UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science, #UniTartuCS
Software Engineering Curriculum

Ashfaq Hussain Ahmed

Harnessing Blockchain and Digital Twin for Security Risk Assessment in Internet of Vehicles

Master's Thesis (30 ECTS)

Supervisors: Mubashar Iqbal, PhD
Sabah Suhail, PhD

Tartu 2023

Harnessing Blockchain and Digital Twin for Security Risk Assessment in Internet of Vehicles

Abstract:

Digital Twins (DTs) are digital representations of their physical counterparts, used for testing, simulation, and prototyping before a physical component is manufactured. DT technology has been widely used in smart manufacturing for several years to improve every area of operations. Moreover, blockchain technology and DTs for industrial systems have gained traction among researchers in the last decade. On the other hand, with recent advancements in smart cities and Intelligent Transport Systems (ITS), the Internet of Vehicles (IoV) concept is becoming more prevalent. DT, which serves as a metric for creating virtual representations of real-world objects, offers enormous potential for enhancing the security and resilience of all parts of smart cities, including IoV, before being deployed in the real world. Very little research exists in the context of blockchain-based DTs as a security enhancer in IoV. In this thesis, we present a comprehensive systematic review of the existing literature where blockchain-based DTs have been suggested as extra security layers for industrial systems. Moreover, a novel blockchain-based security framework is proposed to handle security threats in IoV. Additionally, security threat modeling and risk management for IoV and how the proposed blockchain-based framework mitigates the risks are showcased. The proposed solution is evaluated using DT-based simulation for V2S communication as a use case with the help of the Microsoft Azure Digital Twin platform.

Keywords: Digital Twins, Security Risk Assessment, Blockchain, Internet of Vehicles, Threat Modeling, Security Countermeasures

CERCS: P170, Computer Science, Numerical Analysis, Systems, Control

Blockchaini ja digitaalse kaksiku kasutamine turvariskide hindamiseks sõidukite Internetis

Lühikokkuvõte:
Digitaalne kaksik (inglise keeles Digital Twins ehk DTs) on virtuaalne representatsioon füüsilistest objektidest, mida kasutatakse katsetamiseks, simulatsiooniks ja prototüüpimiseks enne vastava füüsilise komponendi valmistamist. Digitaalse kaksiku tehnoloogiat on laialdaselt kasutatud nutika tootmise valdkonnas mitu aastat, et täiustada kõiki selle tegevusvaldkondi. Veelgi enam, viimase kümnendi jooksul on plokiahela tehnoloogia ja digitaalsed kaksikud tööstussüsteemides muutunud teadlaste seas üha populaarsemaks. Tänu hiljutistele edusammudele targa linna ja intelligentse transpordisüsteemi (ITS) valdkonnas on sõidukite interneti (IoV) kontseptsioon muutumas üha levinumaks. Digitaalne kaksik ehk virtuaalne koopia füüsilisest maailmast pakub tohutut potentsiaali selleks, et tõsta tarkade linnade kõigis osades turvalisust ja vastupidavust, hõlmates seejuures sõidukite interneti turvalisust ja vastupidavust, enne selle kasutuselevõttu reaalses maailmas. Plokiahela tehnoloogial põhinevate digitaalsete kaksikute kasutamist sõidukite interneti turvalisuse suurendamisel on seni vähe uuritud. Käesolevas magistritöös esitatakse põhjalik süstemaatiline ülevaade olemasolevast kirjandusest, kus plokiahelal põhinevaid digitaalseid kaksikuid on soovitatud kasutada tööstussüsteemide täiendavateks turvakihtideks. Lisaks pakutakse sõidukite interneti turvaohtude käsitlemiseks välja uudne plokiahelal põhinev turberaamistik. Samuti tutvustatakse sõidukite interneti turvaohtude modelleerimist ja riskijuhtimist ning selgitatakse, kuidas pakutud plokiahelapõhine raamistik maandab sõidukite interneti riske. Riskide maandamiseks pakutud lahendust hinnatakse V2S-side (Vehicle-to-Sensor) jaoks digitaalsel kaksikul põhineva simulatsiooni abil. Kasutusjuhtumi simuleerimiseks kasutati selles magistritöös Microsoft Azure Digital Twin platvormi.

Võtmesõnad: digitaalsed kaksikud, turvariski hindamine, plokiahel, sõidukite internet, turvaohtude modelleerimine, turvameetmed.

CERCS: P170 Arvutiteadus, arvanalüüs, süsteemid, kontroll

Contents

List of Figures

List of Tables

1 Introduction

Digital Twins (DTs) are virtual replicas of physical objects capable of simulating their real-world counterparts' behaviors, attributes, and conditions in a digital environment. The development of traditional DTs necessitates a reliable intermediary for analytical tasks and data storage. However, blockchain technology boasts distinctive attributes like decentralization, immutability, interoperability, distributed ledgers, and smart contracts. This makes blockchain integration an appealing prospect for enhancing the dependability and credibility of DTs. Consequently, blockchain-powered DTs have garnered significant interest among researchers and are actively employed in diverse industrial domains due to their heightened security and efficiency compared to conventional DTs that rely on centralized structures [68].

The practical applications of DTs have been effectively deployed across various industries, revolutionizing their operational dynamics. By incorporating DT technology, industries ranging from manufacturing to healthcare, facilities management to product innovation, have experienced improved operational processes, innovative employee training methodologies, and advanced product testing practices. Following this, the integration of DT can play a crucial role in the Internet of Vehicles (IoV) context by providing a virtual representation of real-world vehicles and their interactions within the connected ecosystem. This enables continuous monitoring, analysis, and optimization of vehicle behavior, performance, and safety. By mirroring physical vehicles in a digital environment, DTs can enhance decision-making processes, facilitate predictive maintenance, and enable real-time monitoring of vehicle conditions [36]. This is especially valuable in the IoV, where data-driven insights from DTs can contribute to improved traffic management, enhanced road safety, and the overall efficiency of vehicle operations. IoV constitutes a network where vehicles interact with their surroundings through smart communication devices such as sensors, personal gadgets, and roadside units (RSUs). This connectivity empowers vehicles with constant internet access, creating a network that generates data for various applications [62].

Taking into account the distinctive characteristics of blockchain and DT, we leverage their capabilities to enhance security risk assessment within the context of the IoV. In this scenario, blockchain serves as a security enabler, reinforcing the security of DT-based IoV operations and services. However, integrating two complex technologies (e.g., DT and blockchain) in our IoV use case and their plethora of enabling components exposes a system to newer security threats from adversaries. For instance, when different interfaces, technologies, and computing systems are combined, each functional layer of a DT establishes a set of important services that pose major security threats [8] (also discussed in detail in Section 5.5). Similarly, blockchain has various concerning security threats [40] (also discussed in detail in Section 5.5).

1.1 Motivation

IoV creates numerous new opportunities and applications, benefiting drivers, society, and businesses. The IoV-based vehicles communicate with public networks via vehicle-to-vehicle (V2V), vehicle-to-road (V2R), vehicle-to-human (V2H), and vehicle-2-sensor (V2S) communication channels. These interconnected components of an IoV form a Social Network of Vehicles (SIoV), a simple rendition of the Social IoT of vehicles [18]. The integration of various components and their connectivity exposes IoV to security threats and vulnerabilities that may lead to drastic outcomes if not addressed meticulously. Moreover, within the constraints of DTs for IoV, individual vehicles or the whole traffic systems of cities can be modelled as target objects to create an alignment between virtual spaces (in the cloud) and physical spaces (on the road). Also, using multifaceted and extensive modeling, the server can assist in gaining a perception of the situation, anticipating the effectiveness of certain steps, and making decisions [36]. DTs mirror the real vehicles and traffic systems but avoid the associated risks. With the advancements of smart devices and the maturity of enabling technologies, the IoV is coming out of its infancy and pacing toward becoming a reality.

Furthermore, the accessibility of IoV exposes them to distinct types of security threats. If these threats go unnoticed or the hindrance of planning for countermeasures may lead to drastic outcomes like accidents and loss of lives. A facile-secured IoV can be shaped by tackling security aspects like authentication, data integrity, confidentiality, end-to-end encryption, and access control. However, as the network grows into a SIoV, the threats and vulnerabilities go beyond the capabilities of simple countermeasures. Multiple other more efficient countermeasures are needed to build a secure IoV and SIoV. Therefore, the need to consider a blockchain-based DT approach in the context of secure sensory data exchange between components of IoV is now more than ever.

1.2 Problem Statement

The concept of DT is not limited to just providing 3D models of real-life physical objects; it goes further by assimilating real-time data and interconnectivity. For instance, DTs feed data from their physical counterparts, enabling two-way data flow, making it suitable for monitoring, analyzing, and controlling physical objects from a virtual space and vice-versa. Blockchain-based DTs enhance these processes by ensuring data integrity, provenance, and traceability. They also enable interoperability, collaboration, and standardization of processes. In the context of IoV, sensory data, and crucial stakeholder information require privacy and security against many attacks by adversaries. Without identifying threats, efficient risk mitigation strategies and rigorous forcing of standards may lead to dire outcomes.

Moreover, facile-secured IoV networks lack efficiency in terms of security and integrity of the data communication between its components. There is a lack of extensive research in securing IoV. The capability of blockchain-based DTs in the context of IoV sensory data communication is the least explored. Within the communication channels of IoV, security in terms of data exchange is crucial. Blockchain-based DT allows the study of security before the deployment in the real world, and as a result, security gaps and vulnerabilities can be uncovered. Risk mitigation strategies can also be explored beforehand. Blockchain-based DTs bring much-needed credibility to sensory data communication in IoV. But blockchain and DTs both have some security discrepancies as well. So in this thesis, we have tried to find solutions tacking these issues.

1.3 Research Questions

To address the issues discussed in Section 1.2, we set our main research question: *How to securely build a digital twin solution for the Internet of Vehicles using blockchain?* To answer our main research question, we formulated five sub-research questions:

- **RQ1:** What are the building blocks of IoV?
- **RQ2:** What is the state-of-the-art in securing industrial systems using blockchain-based DT?
- **RQ3:** What are the security risks in industrial systems, and how can blockchain-based DTs be used to mitigate them?
- **RQ4:** How to design V2S data communication in IoV using a blockchain-based DT solution?
- **RQ5:** How to implement V2S data communication in IoV using a blockchain-based DT solution?

1.4 Research Method

We use a design science research approach to accomplish our research objectives and to answer our defined research questions. This method leads to innovation, ideas, practices, and products by analyzing, designing, implementing, managing, and using the information effectively and efficiently. Firstly, further research is required to analyze the existing solutions to implement the project. The best suitable methodology suggested according to the scenario is design science methodology [38]. The guidelines for using design research methods for information systems research and how they align with this thesis are described in Table 1.

In this thesis, the guidelines provided a pathway to be considered effective research. Firstly, the research should be consistent with previous literature. So, we performed a SLR to find the state-of-the-art for using blockchain-based DTs in the context of the

Table 1. Design science guidelines and artefacts produced in this thesis.

Guidelines	**Description**	**Thesis Artefact**
DG1. Design as an artefact	Research must produce a viable artefact	Blockchain-based DT solution
DG2. Problem relevance	Relevant business problem solution using technology	SLR results
DG3. Design evaluation	Rigorous evaluation of the solution	Security risk analysis and implementation of blockchain-based DT. Evaluation of the implemented solution.
DG4. Research contributions	Must provide clear and verifiable contributions in the areas of the design artefact, design foundations, and/or design methodologies.	UML diagram, Sequence diagram, SLR results, Security framework using blockchain-based DT
DG5. Research rigour	Application of rigorous methods in both construction and evaluation of design artefact.	Rigorous survey of recent literature, ERD, UML diagram, and Sequence diagram all leading to the suggested solution. Understanding the necessary building blocks of IoV.
DG6. Design as a search process	The search for an effective artefact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.	Use case design and implementation of the blockchain-based solution.
DG7. Communication of research	Must be presented effectively to technology-based and management-oriented audiences.	Tabular representation of SLR results, Use case diagram, sequence diagram, and flow diagram cater for a wider range of audiences.

security of industrial systems (**DG2**). We remained consistent with the blockchain-based DT implementation and proposed solution (**DG4, DG5**). Secondly, the research should provide a nominal process model on which research is conducted. We accomplish this by creating multiple design artefacts which comprise use cases, sequence diagrams, ERD, and class diagrams (**DG7, DG6**). Thirdly, the research should provide an evaluation model. We used a simulation approach for implementation and security risk analysis leading to the conclusion of this thesis (**DG1, DG3**).

1.5 Contributions

In light of the precarious threat landscape of the complex paradigm of IoV and the interconnections between the enabling components, a comprehensive study into the security threats associated with such systems has become more imminent than ever. Narrowing down the scope of this thesis, firstly, we present detailed elaboration about the building blocks of the IoV and applications of IoV in real life. Secondly, we present a SLR of existing research on securing industrial systems using blockchain as a security enabler. Thirdly, a novel blockchain-based DT solution is presented for handling security risks in IoV. Fourthly, a comprehensive security risk analysis with the identification of threats and risk mitigation using blockchain-based DTs is illustrated. In this analysis, a generic threat model for IoV is elaborated using STRIDE and security risk management using the information systems security risk management (ISSRM) domain model. Moreover, threat modeling and risk management are also showcased for the V2S use case scenario. Finally, the proposed concept is evaluated using a simulation implementation with the help of Microsoft Azure DT services.

1.6 Thesis Structure

The remainder of this thesis is organized as follows. Section 2 presents the details of the main concepts a reader might encounter in this thesis. Details of DTs with their definitions, components, architecture, history, recent practices, and applications. Blockchain technology's influence in DT paradigm, security threats, and analysis models used in the thesis. Section 3 is dedicated to clarifying and providing readers with a broader perspective on IoV and its components. It also discusses the benefits of IoV and how it is connected to this thesis. Section 4 maps out the systematic literature review (SLR) process on existing literature where blockchain-based DTs have been suggested as an extra security layer for industrial systems. Section 5 exhibits the STRIDE threat model and security risk management model using the ISSRM domain model for IoV. Section 6 illustrates the design artifacts for implementing the proposed V2S communication use case. Section 7 accumulates the implementation of the proposed solution using V2S data communication in IoV as a use case. Section 8 discusses how efficient our proposed solution is regarding security and performance. It also showcases the limitations of the proposed solution and points out the future direction of the thesis. Finally, Section 9 concludes this thesis, mapping out my postscript on the findings and summarizes the prospect of blockchain-based DTs as a security countermeasure in IoV.

2 Background

This section illustrates the existing definitions, history, architecture, components, recent practices, and applications of DTs. Details about blockchain, the STRIDE threat model and ISSRM, and blockchain-based DTs are also given along with DTs. It also aims to clarify the concepts used in the later sections of this thesis.

2.1 Digital Twin

DT as a technology has garnered significant acclaim in recent years. However, certain facets of its definition have been circulating within academic discourse has been circulating for a considerable period. There have been many partial definitions of DT in the past. In the past, due to the constraints of enabling technologies such as low computing power, IoT, low storage capacities, rudimentary machine algorithms, etc., DT had no practical applications at that time. With recent advancements in terms of technologies, DT was quick to gain pace.

