://cve.mitre.org/

Create/Update Models process may create or update system or attack models or knowledge graphs and train or retrain ML/AI Models using both real-time and historical data obtained from *Process Data* process and *Database Server(s)*. Used data could include results from *Analyze* process, and knowledge graphs can be used to visualize system state and security-related knowledge.

As discussed in Section 4, we assume that the *Create/Update Models* process maintains and updates the required *DT*'s models throughout *RS*'s LC to ensure fidelity.

The *Serve Data* process delivers diverse information to *DT*'s other processes. The served data includes attack, ML/AI, system state and behaviour models, knowledge graphs, real-time and historical system and analysis results data.

The *DT*'s *Analyze* process examines real-time and historical data using various methods and creates knowledge graphs representing knowledge models. Besides, this process incorporates predictions from the *Predict* process, which utilizes ML/AI models and real-time data to predict future system states, behaviours, or threats. The analysis results could be used by the *Monitor*, *Visualize*, and *Create/Update Models* processes. Monitoring and visualizing knowledge graphs and possible safety and security rule violations facilitate testing engineers' reasoning on the *RS* and support knowledge-based decision-making [8]. Also, such results could be used to update models and knowledge graphs.

The *Simulate* process allows virtual exploration and experimentation of the *RS* and its behaviour under different scenarios through *Select Scenario Settings* process. *Simulate Scenario* process uses various *RS* models, including system state, system behaviour and attack models, along with replicated real-time and historical *RS* data. Consecutive attack simulations allow for comparing the effectiveness and efficiency of different countermeasures against the tested attacks utilizing the historical system data and analysis results. The outcomes of these simulations are analyzed, monitored, and visualized to allow tracking of simulation, system states, behaviour, and security metrics.

DT also enables optimizing *RS* through its *Optimize* process, which implements the requested modifications to the *RS* and to the corresponding *DT* models to maintain their fidelity using *Create/Update Models* process. These optimizations can involve adjustments to *RS*'s components and their arrangements, configurations, safety protocols, security rules, or other relevant parameters.

After introducing our detailed DT model, we proceed to the example nuclear power plant's SS-specific *Smart Services* model, a model of a *Testing Application* enabling what-if testing, decision-making support and optimization of the *RS*.

4.4 Smart Services Example: Testing Application

To ensure the security of our example application use case - a nuclear power plant's SS throughout its LC - we illustrate a *Testing Application* as a *Smart Services* triggered by *Security Concern*s and built on top of DT's functionalities in Figure 4. *Testing application* allows the testing engineers to perform DT-enabled simulations, i.e. *What-if Testing* of various configurations, test scenarios and countermeasures, such as existing or planned countermeasures against

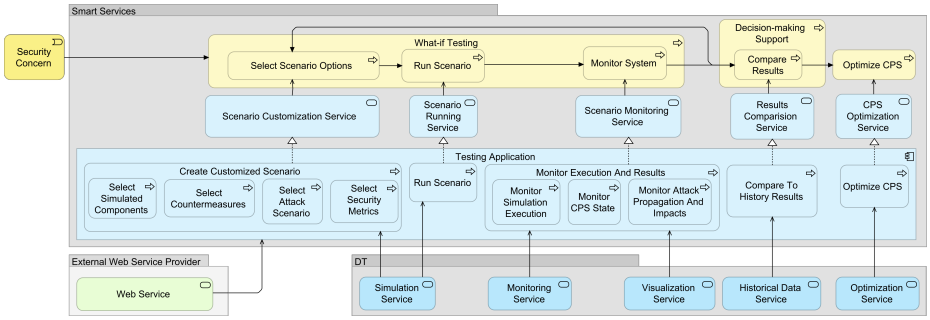

Fig. 4. Testing Application Model

possibly evolving or new threats, to gain insights into their efficiency, effectiveness, and impact. DT can also visualize these simulations and insights along with historical data, facilitating comparisons (*Compare Results*) and gaining *Decision-making support* regarding required and the most efficient and effective actions/options, which then can be deployed to the *RS* using *DT*'s *Optimization Service*.