2.1.1 Definitions and History

The interest in DT has increased over the past few years across various industries, which has resulted in a wide range of definitions and characterizations that have diluted the original initial description. Some misinterpreted concepts about a Digital Model (DM), a Digital Shadow (DS), and a DT remain. A noteworthy clarification was given in [43], where the authors distinguished between the digital forms concerning data flow between the physical and digital parts. A software simulation of the physical process with manual data flow between the two counterparts can be defined as a DM. Whereas, with automated one-way data flow between the real and digital entities, an emulation of the physical process can be considered DS. A DT can be expressed as an intangible technology with bi-directional data flow between real and virtual entities, having features like the ability to perceive, make decisions, self-sustenance, and awareness. It forms an automated feedback loop between the two counterparts, which is the determining factor between these three technologies. Figure 1 shows an adaptation of these three concepts.

In 2012, DTs were formally defined by the National Aeronautics and Space Administration (NASA) as "*an integrated multi-physics, multi-scale, probabilistic simulation mirroring the physical system based on models, sensors, history data, and so forth*" [33]. Michael Grieves informally announced the concept of DT that we are aware of today

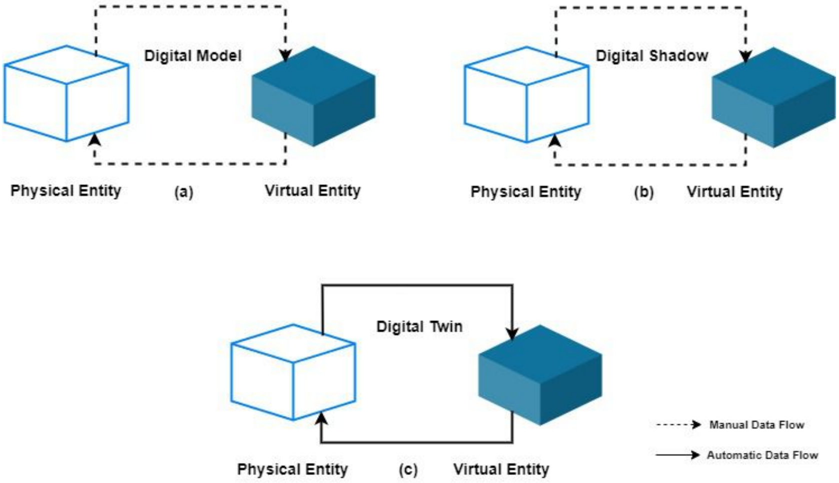


Figure 1. Digital Model **(a)**, Digital Shadow **(b)**, and Digital Twin **(c)** [43]

during his executive course on product lifecycle management (PLM) in 2002, and later he released a white paper with a formal definition of DT [34]. According to the concept there, a DT can be summarized as a collection of "*machines (physical and/or virtual) or computer-based models that are simulating, emulating, mirroring, or twinning the life of a physical entity*" [35]. The concept was then termed the 'Mirrored Spaces Model'. A related idea that states a software model mimicking natural objects using details from the real world was imagined by David Gelernter in 1991 and was called 'Mirror Worlds'. By 2006, the name of the conceptual model proposed by Grieves was changed from 'Mirrored Spaces Model' to 'Information Mirroring Model' [63]. Figure 2 shows the timeline of DT evolution.

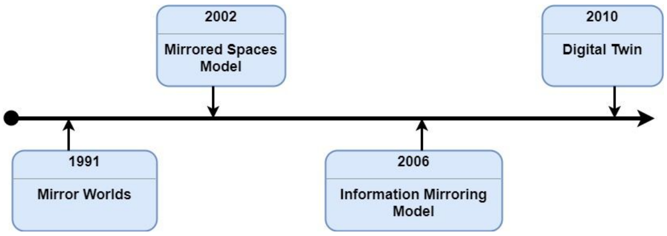


Figure 2. The evolution timeline of DT

Therefore, for recent complex systems with interdependent enabling components, we can define that the main objective of a DT is to pervade physical assets through digital assets by applying specification-based techniques, mathematical models, and application programming interfaces (API), all of which run on servers and/or virtual resources (e.g., virtual machines (VMs), containers and virtual networks), with the primary goals being to forecast errors, variations, and relevant fluctuation that may change a system's natural behavior [11].

2.1.2 Architecture and Components

The preliminary architecture of DTs, conceptualized by Grieves in his white paper [34], was introduced as a three-dimensional model containing physical space, virtual space, and a connection allowing harmony between the data from both spaces (as Figure 3 shows). This 3D model was later extended by Tao et al. into a 5D one which consists of physical space, virtual space, connection, DT data, and service (as Figure 4 shows). The tangible (physical) layer aids the virtual one by providing real asset data. On the other hand, the virtual part helps the physical one with replication and making decisions. The service acts as a support for both spaces while they operate, evolve, and optimize. The primary source of inside information is the DT data repository, while the connection bridges all model components and creates the interconnected loop it tries to achieve [68]. But for today's complex systems, some basic components are sensors (to gather information from the real world), a physical twin, edge processing competency, data security, the digital twin itself, data storage and processing, and reporting interfaces. Another crucial part of DT architecture is visualization for the user. The physical and the virtual layers are connected in the communication element where several protocols and interfaces are available such as Wi-Fi, Bluetooth, and wired connections.

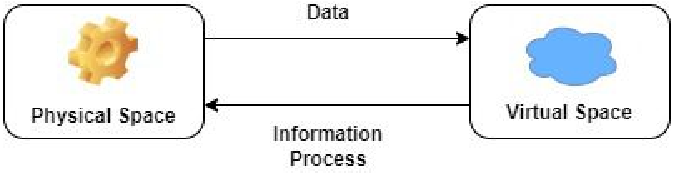


Figure 3. Representation of 3D DT model, adapted from [35]

However, each application domain modifies a DT to reflect its business requirements and data types. The elements pointed out in [34] are integrated with numerous other components. In recent years, most literature containing architectural information about DTs has been domain- and use-case-specific. DT architectures discussed in the bulk of the research differ substantially in the context of their respective components. Table 2 summarizes the mapping of DT requirements with the architectures presented in recent literary works. A more detailed review of these research works is given in [54]. Considering this, it is evident that DT application scenarios are diverse, and their requirements and building blocks depend on the operational use case. DTs can also act as an additional layer of security for systems. From a security perspective, DT

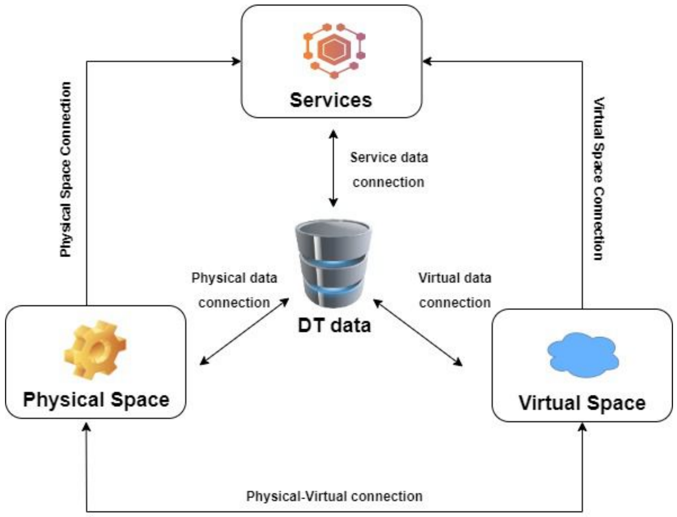


Figure 4. Representation of the 5D DT model, adapted from [68]

building blocks are replication, simulation, and historical data analytics. *Historical data analytics* examines the observed historical behavior of physical assets and predicts the future. *Simulations* are built on user-specific parameters and model the semantics of the real world. Lastly, the *replication* incorporates real-world data to operate and model a digital twin semantically identical to its physical counterpart. These operation modes aid security operations. For instance, behavior-based modeling encourages more effective intrusion detection and security training might benefit from the digital twin's virtual representation. Moreover, replication-based digital twins point to features for incident response and security orchestration [24].

Table 2. Available DT architectures from the literature

Research Work	***Characteristics***					
	Synchronization	***Diagnostic***	***Control***	***Predictive/Prescr.***	***Monitoring***	***Optimization***
[7], 2017	✓		✓			
[15], 2020	✓	✓		✓	✓	✓
[16], 2021	✓					✓
[27], 2021	✓					✓
[32], 2020		✓			✓	
[42], 2019		✓		✓	✓	✓
[47], 2021		✓		✓	✓	✓
[52], 2021					✓	✓
[57], 2020	✓	✓		✓	✓	✓
[65], 2019	✓	✓	✓		✓	
[66], 2020		✓	✓	✓	✓	
[25], 2021	✓	✓		✓	✓	✓

2.1.3 Key Characteristics

DTs exhibit features that make them unique for each use case, and their competency is dependent on the characteristics they possess. The characteristics of DTs are compiled based on two factors: application area and objective. Specific building blocks of a DT are integrated into the corporate environment. However, a generic list of DT building blocks (as seen in Figure 5) are:

- ***Enterprise assets***: Items contributing to enterprise benefits such as an object, subject, system (tangible), or processes (non-tangible) are defined as enterprise assets. These assets mature in different stages of their life cycle. Some physical enterprise assets might not have existed in the initial and last phases of their lifecycle [20].

- ***Asset lifecycle***: The stages of life an enterprise asset goes through, but the number of life-cycle phases of an asset differs between assets. Therefore, the first stage usually refers to the idea of creating an asset, and the final stage contains the closure of the asset. Each stage yields data related to the asset, enabling outside stakeholders to be involved.

- ***Process knowledge***: Efficient autonomous operation of a DT is acutely dependent on process-driven knowledge. The knowledge can be gained by manual input specifications about the physical target object the DT is created for or, in recent times, using artificial intelligence and machine learning to gain insight about the physical object [11].

- ***Process data***: Data in a DT system are not single-layered; rather, they are of various types, such as static, dynamic, behavioral, functional, and environmental. These data can be attained from many sources, including sensors, domain, context, history, and instances [20]. These data are required to be fused to ensure integrity and interoperability. DTs must also have data storage abilities for safekeeping different forms of divergent data sourced from life-cycle phases and even from an involved outside stakeholder [11].

- ***Interfaces and services***: DTs should be able to perform services, including monitoring, simulation, and replication of a target system. They must also perform predictive and prescriptive analysis [20]. Autonomous self-adaptation and self-parameterization capabilities are another set of features recent DT adaptations should include. Real-time data retrieval from a bi-directional close-loop between the real asset and the DT offers unique opportunities to gain real-time insights and visualization for the authenticated user in digitized platforms. These platforms allow commands to be sent to the physical counterpart for their optimization [11].

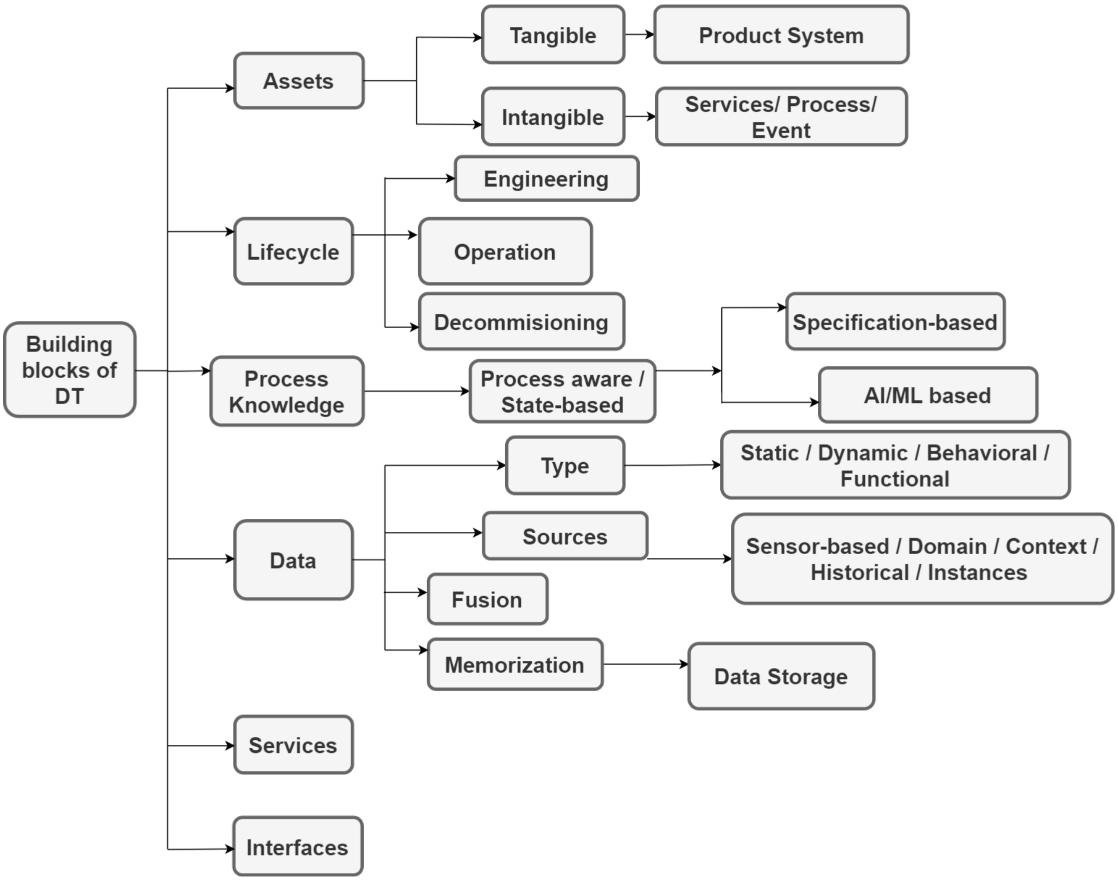


Figure 5. Building blocks of a generic DT, adapted from [70]

As mentioned above, DTs can be multi-faceted depending on the context of their application area and objective. Hence, they can be categorized into multiple aspects which do not sway away from the principle of having three basic parts: a physical asset, a digital asset, and the bi-directional communication between the two counterparts. However, in recent advanced complex systems, DTs can be considered the combination of basic DTs [20]. We can consider a network of similar smart vehicles (cars) as an example. Each vehicle produces its distinct data in the network, so it needs to be monitored by its own DT. However, a reference-DT that describes a vehicle in general (blueprint) might exist, from which each vehicle instance is derived [62].

2.1.4 Functional Layers and Enabling Technologies

Construction of a DT involves a variety of enabling technologies to be implemented. With the proposed 5D model illustrated in Figure 4 as a basis for constructing DTs as an extra layer of security for industrial complex systems, layers of functionality can be mapped. For their research presented in the paper [8], the authors described the four functionality layers as follows:

- ***Layer 1 - data dissemination and acquisition***: Physical space actions and reactions from its surroundings are captured, and instructions for the physical enterprise assets are compiled in this layer.

- ***Layer 2 - data management and synchronization***: Diverse data from multi-faceted sources are formalized and refined here so that Layer 3 functions can be carried out. Both spaces in a DT (physical and virtual) are interdependent; hence, services to allow harmony between data from both spaces must be considered.