The initial phase of *What-if Testing* facilitated by *Testing Application* involves configuring the simulation scenario through *Create Customized Scenario* process comprising processes such as *Select Simulated Components*, tested *Countermeasures*, *Attack Scenario* parameters, and pertinent *Security Metrics*. Given the complexity of System-of-Systems, one, several, or all *RS*'s components can be simulated. The *Select Attack Scenario* consists of attacker properties definition, which could include defining the attacker's goals, target vulnerabilities, and attack methods, such as attack tactics, techniques and procedures. The final step of *Create Customized Scenario* is to *Select Security Metrics* from the *DT*'s predefined set, validated by CPS security experts. This step-by-step approach ensures a thorough and organized simulation process, enabling detailed testing of various System-of-Systems scenarios and variables. Customizable parameters and metrics are crucial for a comprehensive security evaluation.

After initialization, the *Run Scenario* process initiates the simulation execution. It allows security engineers to *Monitor Execution And Results*, including *Monitoring Simulation Execution/CPS State/Attack Propagation And Impacts*. *Visualization* of the monitored information, such as CPS or attack parameter values and security metrics, is a crucial functionality enabled by DT, aiding in detecting alerts and extracting key insights from extensive data. The *Compare to History Results* process facilitates *Decision-making Support* by allowing to *Compare Results* of current and historical simulations, helping determine the most effective and efficient set of countermeasures. Finally, testing engineers can decide to start another simulation with other options or *Optimize CPS* by optimizing its configurations or updating safety&security rules or countermeasures.

We have represented the *Testing Application* as an externally provided *Web Service*. If it is self-hosted, this component requires more detailed modelling.

As outlined in this section, the example application use case model completes our detailed examination of the RA components. The upcoming model will explore security mechanisms encompassing the entire RA.

4.5 Security Overlay

The security goals, controls, and principles outlined in our security model illustrated in Figure 5 should be implemented across the entire RA, including all its components and layers. While DT can enhance the cybersecurity of mirrored sys-

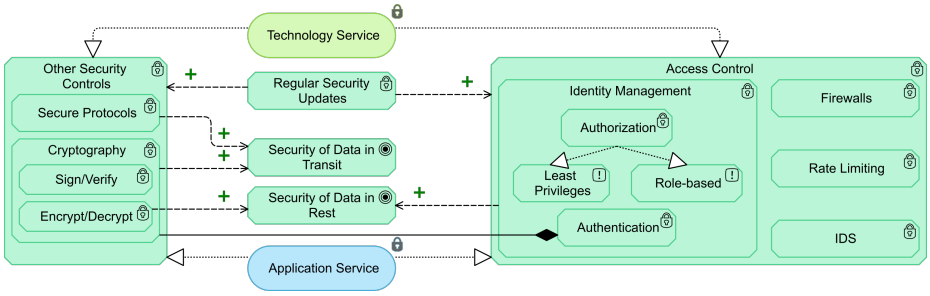

Fig. 5. Security view

tems, it also expands attack surface [6]. Unauthorized access to DT's data could provide valuable insights to attackers on targeting the RS without detection, using techniques like zero-day attacks or Advanced Persistent Threats (APTs). This includes gaining access to confidential information, details about the RS's most vulnerable components, countermeasures, and valuable assets [6]. Exploiting access to the DT, attackers could control or manipulate the RS, causing disruption or malfunction to RS's operation [15].

Ensuring *Security of Data in Rest* entails *Encrypt*ing all the stored data to ensure privacy and confidentiality, while *Security of Data in Transit* can be protected using *Secure Protocols* and cryptographic algorithms to *Encrypt* and/or digitally *Sign* all communications between different layers to ensure integrity. Furthermore, services accessible through networks should be guarded using *Firewalls*, *Rate Limiting* techniques, and Intrusion Detection Systems (*IDS*). All incoming messages must undergo *Authentication*, and subsequent actions should adhere to proper *Authorization*, following principles such as *Least Privileges* and *Role-based Access Control*. To maintain security throughout the extended LC of CIs, performing *Regular Security Updates* is essential across all RA components.

5 Discussion

Although the suggested approach is a generalization, we believe that it could be extended and adapted to match any RS aiming to integrate DT-enabled services.