- ***Layer 3 - data modeling and additional services***: Modeling refers to the process of representing a physical entity in digital forms that can be processed, analyzed, and managed by computers. Through digital modeling, this layer cites the different states, their dependent actions, and geometric shapes. DT-related modeling involves geometric, physical, behavioral, and rule modeling. Other additional services for perception, analysis for security, and countermeasures are also managed within this layer.

- ***Layer 4 - data visualization and accessibility***: Data visualization presents data analysis results in a straightforward, intuitive, and interactive manner. This DT layer allows stakeholders to represent insights accumulated from the physical spaces and make real-time decisions affecting the states of the physical assets. Since stakeholders of different trust levels are involved throughout the lifecycle of a DT, sensitive data needs to be presented only to relevant end users. Stakeholder access management is another function of this layer.

Therefore, data manipulation and concerned technologies are the two determining factors in distinguishing between the layers of a DT. Layer 1 devices installed close to the physical assets often capture and process measurement and control values corresponding to real-world assets. These captures give the DT and its models the opportunity to synchronize with their physical counterparts and start simulation processes that result in a more thorough and precise grasp of the real-world event. Technologies affecting data interpretation and visualization are deployed in Layers 2-4, where all DT logic and their simulation to end users are handled [8].

Existing technologies used to perceive the physical space in Layer 1, the usage of Cyber-physical Systems (CPSs), and the Industrial Internet of Things (IIoT) are a norm. CPSs are embedded systems with combinations of computation, networking, and physical processes which allow IIoT to impact the computation result. As a result, a closed loop between sensors-controllers-actuators is formed. Controllers synchronously compute real-world states and execute command and control (C& C) instructions to alter real-world behavior. Whereas IIoT is a technology that allows smart and autonomous devices to connect to the internet without the necessity for synchronous communication among them or closed-loop communication with the real world. Using automation protocols, both CPS and IIoT feed off of each other. As a part of IIoT, DTs also benefit from inter- and intra-twin communication. Using the DT paradigm, new cellular technologies like 6G can also aid in digital transformation. More precise and harmonic updates between both spaces can benefit from new technologies. Computing infrastructures required in Layers 2-4 may reside on a single server or spread throughout the system. Computing resources based on edge, fog, and cloud are more prominent for DTs. Data storage stores the collected data for further processing, analysis, and management. Data storage is inseparable from database technologies.

However, traditional database technologies are no longer feasible due to multi-source DT data's increasing volume and heterogeneity. Big data storage technologies like distributed file storage (DFS), NoSQL database, and cloud storage are drawing more attention. Data visualization is manifested in various ways, such as histograms, pie charts, line charts, maps, bubble charts, tree charts, and dashboards [8]. Recently, blockchain technology as an enabler for a distributed ledger, data storage, authorization, and data integrity has also gained traction among researchers [68].

2.1.5 Existing Practices and Applications

DTs are virtual replicas of physical objects, systems, or processes that can be used to simulate, monitor, and analyze their behavior and performance in real-time. Digital twins have numerous applications in various fields, including manufacturing, healthcare, transportation, energy, etc. Some of the most common applications of digital twins are:

- ***Manufacturing***: DTs are used in manufacturing to simulate production processes and optimize them for improved efficiency and quality, including its logistical aspects and enable detailed visualization of the manufacturing process from single components up to the whole assembly. DT systems for manufacturing have four key components, including physical entity, virtual model, data, and service system [43]. The following three core techniques have been researched in recent literature: (1) construction of the DT model; (2) acquisition and management of data, and (3) services that consist of prediction and production management.

- ***Healthcare***: In the healthcare industry, DT technology is being utilized to improve, analyze, and predict outcomes for patients, hospitals, and the pharmaceutical industry. While DT technology is commonly used for machines and processes, creating a DT of the human body is more complex and challenging due to the expense and time-consuming nature of acquiring patient data through blood tests,

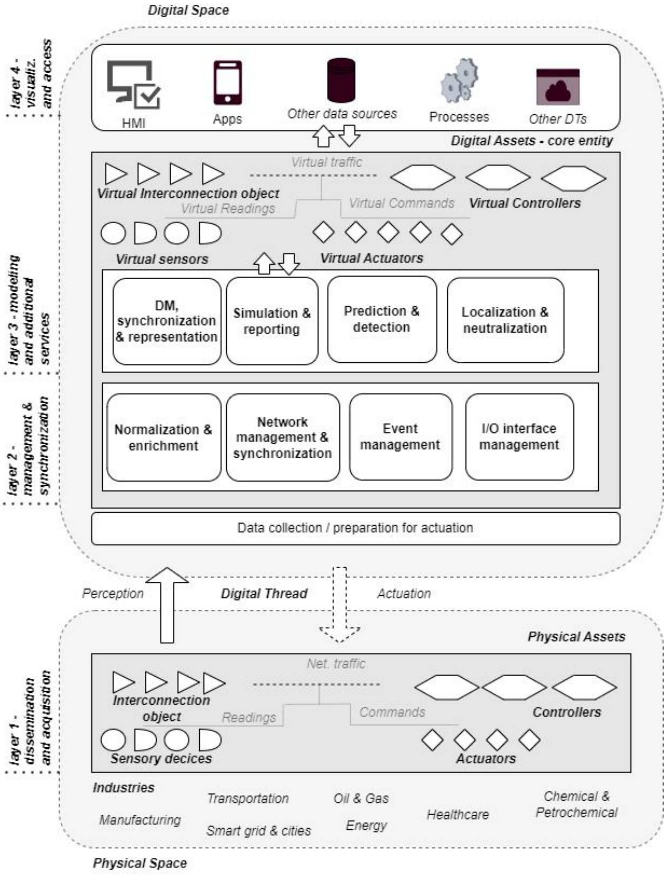


Figure 6. Digital Twin layers, adapted from [8]

imaging systems, and health scans. Developing a Digital Twin of the human body is expected to accelerate the diagnosis and treatment processes and increase accuracy rates. As a result, the concept of the "Digital Patient," or patient-specific modeling to support medical decisions, has gained prominence [26].

- ***Smart Cities***: DTs can be used to simulate and optimize the functioning of smart cities, including traffic flow, energy usage, and waste management. This can help cities become more sustainable and efficient. DTs are increasingly being used in the development and management of smart cities. A DT of a smart city is a virtual replica of the city that can be used to simulate, predict, and optimize its infrastructure, services, and resources. This technology can help city planners and decision-makers to visualize the city in real-time, identify potential issues, and test different scenarios before implementing them in the real world. DTs for smart cities can be used to monitor and manage various systems such as transportation, energy, water, waste management, and public safety. For example, a DT can be used to simulate traffic flows, optimize traffic signal timings, and analyze the impact of new transportation projects on the city's infrastructure [19]. Additionally, it can help city managers to predict and mitigate the impact of natural disasters, such as floods or earthquakes, by testing different evacuation scenarios and emergency response strategies. Overall, DTs for smart cities offer an efficient and cost-effective way to manage and optimize city infrastructure and services while reducing risks and improving the quality of life for citizens [29].

- ***Energy***: DT technology is increasingly being utilized in the energy industry, where a digital asset is a virtual replica of the physical asset that can be used to simulate, predict, and optimize its performance throughout its entire lifecycle. DTs for energy assets can be used to monitor and manage various systems, such as drilling, production, and refining operations. For instance, a DT can be used to simulate different drilling scenarios in the oil sector and optimize well placement and production rates. It can also be used to monitor equipment health and predict maintenance needs, allowing for more efficient maintenance scheduling and reducing downtime. Moreover, DTs can enable real-time monitoring and control of energy assets, allowing operators to adjust operations quickly in response to changing conditions or emergencies. It can also facilitate collaboration between different stakeholders and teams, including engineers, operators, and contractors, who can work together to optimize performance and reduce costs [31]. Overall, DTs in the energy industry can significantly improve asset performance, reliability, and safety while reducing costs and minimizing environmental impact.

- ***Transportation***: DTs are also used as an additional maintenance layer for smart transport systems for whole cities. They can be used to simulate and optimize transportation systems, including traffic flow, vehicle performance, and logistics. This can help reduce congestion, improve safety, and optimize supply chains. DTs can act as several entities in an intelligent transport system: **(a)** Autonomous driving virtual simulation test platform, **(b)** Intelligent Traffic Simulation Platform, **(c)** Intelligent Traffic Management System, and **(d)** Intelligent Highway Management System. DT technology offers new ideas, concepts, and methods for intelligent traffic development, driven by the convergence of technological advancement and demand upgrades [10]. It can overcome the current technical challenges in building intelligent traffic management systems by organically combining cutting-edge technology in several areas, such as intelligent perception, network communication, artificial intelligence, and intelligent control. Through virtual service reality and data-driven governance, DT for ITS will continue to evolve and eventually become a fully functional technical operation system with the continued development of related technologies and multidisciplinary cross-fertilization.

2.2 Blockchain

Blockchain was first introduced by Satoshi Nakamoto in 2008 [53]. It combined existing technologies like distributed ledgers, cryptography, hashing, and consensus protocols. Any transaction record is stored in a chain of data packages (blocks) and distributed across a peer-to-peer (P2P) network where each peer is an involved node in the chain with a copy of the blocks. Every transaction or digital event has to be validated using a particular consensus mechanism, defined as a set of guidelines and policies used to maintain the trustworthiness of the records in a ledger and update operation. Each blockchain technology has a different consensus mechanism agreed upon by the stakeholders before the chain deployment [55]. As a transaction is validated by most of the involved nodes using a specific consensus, it is recorded in a new block with a timestamp, previous block hash, and transaction data (as Figure 7 shows). Thus, blockchain technology yields a secure, decentralized, persistent, fault-tolerant, and traceable chain that grants decentralized automated transactions without the need for central governance [12]. Key characteristics of a basic blockchain can be described as follows:

- ***Decentralization***: Unlike conventional transactions, which need a central trusted mediator to approve them, blockchain technology involves distributed, multiple involved nodes to validate each transaction and has an identical copy of the ledger [44]. This leads to the transactions being fault-tolerant and transparent, with flexibility in user control and resistance to attack.

- ***Persistency***: Blockchain with the usage of consensus, timestamp, and a cryptographic seal creates immutable transaction blocks in the chain [55]. This allows data persistence, protection against faults, and authorized ownership of all transaction records.

- ***Anonymity***: All transactions and validations across the blockchain are done maintaining anonymity for the stakeholders, and with the usage of hash addresses [44]. Hence, the confidentiality of user information is protected, distinguishing blockchain from conventional transactions.

- ***Traceability***: Each block contains information about the previous block in the chain, and the transactions are only stored sequentially. As a result, tampering with the blocks or any harmful block can be easily traced.

Moreover, due to its promising secure distributed framework that enables information sharing and coordination across all participating entities, the blockchain has drawn enormous attention from academics and practitioners in diverse disciplines, including finance, law, and computer science. The recent swift acceptance of blockchain was mostly triggered by the enormous success of Blockchain (the prototype of cryptocurrency).

The advent of other alternatives like Ethereum (ETH) and Ripple lead the way to the acceptance of blockchain technology across other non-financial application domains such as intellectual property and proof of location. voting, and healthcare [68].

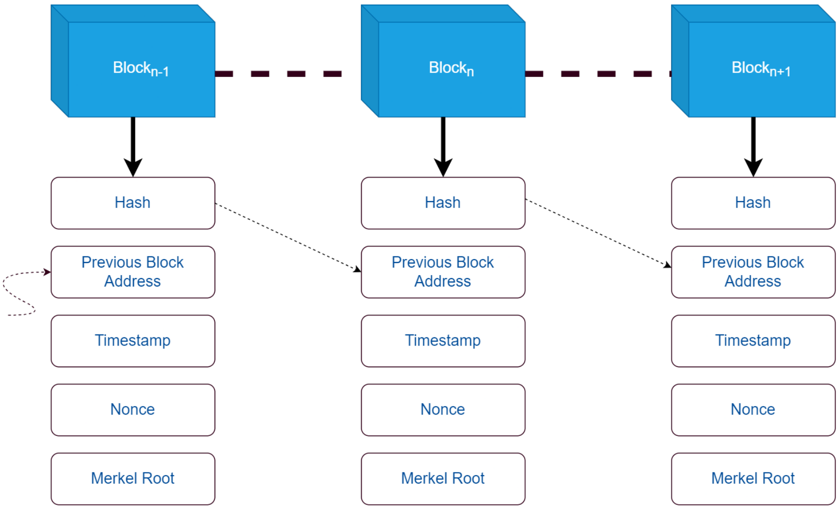


Figure 7. A single block contents of blockchain

2.2.1 Types of Blockchain

Based on the characteristics and policies, blockchain technology is broadly classified into three types:

- ***Public***: Public blockchains are considered open and allow anyone to join the decentralized ledger network. They are basically permissionless, meaning no restrictions are placed on anyone accessing the network or validating any transaction. The public blockchains persist with the characteristics like immutability, transparency, and security through distributed consensus algorithms. Examples include Bitcoin (BTC) and Ethereum (ETH).

- ***Private***: Also referred to as *permissioned* blockchains. These are restricted networks where participants need validations and permission to access the network. These types of blockchains are used mainly in organizations where maintaining control of participants joining the network is mandatory. Private organizations use permissioned blockchains for internal purposes like supply-chain management or intra-organization collaborations. Private blockchain frameworks like Hyperledger Fabric and R3 Corda are examples of private blockchains.

- ***Hybrid***: A hybrid model, also referred to as *Consortium* blockchains combining features from public and private blockchains. Multiple organizations or entities form a consortium and jointly govern the network in consortium blockchains. This type can be useful for large organizations with multiple stakeholders, all sharing certain control and privacy levels. These blockchains perform better than public and private blockchains regarding transactions per second (TPS) and scalability while persisting with a certain level of decentralization.

2.2.2 Ethereum

Ethereum is a public blockchain technology that is not only limited to cryptocurrency or payment. It is an open-source programmable technology that can be used for many other digital assets. Ethereum makes it possible to perform transactions in a decentralized manner [51]. The building blocks of Ethereum are illustrated below:

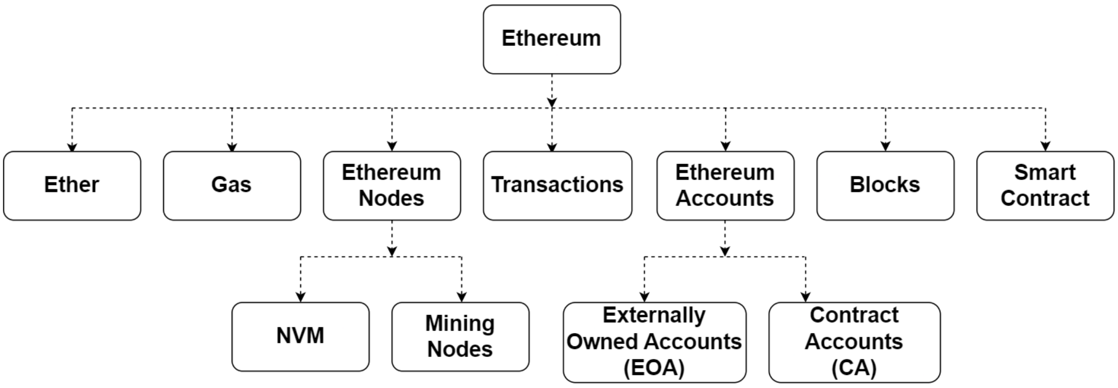


Figure 8. Building Blocks of Ethereum.