We have simplified the RA to offer a broad overview of the system. It may be necessary to introduce additional abstraction levels or break down models into smaller, more specific sub-models. For instance, details such as data collection methods, periods and processing, DT functionalities, detailed analysis methods, required fidelity and granularity, security requirements, data synchronization, and other relevant factors should be further specified to align with the specific use case and its expected outcomes or requirements.

Ensuring synchronization is important, especially in CIs, which include multiple heterogeneous systems, each possibly equipped with its DTs. These diverse DTs may depend on each other's services, requiring synchronization of data collected from various systems and sources to generate accurate simulation and prediction models. Moreover, data should undergo maximal processing at the edge to reduce the network load and enhance the DT's processing speed. The use case specifications, such as latency, fidelity, granularity, inputs, and expected outputs, determine the boundaries for edge data processing capabilities.

These blueprints can provide valuable insights into the essential aspects that need consideration when designing a new system. When extending an existing system with DT-enhanced services, RA can be employed to develop migration and implementation plans for transitioning from the existing system to the target system.

The proposed RA is designed with modularity in mind, enabling the changing/reusing of individual components without disrupting the entire system. However, when modifying the physical system, corresponding adjustments in its DT are imperative. The DT-enabled *Smart Service*s stand out as the most adaptable module for modification, as shown with what-if testing scenario.

While one might argue that our RA modelling should have initially centered around the essential smart services of security monitoring and attack detection, our ongoing systematic literature review highlighted a notable gap: existing research [2, 3, 15] has extensively explored these aspects. Additionally, our review identified relevant studies on 'what-if testing' discussed in Section 2. Unlike our approach, these works [12,17] focus on identifying a set of countermeasures based on architectural models to achieve an acceptable risk level instead of integrating real-time data from the RS.

6 Conclusions

Securing CIs has grown increasingly complex due to the evolving nature of these systems [15]. Contemporary CIs have transformed into complex System-of-Systems, seamlessly integrating the IoT with numerous interconnections and dependencies [10,15]. These changes have notably expanded the potential attack points within these systems, effectively enlarging their attack surface [15].

Given their essential role and the potential for significant economic and societal repercussions in the event of a successful attack [1], CIs have become prime targets for malicious actors, including those with substantial resources and capabilities at the governmental level. However, testing these systems has become

increasingly challenging due to the high costs and complexity of building and sustaining high-fidelity test environments [15]. Additionally, traditional penetration testing methods pose a risk to the operation and availability of these systems.

This paper proposed a novel RA for designing a new cybersecurity DT-enhanced CI and extending an existing CI with a DT. Our proposed RA is novel in considering CI as an IoT and cloud-based complex System-of-Systems and modelling DT-enabled what-if testing, decision-support, and optimization security use cases. Moreover, we have exemplified how these proposed architecture models could be applied to a real-world use case, namely a camera surveillance system commonly used in various systems, like in a nuclear power plant. We illustrated how DT's services could support three smart security services to improve SS's security: what-if testing, decision-making support, and optimization. The main idea behind the what-if testing was to identify and decide on effective and efficient countermeasures to mitigate existing or emerging risks during the LC of RS. By incorporating the outcomes back into the architectural model, i.e. optimizing the RS, we make necessary modifications to maintain an acceptable risk level and promptly act on emerging threats. This approach showcased the effectiveness of utilizing a DT, enabling tasks like vulnerability assessment and security testing without disrupting the operational system.

In the future, we plan to extend our RA to cover other smart security services, such as security monitoring and attack detection and we intend to apply the RA to an operational CI for testing and development purposes.

References

[1] CISA Cybersecurity & Infrastructure Security Agency: Critical infrastructure sectors, https://www.cisa.gov/critical-infrastructure-sectors, last accessed 2024/08/08

[2] De Benedictis, A., Esposito, C., Somma, A.: Toward the adoption of secure cyber digital twins to enhance cyber-physical systems security. In: Vallecillo, A., Visser, J., Pérez-Castillo, R. (eds.) Quality of Information and Communications Technology. pp. 307–321. Springer International Publishing, Cham (2022)

[3] Dietz, M., Hageman, L., von Hornung, C., Pernul, G.: Employing digital twins for security-by-design system testing. In: Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems. p. 97–106. Sat-CPS '22, Association for Computing Machinery, New York, NY, USA (2022)