- ***Ether (ETH)***: The currency name used in Ethereum is known as Ether.
- ***Gas***: The fixed fee for every transaction depends upon the current market value of Ether. It protects the network from attacks like distributed denial of service(DDoS) attacks.
- ***Ethereum Nodes***:
  - ***NVM***: Nodes responsible for executing functions written in smart contracts. They have minimal access to the network, with access to only EOA, CA, and its storage.

  - ***Mining Nodes***: Nodes with direct access to the network chain can get gas whenever they perform the transaction. All the mining nodes are given a puzzle to solve, and the first to solve it will be able to write the transaction on the chain

- ***Transactions***: An agreement between two parties to exchange assets.

- ***Ethereum Accounts***:

  - ***Externally Owned Accounts (EOA)***: Accounts owned by holders of Ether balance in the Ethereum network. A private/public key pair is generated with each new account creation.
  - ***Contract Accounts (CA)***: These are similar to EOA but lack private and public keys.

- ***Blocks***: Ethereum block is a collection of transactions and forms a blockchain, where the first block is called the genesis block without any parent.

- ***Smart Contract***: A digitized business rule which is written in Solidity programming language with the consent of all account holders to perform business transactions.

2.2.3 Hyperledger Fabric

Hyperledger Fabric (HLF) is a permissioned blockchain. It is distributed and has an access control mechanism to disable unauthorized users accessing the network, improving permissionless blockchains. It also consists of channels and organizations, each with its smart contract. Each organization can make its distinct transactions without disrupting the other network infrastructure. HLF is written in (e.g., Go, Node.js, and Java) and is used in automating various business processes. What makes it unique and customizable is the fact that its architecture involves an execute-order-validate algorithm. Hyperledger fabric network consists of following components [1]:

- ***Assets***: An asset can be any valuable entity that is a part of the business having some monetary or business value. It has to have some state which needs to be stored and owned by an owner. Regarding HLF, a collection of states as key-value pairs in the ledger.

- ***Shared Ledger***: HLF needs to store blocks in the shared ledger. It consists of World State and Blockchain.

[1] https://developer.ibm.com/articles/blockchain-basics-hyperledger-fabric/

- ***Smart Contract***: Chaincode in HLF is the smart contract equivalent of other blockchain technologies. It holds the system's business logic and is responsible for interacting with the ledger and performing transactions.

- ***Peer Nodes***: Core components hosting the chaincode and the ledger. There are three types of peer nodes:

  - ***Endorsing Peer Nodes***: Based on the chain code results, for a single chaincode container, these nodes create the transaction proposal. Installation of chain code is required on these nodes.
  - ***Committing Peer Nodes***: These nodes solely bear the responsibility of maintaining the complete ledger and do not trigger any chaincode functions.
  - ***Ordering Peer Nodes***: Special peer nodes containing the entire ledger transaction history, including valid and invalid transactions. They differ from other nodes because other nodes only contain valid transactions. Additionally, they are accountable for accepting the endorsed transaction proposal and disseminating it to other nodes for validation.

- ***Channel***: Logical unit forming a cluster of peer nodes that perform transactions within a channel without the other channels knowing about it.

- ***Organizations***: Members of the network with one or more peer nodes. HLF contains multiple organizations where inter-organization peer nodes interact with each other and fulfill business nodes.

- ***Membership Service Provider (MSP)***: Certificate manager in HLF responsible for authenticating the users. It is implemented as a Certificate Authority (CA) and is responsible for the enrollment of users to the network and transaction invocations. Furthermore, it guarantees a secure linkage between the users and components within the blockchain network.

- ***Ordering Service***: A collection of transactions and transfers among diverse peer nodes is termed as an ordering service. Additionally, it ensures the smooth flow of transaction information across the network. Solo and Kafka [2] configuration mechanisms are used for ordering service. Which is deprecated in HLF version 2.0 onwards.

2.2.4 EOS

The open-source, next-generation EOSIO [3] blockchain platform has adaptable usefulness and industry-leading transaction speed. EOSIO is a blockchain platform made for public

[2] https://kafka.apache.org/intro

[3] https://developers.eos.io/welcome/latest/introduction-to-eosio/index

and private blockchain deployments and intended for enterprise-grade use cases. With its role-based permissions system and secure application transaction processing, EOSIO is adaptable to a variety of business demands across sectors. On EOSIO, distributed apps are created using the same development methodologies and programming languages as non-blockchain applications. Because it enables them to utilize their chosen development tools, familiarity with the development environment benefits application developers and creates a smooth user experience. The core concepts[4] of EOS blockchain are :

- ***Account***: An account is a name that can be read by humans and is kept on the blockchain. Depending on the setup of the permissions, it may be owned by a single person or a group of people through authorization. Any legitimate transaction must be transferred or pushed to the blockchain using an account.
- ***Wallets***: These clients store keys that may or may not be associated with permissions or one or more accounts. It has two states, locked(encrypted) and unlocked(decrypted). A high-entropy password protects the unlocked key. EOS repository is bundled with a CLI client called cleos that interfaces with keosd and showcases this pattern together.
- ***Authorizations and Permissions***: Permissions are arbitrary names that serve to specify the conditions for a transaction sent on their behalf. By using linking authorization or *linkauth*, permissions can be granted for control over particular contract actions.

Moreover, the EOS blockchain uses Delegated Proof of Stake (DPoS), which is a decentralized consensus algorithm capable of meeting the performance requirements of applications on the blockchain. A continuous approval voting system allows token holders to select block producers.

2.2.5 Why Ethereum?

Depending on the business requirements and the nature of the stakeholders involved, we need to decide whether blockchain is necessary. The authors of the article [72] provide a flow chart to decide which blockchain to use for the business use case (as shown in Figure 9).

In this thesis, we have chosen Ethereum as the blockchain to implement the proposed solution. Due to the comparatively lesser complexity in setting up and the simulation nature of our implementation, we have gone with this technology. The more complex and organization-specific solutions like HLF requires much more resource and time for implementation. We are implementing the blockchain-based DT solution in Azure Digital Twin as Platform-As-A-service (PaaS).

[4] `https://developers.eos.io/welcome/latest/introduction-to-eosio/core_concepts`

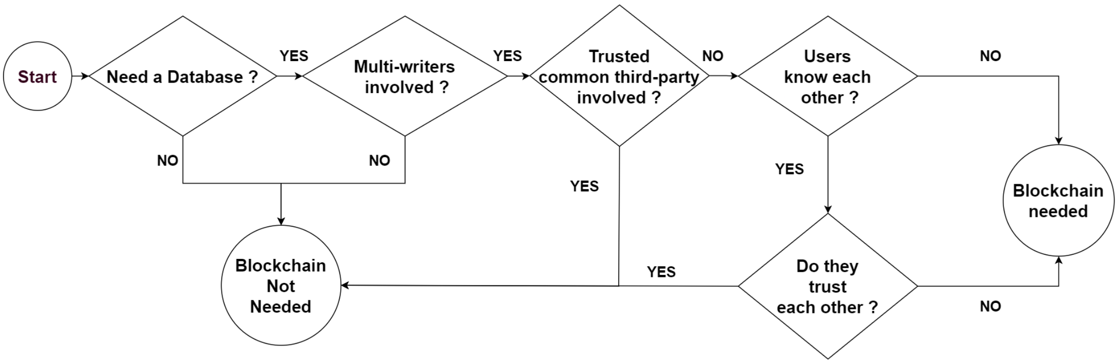


Figure 9. Do you need blockchain? adapted from [72].

As we can see in Table 3, Ethereum is one of the most adopted blockchain technology with the largest community of developers. With Ethereum V2.0, the less efficient Proof-of-Work consensus has been replaced with Proof-of-Stake (PoS) mechanism. This upgrade made it more efficient as blockchain technology. EOS as a blockchain technology is comparatively new and needs more acceptance among developers. So, for the naive implementation of our suggested solution in this thesis, Ethereum is sufficient. The unavailability of C# sdk for HLF is another reason for choosing Ethereum for our solution. A Custom HLF network for the implementation is considered out of scope for this thesis and suggested as future work.

2.3 Blockchain-based Digital Twin

Blockchain-based DTs refer to creating a digital replica of a physical object or system, which is then stored and managed on a blockchain network. Blockchain has captivated mainstream attention because of its distinctive features, such as security, anonymity, traceability, accountability, and trust. Whereas the backbone of the ability of DTs to provide its services for the industry is the assumption that the data shared between its spaces (physical and virtual) is secure, valid, and readily available. Therefore, authentic and protected data is a priority to unlock the full potential of DTs [68]. With its inherent properties discussed above, blockchain ranks on top to ensure the integrity of data sources and data within a DT paradigm. One of the key benefits of blockchain-based DTs is that they can help reduce the risk of fraud and counterfeiting. By storing information on a blockchain, the digital twin data, stakeholders, and automated actions can be easily verified and validated, ensuring that it is an accurate representation of the physical object or system. Another benefit of blockchain-based DTs is that they can be used to create a shared digital history of an asset or system. Services provided by DTs can be automated with smart contracts. This can be particularly useful in industries such as manufacturing

Table 3. Ethereum compared to Hyperledger Fabric and EOS.

Core Concepts	**EOS**	**Ethereum**	**Hyper-ledger Fabric**
Design	Focus on performance and scalability	Generic purpose	Enterprise-grade permissioned blockchain
Consensus Algorithm	Delegated Proof of Stake (DPoS)	Proof of Stake(PoS)	Practical Byzantine Fault Tolerance (PBFT) and other consensus mechanisms.
Smart Contracts	Smart contracts in EOS is known as Action. They are typically programs written in C++	Allow smart contracts written in Solidity, Vyper and other programming languages	Practical Byzantine Fault Tolerance (PBFT) and other consensus mechanisms.
Privacy	Very limited privacy with transparent transactions by default	Although transactions are transparent, privacy can be achieved through zero-knowledge proofs	Allows the most privacy features with channels, private transactions, etc.
Governance Model	Decentralized with token holders electing fixed number of block producers	Decentralized and decisions made with community consensus	Multiple organization consort and participate in collectively governing the network
Adoption	Considerable amount of adoptions by dApps	One of the most widely adopted blockchain platforms, with a large developer community	Mostly adopted by organizations and enterprises for specific use cases

and supply chain management, where it is important to track the entire lifecycle of an asset, from its creation to its disposal [73]. Overall, blockchain-based digital twins have the potential to revolutionize the way we manage and track physical assets and systems, providing enhanced security, transparency, and immutability. This paper provides a detailed look into blockchain-based DTs' benefits, security issues, and countermeasures in Section 5.

2.4 Internet of Vehicles

In the new Internet of Things (IoT) era, conventional VANETs have evolved into the IoV. Each vehicle in IoV is envisioned as an intelligent object equipped with sensing platforms, computing facilities, control units, and storage. They are also connected to any entity (other vehicles, RSUs, charging/gas stations, cloud, and so on) via vehicle-to-everything (V2X) communications [17]. Intelligent vehicles can take different roles, i.e., being both a client and a server, taking and providing big data services, leading to numerous new IoV applications, from assisted/autonomous driving and platooning, secure information sharing and learning traffic control and optimization [58].

IoV technologies will serve to address the major issues of contemporary transportation, including improving traffic flow, reducing pollution and fuel consumption, and enabling car-sharing services to save money and space. Additionally, each vehicle can be an edge server or information center for users and IoT devices on the road. A wide range of industry sectors, including transportation, automobile manufacturing, energy, automation, software, and information and communication technology, are all significantly impacted [61]. Undoubtedly, the IoV occupies a central position in the upcoming Industrial Revolution.

Even though IoV supports a wide range of promising applications, its heterogeneous architecture, complicated resource needs, and diverse service requirements call for novel designs, including new radio technologies as well as new network architecture and protocols [62]. IoV is distinct from other IoT networks and the conventional Internet. Whether automated or driver-assisted, each vehicle would produce a torrent of data that might be thousands of times more than a person would produce. Vehicle density can vary significantly over time compared to peak hours and in different locations (main roadways versus side streets). While infotainment applications may be able to accept a certain amount of latency, emergency alerts, and real-time cooperative control messages have strict delay limits (a few milliseconds).

2.5 Security Threat Modeling

Building secure software means handling security risks beforehand so that the software is not affected when exposed to attacks by adversaries. One of the main activities to make secure software is threat modeling. It is defined as a process by which security threats of software can be identified, enumerated, and prioritized, using the perspective of a user, which mainly takes place during the design phase or sometimes in later stages of the software development life-cycle [64]. Since architectural flaws in software systems initiate most vulnerabilities, threat modeling is of the essence in building secure software [50]. In the process of creating safe systems, various threat modeling techniques and procedures, ranging from theoretical frameworks to practical methodologies, have been created over time. The steps to perform threat modeling are **(a)** asset identification, **(b)** data flow diagram (DFD), and **(c)** STRIDE modeling. Asset identification refers to finding objects or stakeholders involved in different stages of the software process. A DFD is a graphical representation of the most important actors, processes, services, components, and data stored in the system, highlighting how information flows between each of them. Microsoft Threat Modeling Technique with STRIDE in agile projects has recently gained traction in academics and industry [64]. STRIDE is commonly utilized to identify different types of relevant threats in each interface defined in DFD. STRIDE is based on the abbreviation of each of the attack types [64]:

- *Spoofing (S)*: refers to a rouge person or program which impersonates a legitimate user or program.
- *Tampering (T)*: an adversary modifying application resources, such as in-memory data, to harm the system.

- *Repudiation **(R)***: refers to a user (legitimate or malicious) intentionally or unintentionally denying the execution of an action within the system,
- *Information disclosure **(I)***: unwanted but risky exposure of private and sensitive information of the system.
- *Denial of service **(D)***: adversaries making resources unavailable to legitimate users.
- *Elevation of privilege **(E)***: occurs when a user with restricted access uses security flaws to acquire privileged access to a resource.

In this thesis, we have used STRIDE to identify the security threats in IoV in Section 5. We also use this model to identify the threats in the V2S communication use case to evaluate the proposed blockchain-based DT solution. The six threat categories in STRIDE encompass a broader range of threat landscape and allows systematic examination of potential threats. In the case of IoV, its multi-faceted communication means and the myriad of components presents a system for attacks from multiple directions of threat vendors. So, STRIDE provided the necessary standard framework to identify these threats for our proposed solution.

2.6 Security Risk Management

The Information System Security Risk Management (ISSRM) domain model provides a framework for identifying, assessing, and managing security risks to an organization's information systems. This model addresses the security strengths related to Information System (IS) domain [21], the main aim being protecting business assets from all harm that may arise accidentally or deliberately. Some core definitions of ISSRM concepts can be grouped into three categories: **(a)** Asset-related concepts, **(b)** Risk-related concepts, and **(c)** Risk treatment-related concepts [6]. A brief description of these concepts (as Figure 10 shows) is given below:

- ***Asset-related concepts***: Assets that are critical to protecting and the criteria that ensure asset security are contained in these concepts. An *asset* can be defined as anything that adds value to the organization, whereas *Business assets* are information, process, capabilities, and skills inherent to the business that adds value to the organization [6]. *IS assets* are components of the IS that support business assets. Confidentiality, integrity, and availability are considered *Security criteria*, which are extended later on depending on the business context [49].
- ***Risk-related concepts***: These define the risk components, i.e., threat, vulnerability, threat agent, attack method, and impact [5].

- ***Risk treatment-related concepts***: Defines how countermeasures are taken against induced threats to the business assets and how the system can be improved to have more security [49].

Moreover, the ISSRM process can be broken down into seven steps: **(a)** asset identification, **(b)** security objectives determination, **(c)** risk assessment, **(d)** risk treatment decisions **(e)** security requirement definition, **(f)** control selection, and **(g)** control implementation. In Section 5, we have used this domain model to categorize security risk countermeasures of blockchain-based DTs using IoV as a use case.

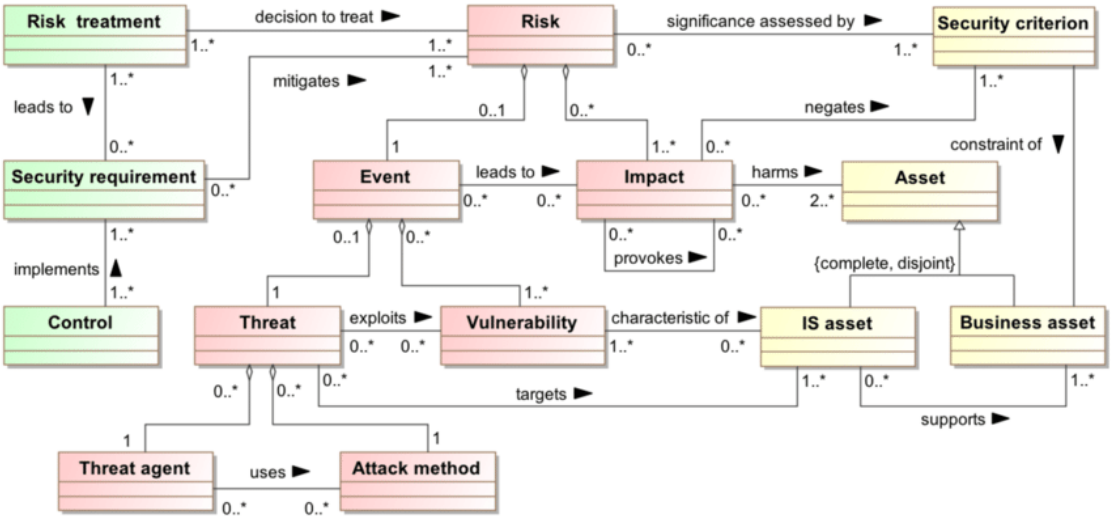


Figure 10. Adaptation of ISSRM domain model [6]

Moreover, ISO/IEC 27001[5] is an international standard to provide requirements for periodically establishing, implementing, maintaining, and improving information security practices in systems. In the article [30], the authors conducted an extensive survey on information security management systems to explore the extent of compliance with ISO/IEC 27001 standards. Out of the models or frameworks discussed in the survey, Information System Security Risk Management (ISSRM) covered most of the Plan-Do-Check-Act (PDCA) model used for evaluation. ISSRM is classified as Proficient because it meets 10 to 14 ISO/IEC 27001 standard requirements.

All things considered, in this thesis, we have used ISSRM as a means for security risk management of our proposed blockchain-based solution. After identifying the threats using STRIDE, it was necessary to explicitly discuss what event might trigger the threat, what risks are associated with it, the vulnerability causing the risk, and what assets or business assets can be affected by each threat. ISSRM domain model aided in this

[5] https://www.iso.org/standard/27001

process, and the results were considered while designing the proposed blockchain-based DT solution for IoV.

3 Building Blocks of IoV

This section aims to answer the research question **RQ1: What are the building blocks of IoV?**. IoV refers to the amalgamation of vehicles, transportation infrastructure, and the internet to create an interdependent ecosystem. The aim is to enable vehicles to communicate with each other, with infrastructure, and with other devices or systems. It also concerns using myriad technologies like sensors, communication networks, and data analytics. With the rapid development of computation and communication technologies, IoV assures huge potential for commercial interest and research value. As a result, it is getting interest from researchers and commercial companies.

3.1 Design of IoVs

The design of an IoV system requires various components and considerations to ensure seamless communication, efficient data exchange, and reliable interconnectivity. It is a complex integrated network system that interconnects people within and around vehicles, intelligent systems on board vehicles, and various cyber-physical systems in urban environments. Some key design aspects of IoV are:

- ***Communication Infrastructure:*** IoV has severe reliability on robust communication infrastructure to enable its array of communication channels. The infrastructure includes wireless networks, such as cellular networks, dedicated short-range communication (DSRC), emerging technologies like 6G, roadside units, and communication protocols.

- ***Vehicle-to-Vehicle (V2V) communication*****:** Vehicles in IoV confide in each other for data about surrounding, road conditions, and so on. V2V communication allows information exchange between vehicles directly with nearby vehicles. It typically uses wireless networks like Wi-Fi or DSRC. They share information about speed, direction, acceleration, braking, and cooperative driving.

- ***Vehicle-to-Infrastructure (V2I) communication:*** IoV also depend on V2I communication which facilitates traffic management,traffic signal optimization. It also empowers the transference of real-time traffic information and warning to vehicles. The communication involves elements like traffic lights, road signs, toll booths, and roadside units.

- ***Vehicle-to-Sensor (V2S) communication:*** In the IoV ecosystem, vehicles are typically equipped with a range of sensors and devices which include GPS receivers, accelerometers, cameras, radars, LIDAR, and vehicle diagnostics systems. Capturing and providing real-time data related to the vehicle location, environment, performance, and surrounding objects is done by these onboard sensors and devices.

- ***Security and Privacy:*** One of the most critical considerations that need to be considered is the security and privacy of IoV. The potential risks associated with access, data manipulation, and malicious by adversaries result in drastic outcomes and might cause life risks. Facile-secured IoV is not enough to protect against these threats, but more robust security measures are necessary. Encryption, authentication protocols, role-based access control, and intrusion detection systems are necessary to protect the integrity, confidentiality, and privacy of vehicles, vehicular data, real-time sensory data, etc.

- ***Standards and Interoperability:*** Standard protocols and interfaces need to adhere to ensure interoperability and compatibility across different components of IoV. Vehicles, infrastructure components, and service providers must communicate and exchange data between them. Standards need to be followed and agreed upon by the industry. V2V communication, V2S data exchange, and V2I communication protocols should follow protocols for secure data flow between components in IoV.

3.2 Applications of IoVs

IoV application is extensive, and the research being done in countries implementing it is different [41]. V2V, V2I, and V2S communication technologies result in a better driving experience and ensure safety. IoV applications can be divided into two categories, namely **a)** ***service applications*** and **b)** ***safety applications***. The onboard service application is the most common IoV application. The service applications aim to enhance the experience of the driver and passengers. Service applications are further categorized into three sub-categories:

- ***Daily Service***: This application includes various life service information provided to the vehicle and driver by accessing a network, such as real-time updated weather information, navigation information, entertainment information, and online news.

- ***Business Service***: Due to the vehicle's mobility in IoV, it needs information about local shops, scenic spots, and shopping malls. Vehicles get their business addresses, operating hours, and product availability information. This information is obtained using the RSUs nearby.

- ***Diagnostic Service***: This type of application includes Diagnostic services for vehicles. Cloud-based maintenance systems aid in the diagnosis of driver-reported vehicle problems and might suggest audio or video solutions.

Moreover, Safety applications are designed to aid in improving the safety of drivers, passengers, and passers-by. V2S, V2V, and V2I communications enhance information exchange between vehicles and other devices ensuring safety, achieving emergency warnings, and avoiding accidents. These applications are classified into three sub-categories as well. They are a) *Public Safety*, b) *Road Information Warning*, and c) *Intelligent Traffic Management*. Without a doubt, IoV requires secure information exchange between different layers of IoV. The multi-faceted nature of IoV involves heterogeneous and enormous amounts of stakeholders and their interactions. Data exchange between these components across layers of IoV must ensure data confidentiality, integrity, and security. As a security enabler in IoV, blockchain-based DTs have been suggested in this thesis to target the security discrepancies of the conventional IoV.

3.3 Use Case

Sensory data from the onboard sensors are aggregated in the onboard unit in the vehicles, and the real-time data is crucial for various purposes. This can include traffic prediction, adaptive routing, real-time alerts, and driver assistance systems. The data can also be used to enhance situational awareness and decision-making for vehicle occupants. In some cases, vehicles can also provide feedback or response to the sensors. Although DTs for IoV pose many attack vendors for adversaries, V2S communication is where adversaries can sabotage the whole system by manipulation. So this scenario is taken into account for the case study. The scenario of using sensors in the vehicle and how the collected data can be analyzed to identify an attack with the help of blockchain-based DTs. The proposed solution can help to identify an attack and the originating sensor using DT.

3.3.1 Vehicle-To-Sensor Communication

The real-time data from the vehicle is collected by the various sensors (such as radars, INS, temperature sensors, mechanical sensors, etc.) on board the vehicle. The onboard unit (OBU) integrates data from all the sensors and sends it to the IoV network cloud in real-time for data and perceptive analysis. The data analysis process is done in the analysis layer. These data are later used for different services provided by IoV. In an IoV environment, vehicles are mobile, consuming and sharing information simultaneously to ensure road safety. Hence, it is essential to provide secure data with limited accessibility. A V2S communication occurs in the perception layer of IoV and sends data to the data analysis layer using means of the transport layer.

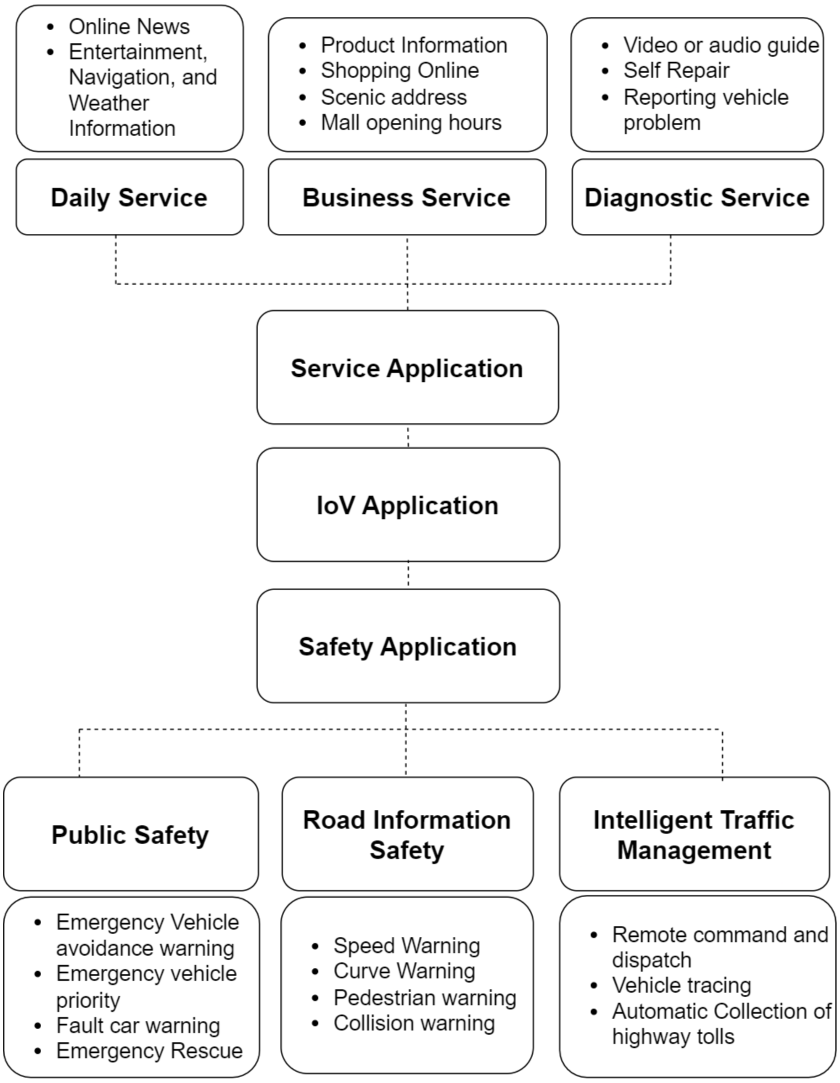


Figure 11. Applications of IoV, adapted from [41]

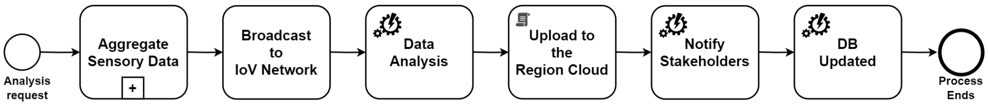


Figure 12. Vehicle-To-Sensor Communication As-Is Process Diagram

3.3.2 Components of V2S communication

Several components work together to enable seamless real-time data exchange between vehicles and the IoV network in V2S communication. Here are the key components involved in the V2S communication process:

- ***Vehicle & Onboard Sensors***: Vehicles are central components in the IoV ecosystem. It is equipped with communication modules, onboard sensors, and computing capabilities. They play a central role in the V2S communication scenario.

- ***Onboard Sensors***: Sensors like GPS receivers, accelerometers, cameras, radar, LIDAR, temperature sensors, and so on. The sensory data collected in real-time are aggregated by onboard units (OBUs) are transmitted to the region cloud.

- ***Sensor Data Transmission***: Sensory data accumulated from sensors wirelessly by the OBU in vehicles through communication protocols and technologies. This includes broadcasting data over a specific frequency or establishing channels between sensors and vehicles. These data are transmitted after initial analysis to the region cloud, which involves filtering and analyzing for anomalies to extract meaningful insights and make informed decisions.

- ***Roadside Units***: Roadside Units (RSUs) are one of how sensory data is broadcasted to the region cloud. Considering the high mobility of the vehicles and the effective range of the RSU, vehicles might not be able to complete the broadcast of data using the RSU as it goes out of range.

- ***Wireless Communication Module***: Wireless communication module allows vehicles to establish communication with the region cloud. This module utilizes communication technologies such as Wi-Fi, cellular networks, or dedicated short-range communication (DSRC) to transmit and receive sensor data.

3.4 Proposed Blockchain-based DT Solution

The proposed V2S communication process enhanced with blockchain-based DT can be divided into two parts. Figure 13 shows the authorization verification of the vehicle and broadcasting analysis request to the IoV region cloud. The main layers of a basic IoV are the Perception layer, Transport Layer, and Data Analysis layer [58]. The proposed blockchain-based DT solution aims to secure information exchange between sensors and the vehicle and thus enhance security for services required by the vehicles in an IoV region, maintaining road safety and infotainment, and so on.