[4] Hallmans, D., Sandström, K., Larsson, S., Nolte, T.: Challenges in providing sustainable analytic of system of systems with long life time. In: 2021 16th International Conference of System of Systems Engineering (SoSE). pp. 69–74 (2021). https://doi.org/10.1109/SOSE52739.2021.9497465

[5] Han, Q., Zhang, J., Ding, H., Sun, J., Zhang, H., Yuan, D.: Cloud-edge collaborative-based digital twin system for hardware limited IIoT scenario. In: 2023 IEEE Smart World Congress (SWC). pp. 1–8 (2023). https://doi.org/10.1109/SWC57546.2023.10448579

[6] Holmes, D., Papathanasaki, M., Maglaras, L., Ferrag, M.A., Nepal, S., Janicke, H.: Digital twins and cyber security – solution or challenge? In: 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks

and Social Media Conference (SEEDA-CECNSM). pp. 1–8 (2021). https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566277

[7] Itäpelto, T.: Digital twin enhanced critical infrastructure life cycle security. In: 2023 IEEE Smart World Congress (SWC). pp. 1–3 (2023). https://doi.org/10.1109/SWC57546.2023.10448804

[8] Jia, Y., Gu, Z., Li, A., Han, W.: Introduction to the MDATA model. In: Jia, Y., Gu, Z., Li, A. (eds.) MDATA: A New Knowledge Representation Model: Theory, Methods and Applications, pp. 1–18. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-71590-8_1, https://doi.org/10.1007/978-3-030-71590-8_1

[9] Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W.: Digital twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine **51**(11), 1016–1022 (2018). https://doi.org/https://doi.org/10.1016/j.ifacol.2018.08.474

[10] Lampropoulos, G., Siakas, K.: Enhancing and securing cyber-physical systems and industry 4.0 through digital twins: A critical review. Journal of Software: Evolution and Process **35**(7), e2494 (2023). https://doi.org/https://doi.org/10.1002/smr.2494

[11] Lankhorst, M.: Introduction to Enterprise Architecture, pp. 1–11. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01310-2_1

[12] Masi, M., Sellitto, G.P., Aranha, H., Pavleska, T.: Securing critical infrastructures with a cybersecurity digital twin. Softw. Syst. Model. **22**(2), 689–707 (Jan 2023). https://doi.org/10.1007/s10270-022-01075-0

[13] Mayer, N., Feltus, C.: Evaluation of the risk and security overlay of archimate to model information system security risks. In: 2017 IEEE 21st International Enterprise Distributed Object Computing Workshop (EDOCW). pp. 106–116 (2017). https://doi.org/10.1109/EDOCW.2017.30

[14] Noor, M.M., Selamat, A., Husain, N.A., Krejcar, O.: Security and safety in cyber-physical system (cps): An inclusive threat model. Journal of Advanced Research in Applied Sciences and Engineering Technology **40**(2), 176–202 (Feb 2024). https://doi.org/10.37934/araset.40.2.176202

[15] Patzer, F., Meshram, A., Birnstill, P., Haas, C., Beyerer, J.: Towards computer-aided security life cycle management for critical industrial control systems. In: Luiijf, E., Žutautaitė, I., Hämmerli, B.M. (eds.) Critical Information Infrastructures Security. pp. 45–56. Springer International Publishing, Cham (2019)

[16] Pawlicka, A., Puchalski, D., Pawlicki, M., Kozik, R., Choraś, M.: How to secure the iot-based surveillance systems in an elegant way. In: 2023 IEEE International Conference on Cyber Security and Resilience (CSR). pp. 636–640 (2023). https://doi.org/10.1109/CSR57506.2023.10224938

[17] Sellitto, G.P., Masi, M., Pavleska, T., Aranha, H.: A Cyber Security Digital Twin for Critical Infrastructure Protection: The Intelligent Transport System Use Case. In: Serral, E., Stirna, J., Ralyté, J., Grabis, J. (eds.) The Practice of Enterprise Modeling. pp. 230–244. Lecture Notes in Business Information Processing, Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-91279-6_16