Figure 14 shows an already verified analyzer's subscription of the analysis request in the region and data analysis process. Here, only the entities involved in V2S communi-

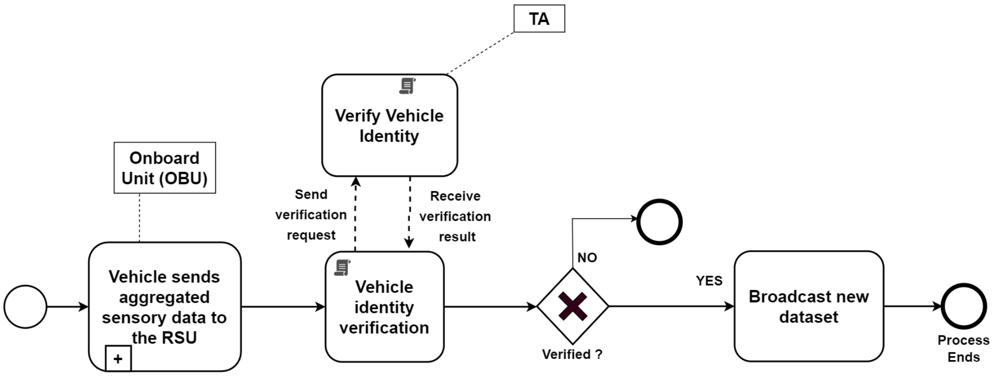

TA
Verify Vehicle Identity
Onboard Unit (OBU)
Send verification request
Receive verification result
Vehicle sends aggregated sensory data to the RSU
+
Vehicle identity verification
NO
YES
Verified ?
Broadcast new dataset
Process Ends

Figure 13. Vehicle-To-Sensor Communication - Authorize and Broadcast

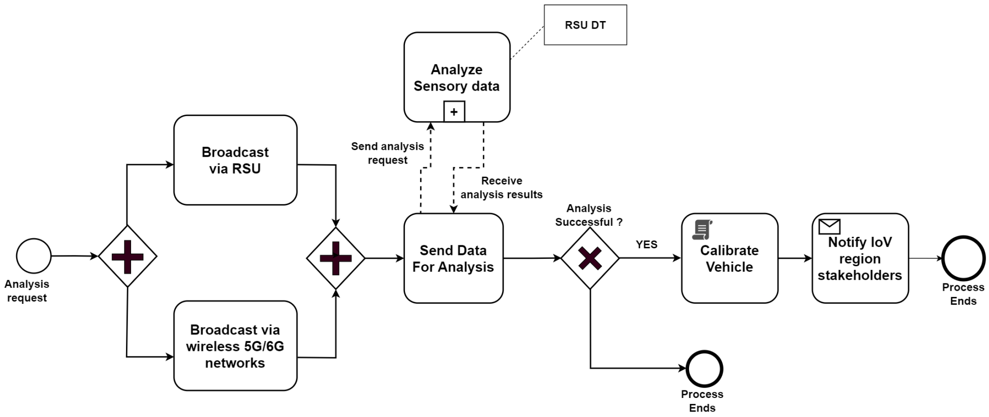


Figure 14. Vehicle-To-Sensor Communication - Subscription to request and Data analysis

cation are presented. The other entities for other communication are assumed to be not affected. There are six entities involved in the blockchain-based DT model:

- ***Trusted Authority (TA)***: A fully trusted third party responsible for registering vehicles and revoking malicious vehicles that are flagged by the incentivized vehicles through the RSU of the region. The cars are flagged as malicious when the data verification in the analysis phase finds inconsistent data reported by the new vehicle. In addition, it deploys and maintains the smart contracts used for vehicle registration and revocation. It is assumed that TA is fully trusted in the system with the real identity of vehicles and has high computation and communication power. TA is responsible for the overall management of the IoV Region Cloud.

- ***Vehicles***: A vehicle in this system is an authorized entity equipped with an Onboard Unit (OBU), aggregating sensory data and communicating to the IoV region cloud governed by the TA to update its DT with only changed information. The vehicles communicate with the regional IoV Cloud via the RSU or 5G wireless networks. Transmitting only the changed information rather than re-synchronizing all the information saves bandwidth and reduces latency. Once a vehicle is flagged malicious in the analysis phase, TA initializes the smart contract to revoke its authorization from the region cloud. All incoming requests from the flagged vehicle will be discarded.

- ***Road Side Unit (RSU)***: In the proposed system, an RSU communicates with the vehicle directly via radio coverage and forwards messages to the region cloud via its digital twin in the cloud. RSUs also have a blockchain-based DT for them so that they do not directly possess threat elements to the system. The RSU-DT(s) are responsible for verifying the vehicles' legitimacy and the assignment or removal of authorization to the vehicles. They also shared regional secrets to vehicles that passed verification.

- ***Vehicle DT***: A digital twin of the vehicle in real life, the DT is always synchronized in real-time after data analysis. They are responsible for publishing the analysis job to the incentivized DTs of other vehicles in the cloud. They maintain the information and have access to the blockchain and add data to the blockchain via a smart contract governed by the RSU-DT.

- ***Blockchain***: Blockchain is used in the proposed systems as a fully trusted, decentralized, and distributed ledger. The values stored in the blockchain are the vehicle DT's public key, the analyzer DT's public key, and the Merkle tree root. The costlier data like the timestamp, data from sensors, unique identifiers of vehicles, and analyzer are stored on the InterPlanetary File System (IPFS). The IPFS hash is also stored in the blockchain. By deploying and invoking smart contracts, the incentivized DTs in the region cloud access and use the historical sensory data in the analysis phase.

- ***IPFS Storage***: IPFS stores details of the DTs and large transaction logs with historical sensory data for reliability, accessibility, and integrity of the stored data. The cost of storing on IPFS is very minimal when compared with storing on-chain [37]. IPFS hash is stored on-chain and on the smart contract governed by the TA. Any malicious attempt to modify the information stored on the IPFS would generate a new hash for the operation, which will not match the one stored in the smart contract. So IPFS is fully trusted in the proposed system.

3.5 Sensory Data Communication and Analysis Process

The proposed solution goes through phases: Initialization, Authorization, DT creation & Synchronization, Broadcasting & Verification, Storage, and Revocation. Figure 15 shows the V2S communication scenario and the entities involved in the proposed solution.

- ***Initialization*:** In this phase, the prerequisites of setting up the IoV region cloud are to be done. It is assumed, as mentioned earlier, that the TA is a fully trusted authority throughout the phases and the responsible entity for this phase. TA chooses the appropriate Elliptic Curve Cryptography (ECC) and the master secret S_i for this region cloud is generated along with the public key ($PubK_{RSUi}$) and hash function. TA builds the Blockchain-based DT for the RSU ($RSUDT_i$) belonging to the region. TA also builds the consortium chain and deploys smart contracts for authentication to the chain and stores its public key on the chain. During this phase, the $RSUDT_i$ are automatically loaded with these parameters $S_i, PubK_{RSUi}$ upon creation.

- ***Authorization*:** When a vehicle, let's say V_i enters a region of an RSU, RSU_i communicates with its DT $RSUDT_i$, to get the information about the vehicle from the TA to check the legitimacy of the vehicle. In other words, it gets the real identity of the V_i i.e., the $UUID_i$. Upon successful verification of the V_i, the $RSUDT_i$ will publish an insertion transaction on the blockchain adding a pseudonym PS_{V_i} and a public key $PubK_{V_i}$. A notification message with a signature of the RSU_i is sent to the V_i for the successful join. The message contains a signature σ_{RSUi} using the private key of RSU_i. This eliminates the need for redundant verification of the vehicle V_i when it again reaches the region of the RSU_i.

- ***DT creation & Synchronization*:** After receiving the success message from the RSU, a blockchain-based DT is created for the vehicle VDT_i. The vehicle V_i, upon gaining access to the cloud, encrypts the aggregated sensory data and sends it to the region cloud via one of the two ways: **(1)** Using the RSU_i, **(2)** Using wireless *5G* network. The vehicle only communicates the updated information to the cloud in real time. The transmission of changed information instead of re-synchronizing all the information saves bandwidth and reduces latency. Considering the movement of the vehicles in IoV, there can be disruptions in the process of synchronization of the VDT_i using the RSU_i when the vehicle V_i goes out of the region of the RSU_i. As a result, the process continues via wireless *5G* network.

- ***Broadcasting & Verification*:** The previous two phases authenticate a vehicle and authorize it to broadcast changing sensory data to the IoV region cloud. To broadcast the data, a vehicle DT, when synchronized with real-time data, generates a temporary anonymous identity and private key, which are used to generate signatures for the aggregated data. A vehicle DT VDT_i, requests for verification to DTs of nearby vehicles present in the region cloud. After the vehicle DT, say VDT_i obtains regional secret through authentication, it can generate a request signature $[Req_{VDTi} = AID_{VDTi}, T_{VDTi}, \sigma_{VDTi}, D_{VDTi}, Reward_{VDTi}]$. AID_{VDTi} represents the anonymous identity of VDT_i, σ_{VDTi} represents request signature, T_{VDTi} is the timestamp, $Reward_{VDTi}$ is the reward for processing the request and D_{VDTi} is the data from VDT_i. The other vehicles become service providers and pick the Req_i analysis request. A free vehicle DT, say VDT_j receives the task and verifies the signature of the request Req_i with the help of RSU_i 's public key ($PubK_{RSUi}$). If it fails to verify, the Req_i is discarded.

- ***Analysis & Storage*****:** Once verified, VDT_j matches the new data D_i from Req_{VDTi} with the average of its last three recorded data. If matching data is found within a specified time in the smart contract, VDT_j sends a success message signed using its private key $PriK_{VDTj}$ to the VDT_i. Upon receiving the analysis success message, VDT_i publishes an insertion transaction on the blockchain, adding a pseudonym PS_{V_i} and a public key $PubK_{V_i}$. The block contains the current DT state, $T_{Req_{VDTi}}$, and IPFS hash. The IPFS stores the costly sensory data D_{VDT_i} , and requests information ($UUID_{V_i}$, Req_{VDTi}, $UUID_{V_j}$).

- ***Revocation*****:** When mendacious data is found as a result of the analysis phase, the service provider DT VDT_j, sends a signed failure message to the RSU_i. The RSU verifies if the message is from a valid vehicle, and indeed, the reported data from the source DT is malicious, it informs the TA. After receiving the message, TA first uses the system master private key to recover V_i's pseudonymous identity, then queries the vehicle list to obtain the V_i's real identity $UUID_{V_i}$ and flags it as malicious and adds it to the revocation list. Next, TA will publish a revocation transaction on the blockchain, which removes the mapping of V_i's mapping of pseudonym PS_{V_i} and public key $PubK_{V_i}$ by invoking the smart contract. As the transaction is successfully submitted to the blockchain, TA instructs the RSU (where V_i belongs to) to regenerate the regional secret value and update it to all the nodes in the cloud.

Moreover, the notations used in the phases are given in Table 4. Only the notations in the proposed blockchain-based DT-enabled V2S communication scenario are illustrated here.

3.6 Summary

In this section, we have an overview of IoV in general, the components involved, benefits in real life, and current applications of the concept of IoV. This section aids in answering our research question **RQ1: What are the building blocks of IoV?**. Moreover, the

Table 4. Notations and their explanation.

Notations	**Explanation**
V_i	(*i*th) New vehicle entering region cloud.
VDT_i	(*i*th) New vehicle digital twin.
RSU_i	(*i*th) Roadside Unit in the region cloud.
RSU_{DT_i}	(*i*th)Digital twin of the roadside unit RSU_i in the region cloud.
V_j	(*j*th) Analyzer vehicle in the region cloud.
VDT_j	(*j*th) Analyzer vehicle digital twin.
S_i	(*i*th) Master secret of the region cloud.
$UUID_i$	Real unique identifier of the new vehicle.
$PubK_{V_i}$	Public key of the vehicle (V_i).
$PubK_{RSU_i}$	Public key of the roadside unit (RSU_i).
PS_{V_i}	Pseudonym of the vehicle (V_i).
Req_{VDT_i}	Analysis request of sensory data from the vehicle (V_i).
AID_{VDT_i}	Anonymous identity of the new vehicle digital twin (V_{DT_i}).
D_{VDT_i}	Sensory data contained in the analysis request from the new vehicle digital twin (V_{DT_i}).
T_{VDT_i}	Timestamp of the analysis request from the new vehicle digital twin (V_{DT_i}).
$Reward_{VDT_i}$	Reward associated with the analysis request from the new vehicle digital twin (V_{DT_i}).
σ_{VDT_i}	Reward associated with the analysis request from the new vehicle digital twin (V_{DT_i}).
$UUID_j$	Real unique identifier of the analyzer.
$PriK_{VDT_j}$	Private key of the analyzer digital twin.

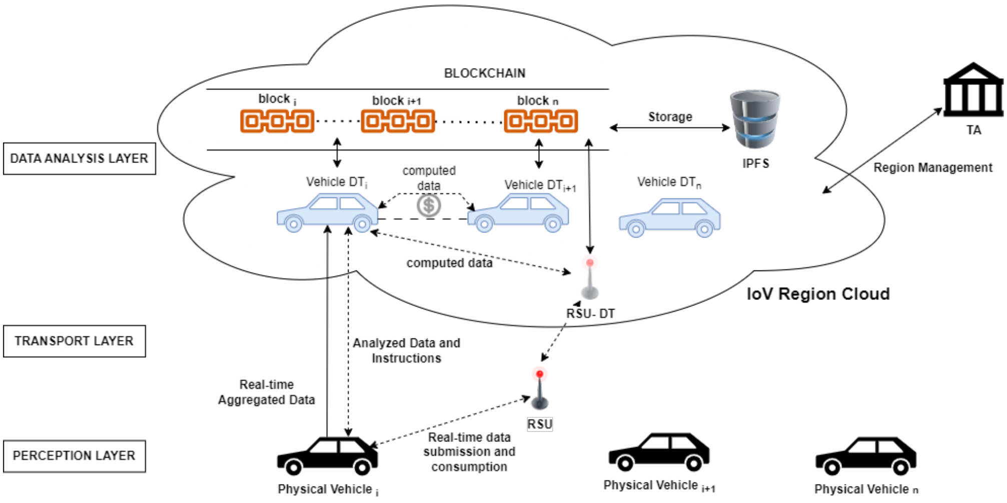


Figure 15. Secure sensory data communication in IoV using Blockchain-based DT

need for research in the security of data exchanges between different components of IoV is also discussed here. In this section, we produced artifacts denoting the research guideline **DG6. Design as a search process**. This section also clarifies to the reader why this research was necessary and why in this thesis, we have focused on the V2S communication in IoV as our use case in accordance with **DG2. Problem Relevance** in the design guidelines of our research method mentioned in Table 1.

4 Literature Review

Modern-day DTs are complex systems comprising many interconnected virtual components sharing communication threads. They present an exact copy of their physical counterparts. The digital models analyze data received from the physical layer. Recent research has shown that confidentiality of Data has the most impact on DT systems [43]. The most essential features of Blockchain technology can act as a safeguard against the security issues faced in the industry. Moreover, integrating DTs with blockchain in industrial IoT has recently gained popularity among researchers. In lieu of that, in this section of the thesis, an SLR to find existing blockchain-based DT solutions for Industry 4.0 (namely Cyper-physical Systems (CPS), Industrial Internet of Things (IIoT), and Internet of Vehicles (IoV)) was done. The main objective of this review was to identify existing blockchain-based DT solutions where the application of DT was to aid in enabling security i.e., an extra layer of security for industrial systems in Industry 4.0. The design research guideline **DG2. Problem relevance**, it conforms that for research to be effective, it is essential to know the state-of-the-art practices in the relevant research area. The SLR produced in this section acts as an artifact for the guideline.

4.1 Research Questions

In the review, we address research question **RQ2: What is the state-of-the-art in securing industrial systems using blockchain-based DT?**. To better grasp the solutions provided in reviewed papers, we further break down the question into sub-research questions. The sub-research questions are :

- **RQ2.1**: What are the security threats in industrial systems?
- **RQ2.2**: What security threats are mitigated using blockchain-based DTs?

The first sub-research question **[RQ2.1]** aims at finding the security issues of Industry 4.0, and papers try to solve the security issue. To limit the scope of the thesis, we limit our search to only CPS, IIoT, and IoV. The second sub-research question **[RQ2.2]** points out the threats covered in the research to be mitigated using their proposed solution using blockchain-based DTs.

4.2 Search Strings

The search terms used are the following: "*(blockchain-based Digital Twin) OR (Blockchain enabled DT) OR (Blockchain Enhanced Digital Twins) AND (Industry 4.0) AND (Security)*". The search terms *blockchain-based Digital Twin*, *Blockchain enabled DT*, and *Blockchain Enhanced Digital Twins* are synonymous phrases and can be used alternatively, but it was observed that the papers use any one of them. A large string *"Blockchain-based digital twins for industry 4.0 security"* was also used but similar results were found and hence not considered. The terms *Industry 4.0* and *Security* were used to narrow down the results. Papers were initially selected if the search terms were present in the title or abstract of the paper.

4.3 Data Sources

The initial search for relevant papers was done through the IEEE digital library, ScienceDirect, and ACM digital library. The reason behind choosing these three databases is the fact the recent state-of-the-art for technological papers has been these three. Additional relevant papers were identified from the related work sections and citations of the papers identified in the initial search. Since the concept of integrating blockchain and DTs is a relatively new paradigm, some grey literature was also considered, but it was carefully made sure that the relevant results matched the inclusion criteria.

4.4 Study Selection and Quality Assessment

For the review, we wanted to search the most recent literary items related to blockchain-based DTs and hence selected items in the range of years 2018-2023. The main focus of the review was to find papers that explicitly mentioned blockchain-based DTs and their use as security enablers or enhancers. Moreover, Early access papers or book sections were also excluded as the scope of the review was to find concrete reviewed research targeting the inclusion criteria.

Inclusion Criteria

- IC1: Literature related to blockchain-based digital twins
- IC2: Literature discussing the usage of blockchain-based DTs as a security enabler
- IC3: Literature using blockchain-based DTs in CPS, IIoT or IoV.

Exclusion Criteria

- EC1: Literary items published before 2018 (592)
- EC2: Literature that is not journals, magazines, or conference papers (442)
- EC3: Literature without explicit mention of Blockchain-based DTs (217)
- EC4: Literature that is tagged Early Access (107)

4.5 Papers Selection

The digital libraries were first searched using the aforementioned search items which expectantly yielded a lot of results but all were not relevant. After EC1 was applied, it yielded 592 results. Moreover, EC2 included only conference material, magazines, and academic journals that were considered for further processing, resulting in 442 papers. Furthermore, with the application of EC3 and EC4, left me with only 107 papers.

Out of these 107 resulting papers, each paper was evaluated manually to fulfill inclusion criteria. The inclusion criteria were applied by analyzing the introduction and abstracts. Finally, 11 papers were included for further study. Additionally, snowballing and references from previously included literature yielded 6 papers which were manually checked for relevance and included in the scope of the review.

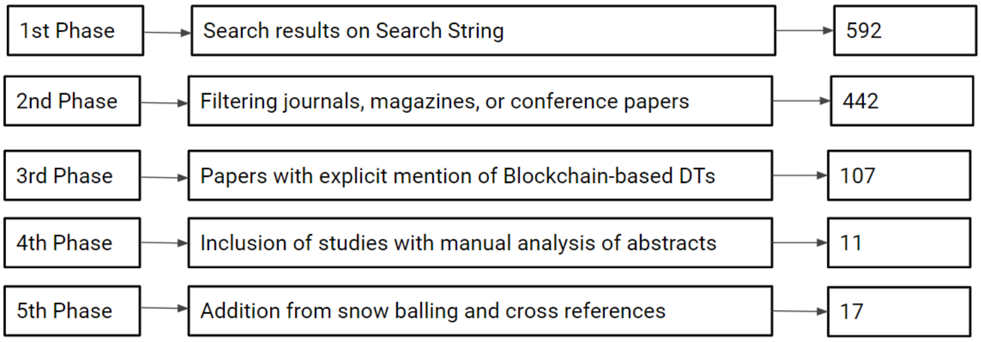

1st Phase
Search results on Search String
592
2nd Phase
Filtering journals, magazines, or conference papers
442
3rd Phase
Papers with explicit mention of Blockchain-based DTs
107
4th Phase
Inclusion of studies with manual analysis of abstracts
11
5th Phase
Addition from snow balling and cross references
17

Figure 16. Papers selection process

4.6 Data Extraction Strategy

Seventeen papers were read through to gather data for the research questions. The data found in the research papers which could answer the research question is noted in a table for further analysis. The data extraction table (as Table 5 shows) consists of the data item as Research Work, Application, Objective, Contribution, Limitation, Tool, and Threats Covered. These columns were chosen in accordance with the RQs. In this thesis, papers were analyzed to find security loopholes or issues in industrial systems (namely CPS, IIOT, and IOV) and what security threats were mitigated using blockchain-based DTs.

Table 5. Data Extraction form.

Data Item	**Value**
Research work, Year	Citations and Year of publication
Application	Target application area of the research
Objective, Contribution, Limitation	The main objective, contribution and limitations of the depicted solution in the research
Threats in industrial systems	Threats discussed in the literature
Threats Mitigated	Threats mitigated using proposed blockchain-based solution

4.7 Presentation of Results

In this section, the review outcomes are presented in the form of a table. The Table 7 contains data aiming at answering the RQs as depicted in Section 4.6. The main aim of the review was to find the existing research where blockchain-based DTs have been used as a security enabler in industrial systems. Firstly, the papers are categorized according to their application area (as Table 6 shows). Out of the 17 papers, five were related to CPS, four were related to IIoT, and the rest discussed issues related to IoV.

Table 6. Application area of research

Application	Studies
CPS	[70] [22] [13] [69] [9]
IIoT	[60] [67] [48] [74] [56] [23] [4]
IoV	[46] [71] [28] [45] [59]

Secondly, 7 out of 17 papers contributed to designing a blockchain-based DT secure communication framework. Most of the papers deal with the security of industrial systems. Some of the frameworks enhance data collaboration between multiple DTs. Novel architecture based on blockchain-based DTs was also the core contribution of other 7 out of 17 papers. And the rest of the papers (3 out of 17) presented the creation of DT architecture with blockchain integration. These solutions enhance the overall security of CPS, IIoT, and IoV systems.

However, a large share of the papers (8 out of 17) lacked implementation and did not describe the practicality of their deduced conceptual model, architecture, or framework. Out of the remaining nine papers, the paper [4] did not include any particular information about the limitation of their solution. [74], [56], and [71] denoted the efficiency of smart contract and consensus algorithm used in their blockchain-based solution as the limitation. Latency and some privacy leakage were mentioned as limitations in papers [46], [60], [45], [9]. Detection of the already affected node is a limitation in the blockchain-based DT model DT2SA model illustrated in [23].

[RQ 2.1]: What are the security threats in industrial systems?

We aim to find security threats and vulnerabilities of industrial systems that were presented in the papers. It was found that many papers discussed multiple security threats. 6 out of 17 papers presented Data Tampering, Spoofing as security threats to industrial systems. However, the authors in [46] included Sybil Attack, [67] included DDos, and [23] included APT as a vulnerability. Network Rerouting, MITM, and DDoS were discussed in 12 out of the 17 papers. The authors in [59] and [60] also included Network security, Remote Spying, and Eavesdropping as vulnerabilities. Wireless connection networks lead to remote spying and eavesdropping in network activity which is another vulnerability mentioned in papers [60], [70], and [28].

[RQ 2.2]: What security threats are mitigated using blockchain-based DTs?

Data Tampering and Spoofing are the primary threats that were mitigated in 6 out of 17 papers with their solution using blockchain-based DTs. The typical security risks MITM and DDos associated with blockchain-based solutions have been covered in 15 papers. Moreover, The authors in [46] also covered Sybil Attack as one of the vulnerabilities. 3 out of 17 papers discussed advanced persistent threats(APT) like Stuxnet. The rest of the papers also tackled issues like low latency, high cost, availability, and interoperability as passive attacks due to other attacks.

4.8 Summary

The SLR process and results were presented in this section, addressing the research question. Based on the answers to the results of research questions of the literature review found in Section 4.7, we conduct our further steps in this thesis. Out of the application areas considered for the review, it is evident that there exists the least amount of research in the field of IoV in the context of security using blockchain-based DTs. Although the concept of IoV is not new, it is also clear that there is a lack of uniformity of standards currently practiced in the industry to enable robust V2V communication and V2S data integrity, reliability, security, and interoperability. In lieu of that, this thesis aims to provide a novel conceptual framework using blockchain-based DTs to enable the secure data exchange and decision-making process in IoV. In later sections, further elaboration of the concept is illustrated. As we have used design science research methodology, for the contribution of this thesis to be viable, it had to be aligned with the state-of-the-art. Hence, this SLR was conducted, which was a guideline for our proposed solution in this thesis.

5 Security Risk Assessment

This section presents the security risk analysis of our case study regarding the V2S communication scenario of IoV. In this section, we answer the research question **RQ3: What are the security risks in industrial systems, and how can blockchain-based DTs be used to mitigate them?**. Before going to the case study-specific analysis, we present the security risk analysis of IoV to gain a broader perspective. First, the attack vendors involved in the context of IoV are considered and illustrated using the STRIDE threat model [39]. Next, how blockchain-based digital twins may mitigate risks and protect business assets is briefly described using the ISSRM domain model [6].

Moreover, the general perspective about IoV gained, we indulged into a more detailed case-specific threat model and risk management process for the V2S communication scenario is explained. Threat modeling is a proactive way to identify, enumerate, and prioritize threats, thus helping to take appropriate safeguards against threats. In information security, attacks and threats can be classified into six main categories in STRIDE Threat Model [39], including spoofing identity, tampering with data, repudiation, information

Table 7. DT as security-enhancing use case: A summary.

Research, Year	***Area***	*⋆ **Objective*** *○ **Contribution*** *◐ **Limitations***	***Threats***	***Threats Mitigated***
[46],2022	IoV	⋆ Secure transport system ○ BC-Based IoV Secure Communication Framework ◐ User privacy protection	DDoS Sybil Attack Spoofing	Sybil Attack Disinformation
[60],2022	IIoT	⋆ Early detection of Bot formation in smart factory environment ○ Blockchain-enabled Secure Digital Twin Framework for early botnet-behavior-detection ◐ IP tracing method need to be developed	DDoS Spying Eavesdropping	MITM DDoS
[70],2022	CPS	⋆ Trusted Twins for Securing Cyber-Physical Systems (TTS-CPS) ○ Trustworthy specification-based DTs creation ◐ Practicality of the concept	APT Spoofing Sybil Attack Remote Spying	APT (Stuxnet) MITM
[22].2019	CPS	⋆ Strengthen the security of CPSs ○ Technical Usecases for DT concept in CPS security ◐ Lacks implementation details	Malware Disinformation MITM	Disinformation MITM
[67], 2021	IIoT	⋆ Reliability of DT data sources ○ Blockchain-enabled Secure Digital Twin for tack and trace providence data. ◐ No Implementation	Disinformation DDoS	DDoS
[71], 2021	IoV	⋆ Track & Secure Resource sharing between DTs in IoV ○ DT communication framework using BCT, Smart contract incentive ◐ Optimum smart contract	Rerouting	Rerouting
[13], 2020	CPS	⋆ Digital Twin for risk prediction and prevention ○ Risk analysis reference DT model ◐ Lack of Practical implementation	Jamming DDoS MITM	MITM
[48], 2020	IIoT	⋆ Secure and reliable collaborative cloud computing ○ BCT based DT wireless network ◐ Lacks implementation	Tampering MITM	MITM

[28], 2021	IoV	⋆ Mapping real world into virtual space ○ BCT-based DT model for VANE ◐ Lacks implementation	Tampering Latency	High-latency Tampering
[69], 2021	CPS	⋆ Intelligent DT using Blockchain ○ Framework designed for DTs security using Blockchain and AI ◐ Practicality of the conceptual framework	MITM APT Disinformation	MITM Stuxnet
[74], 2020	IIoT	⋆ A secure, traceable, and decentralized IMS ○ Novel manufacturing blockchain of things (MB-CoT) platform ◐ Latency performance and inefficient consensus algorithm	Latency DDoS Malware	DDoS
[56], 2022	IIoT	⋆ Secure sensing and actuation in IoT systems ○ Smart contract-based digital twins architecture ◐ Lacks Implementation and Performance Evaluation	Disinformation Malware MITM	MITM Disinformation
[45], 2021	IoV	⋆ Secure and on-demand DT services for future ITS ○ DTaaS architecture for the various DT requirements in ITS ◐ Higher Energy cost, Implementation in IoV	MITM Tampering	MITM DDoS
[9], 2022	CPS	⋆ Secure base operations of CPSs ○ BCPS (Blockchain-enabled CPS) framework ◐ Garbage-in-and-garbage-out issue, Dubious data detection	Spoofing, Tampering, Disinformation	Latency, MITM
[59], 2021	IoV	⋆ Improve DTs collaboration and analysis data in real-time ○ Concrete ledger-based collaborative DTs framework ◐ Lacks implementation	DoS MITM Spoofing	DDoS MITM

[23], 2023	IIoT	⋆ Security analytics for threat detection and mitigation ○ DT2SA model with Twinsight for cybersecurity ◐ Only intrusion detection	APT DDoS MITM	APT Disinformation
[4], 2022	IIoT	⋆ Security and data privacy in IIoT ○ Blockchain-based Proof of Authority (PoA) trust mechanism ◐ N/A	Fidelity Latency Spoofing	Transmission delay MITM

disclosure, denial of service, and elevation of privilege. More details about STRIDE and ISSRM can be found in Sections 2.5 and 2.6, respectively.

5.1 Threat Model for Internet of Vehicles

IoV systems may get attacked by various attack vendors through different methods like jamming, interference, eavesdropping, masquerading, etc. It decreases the stability, robustness, real-time availability, security, and privacy of IoV. As a result, IoV loses the ability to provide effective services, even causing serious accidents, mainly due to the characteristics like dynamic topology, bandwidth limitations, transmission power limitations, abundant resources, mobile limitation, nonuniform distribution of nodes, perception of data depending on the vehicle trajectory, and large-scale network. This section presents the threat model of attacks in IoV. Figure 17 shows the classification of threats according to the STRIDE model for IoV.

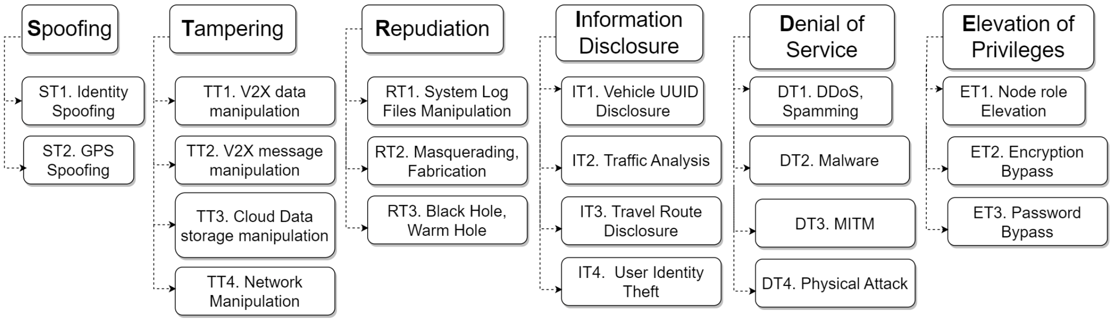


Figure 17. Threat Model for Internet of Vehicles

In terms of IoV, *Spoofing* attacks can refer to identifying as legitimate RSUs, Vehicles, OBUs, Control Units, and so on **(ST1)**. Vehicle geolocation (GPS) **(ST2)** spoofing can also fit into this category. In the transport layer, V2C data manipulation, malicious messages to the network, and vehicle data manipulation **(TT1, TT2, TT3, TT4)** in the cloud are classified as ***Tampering*** attacks and if goes unnoticed, may lead to drastic accidents. Non-repudiation of IoV data is another crucial aspect of ensuring security and privacy in the system. Third-party cloud services used in IoV lead to ***Repudiation*** attacks like log files manipulation, Masquerading, and Fabrication, which affect the services in IoV **(RT1, RT2)**. Black Hole, Warm Hole, and also Grey Holes are also prominent attacks and are harder to detect in traditional IoV systems **(RT3)**. Malevolent architectural components in IoV may lead to eavesdropping, vehicular information leaking, user privacy, and travel traits disclosure **(IT1, IT2, IT3, IT4)**. These attacks can fit into ***Information disclosure*** threats. ***Denial of Service*** is the most common attack category in IoV systems. DDos, MITM, Network spamming, and rerouting are exploited by adversaries to affect IoV systems **(DT1, DT2, DT3)**. Physical attacks damaging the physical devices create the absence of nodes leading to a denial of service attack **(DT4)**. As malicious nodes get access to IoV systems, they may escalate privileges to gain insights into the protected parts of the IoV system and the overall operation of the system. They can bypass encryption and passwords to access sensitive information **(ET1, ET2, ET3)**. This array of attacks can fit into ***Elevation of Privileges*** threat category.

Apart from the above-mentioned threats, an IoV system may be subject to other risks, which can be passive attacks as a result of the mentioned threats. There also exist social drawbacks like trust in the IoV system, the willingness of vehicle owners to share information, and so on, which may affect the realization of IoV.

5.2 Security Risk Management using Blockchain-based IoV DT

Once a threat model is defined, security risks for IoV can be derived. The process of deriving risks from the threat model is iterative, and the process described in this section should be repeated for each threat in the model mentioned in Section 5.1.

5.2.1 Security Risk model

Firstly, the vulnerability on which the threat is based is distinguished. Then, IoV system assets that might have such a vulnerability are identified. The threat targeting this vulnerability is instantiated if such a system with the vulnerability is found. As the next step, how this threat can exploit such a vulnerability is defined using answers to the following questions: ***(i)*** which business assets are supported by the analyzed system asset; ***(ii)*** which security criteria are negated; ***(iii)*** what are the impacts of the threat implementation. Repeating the mentioned procedure, the analysis resulted in 8 security risks that have the most impact and most common to occur in the IoV system and impact its business assets. The complete model can be found in Appendix B.

5.2.2 Security Risk Mitigated in Blockchain-based IoV DT

Once security risks have been identified using ISSRM, the vulnerabilities in an IoV system and the impact of the risks become clear for mitigation. Countermeasures using blockchain-based DTs to mitigate the risks are listed in Table 8.

5.3 Threat Model for V2S Communication

Figure 18 illustrates the derived threat model for the Vehicle–to-Sensor communication scenario in IoV. The model encompasses 21 potential vulnerabilities that an adversary could leverage throughout the V2S communication scenario which are categorized into six distinct groups of threats using the STRIDE approach.

Spoofing attacks can refer to falsely identifying as legitimate RSUs, Vehicles, OBUs, Control Units, and so on **(ST1, ST2)**. Sensory data spoofing **(ST3)** is one of the biggest concerns in the V2S scenario. Misinformation may lead to drastic results as other components in the IoV network need these data. Sybil attacks **(ST4)** where malicious entities constantly manipulate the system, which leads to other threats. ***T**ampering* refers to manipulating data and communication messages **(TT1, TT2)** between components involved in the V2S scenario. An attacker might communicate with false data and messages and store malevolent data in the system, which might affect the communication channel and entities **(TT1, TT2, TT3, TT4)**. Blackhole attacks and Wormhole attacks **(RT4)** create artificial hunger for resources which reduces efficiency in the system considering the V2S scenario. Fabricating messages and log files **(RT1, RT2)** stored in the cloud might present the system to be attacked to gain sensitive insights into the traits of vehicles and user interaction with the IoV system **(RT3)**. These are categorized under ***R**epudiation*.

***I**nformation Disclosure* categorizes threats that compromise confidential data by granting access to individuals who are not authorized to have such access **(IT1)**. Local data accessed by adversaries is referred to by this **(IT2)**, spying or in the transmission between systems or their components**(IT3, IT2)**. These attacks lead to a more adverse future vulnerability.

***D**enial of Service* attack refers to a type of cyberattack that seeks to make a computer system or network resource inaccessible to its intended users by overwhelming it with an excessive volume of malicious traffic or requests. In our use case blocking computational resources of the nodes in IoV **(DT1 and DT2)** are considered as DoS attacks. The primary objective of a DoS attack is to disrupt the normal functioning of the target system, causing it to become slow, unresponsive, or entirely unavailable **(DT3)**. Services might be slow to respond resulting in drastic outcomes in real traffic situations.

***E**levation of Privileges* threats refer to allowing an attacker to have authorization permissions that he was not supposed to have, thereby violating the system's authorization. It can be achieved either using the obtained legitimate credentials **(ET1 and ET2)** or by

Table 8. Internet of Vehicle Security Risks Mitigated using Blockchain-based DTs

Security Risks	**Threat**	**Impact**	**Blockchain-based DT Countermeasure**
SR1	**ST1**	Compromised Privacy and Confidentiality of V2X notifications, Integrity of sensory data, and Vehicle details	Authorization requests through Digital Twins of vehicles and distributed ledger of blockchain mitigates unauthorized access and enables traceability of requests.
SR2	**ST2**	Compromised Privacy and Confidentiality of vehicle owner, Integrity of sensory data and traffic details.	Cryptographic assignment of single blocks in blockchain stops other entities to be tampered with and Verification of geolocation allows anomaly detection
TR1	**TT1**	Compromised Privacy and Confidentiality of the system notifications, entities, central authority, Integrity of sensory data and traffic details, vehicle notifications	Physical layer is unaffected as DTs need to be synchronized with **verification**. **RBAC** using blockchain with **End-to-End encryption** of P2P messages mitigates illicit and malicious emission of messages.
TR2	**TT2**	Integrity of data packets communicated from vehicles to RSUs, Control Units, and cloud storage.	**RBAC** and **Immutability** in blockchain ledgers allow only legitimate Software entities to update the firmware.
RR1	**RT1**	Compromises integrity of Vehicle Details, Confidentiality of System Activities	**Distributed**, **Virtua**l, **traceable**, and **immutable** ledger in blockchain-based DTs. **Encrypted** key authorization allows only **role-specific** traversal through system files and network.
IR1	**IT1**	Compromises Confidentiality vehicle users, vehicles, traffic details	**RBAC** and **ABAC** using blockchain allows vehicles with correct **CA** to connect to IoV Network. **Hash functions** to store traffic data.

DR1	**DT1**	Compromises Availability of IoV Services, Confidentiality of Communication messages	**Group verification** by peer vehicles only allows legitimate vehicles to join the Network. **Trusted central authorities** provide **cryptographic certificates** stored in the blockchain.
DR2	**DT2**	Compromised Privacy and Confidentiality of the system notifications and messages, Integrity of sensory data and traffic details, vehicle notifications	**Decentralized**, **distributed** nodes in blockchain handle a large number of nodes. As the access to **DT layer** is given, other physical assets in the system are protected.
DR3	**DT3**	Compromises Availability of resources for services needed by vehicles	**Smart contracts** with timestamps allow automated freeing up of computing resources. **RBAC** allows authorized nodes to utilize allocated resources.
ER1	**ET1**	Compromises integrity of Vehicle Validation with an intention to compromise security criteria of other assets that use Vehicle Validation as a data source	**DT layer** sits as middleware between communication channels. Authorization by **Public key, Private key pair** allow only legitimate entities to communicate between them

a more sophisticated Advanced Persistent Threats (APT) on the existing authentication mechanisms (ET2 and ET3).

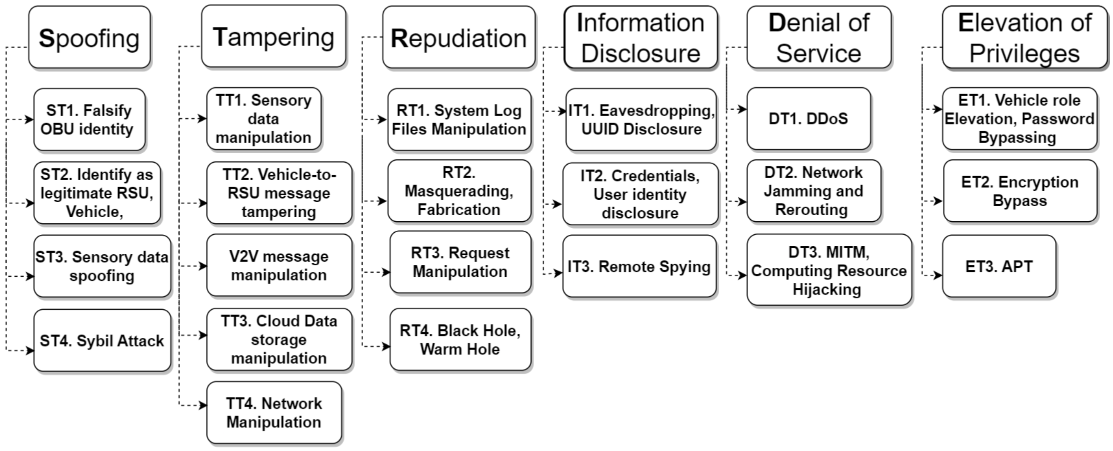

Spoofing
ST1. Falsify OBU identity
ST2. Identify as legitimate RSU, Vehicle,
ST3. Sensory data spoofing
ST4. Sybil Attack
Tampering
TT1. Sensory data manipulation
TT2. Vehicle-to-RSU message tampering
V2V message manipulation
TT3. Cloud Data storage manipulation
TT4. Network Manipulation
Repudiation
RT1. System Log Files Manipulation
RT2. Masquerading, Fabrication
RT3. Request Manipulation
RT4. Black Hole, Warm Hole
Information Disclosure
IT1. Eavesdropping, UUID Disclosure
IT2. Credentials, User identity disclosure
IT3. Remote Spying
Denial of Service
DT1. DDoS
DT2. Network Jamming and Rerouting
DT3. MITM, Computing Resource Hijacking
Elevation of Privileges
ET1. Vehicle role Elevation, Password Bypassing
ET2. Encryption Bypass
ET3. APT

Figure 18. Threat Model for Vehicle-To-Sensor Communication

5.4 Security Threats Mitigated Using Blockchain-based DT

As mentioned above, security threats must be mitigated for secure V2S communication and the proper flow of sensory data in the IoV system. Communication is one of the most crucial processes in the IoV system, as these data are crucial for services provided by IoV. First, the assets which need to be protected for Confidentiality, Integrity, and Availability are defined. In this thesis, the assets involved in the V2S communication are considered and other business assets are assumed to be unaffected. Then, how the proposed blockchain-based DT solution assesses and mitigates the security risks is illustrated.

5.4.1 Protected Assets in V2S Communication Scenario

In the context of V2S communication, the initial step involves identifying and delineating the assets that necessitate safeguarding. *Business assets* (BA) are represented through the transmission of sensory data from the vehicle to the IoV system, which plays a pivotal role in the overall flow of the processes. Supporting these business assets are system assets, responsible for generating, manipulating, and storing new business assets.

The core *security criteria* governing these business assets revolve around the principles of Confidentiality, Integrity, and Availability (CIA). These criteria serve to characterize and evaluate the security posture of the business assets. It is worth noting that the security requirements of a particular business asset are determined by the corresponding security needs of the system to which it belongs. Table 9 illustrates the security criteria of protected assets in the V2S communication scenario.

Table 9. Security criteria for Protected assets

Protected Assets	Confidentiality	Integrity	Availability
Vehicle Details	✓	✓	✓
Vehicle Authentication Request	✓	✓	✓
Region Cloud Network Initiation	×	✓	✓
Sensory Data Uplaod	✓	✓	✓
Vehicle Digital Twin Synchronization	✓	✓	✓
Vehicle Notification	×	✓	✓
Data Analysis Request	✓	✓	✓

5.4.2 Security Risk Model - V2S Communication

As mentioned in the threat model (in Section 5.3) of V2S communication, 21 threats are defined. Repeating the mentioned procedure in Section 5.2.1, the analysis resulted in 9 security risks with the most impact and common occurrence in the V2S communication scenario and impact its business assets. Along with the risk model illustrated in Section 5.2.1, detailed risk models specific to the V2S communication scenario is illustrated in Appendix C.

5.4.3 Security Risk Mitigated using Blockchain-based IoV's DT

Once security risks have been identified, the vulnerabilities in the V2S communication scenario, along with the impact of the risks, become clear for mitigation. In the realization of the proposed solution Ethereum blockchain is used. Countermeasures using Ethereum blockchain-based DTs to mitigate the risks are listed in Table 10.

5.5 V2S Communication Threat Model with Blockchain-based DT

Introducing DTs into the context of the IoV V2S communication scenario enhances security by encapsulating the physical components by representing them by a virtual layer so that the physical devices do not present themselves as direct attack surfaces. Maintaining decentralization, traceability, and validation of communication processes using blockchain also enhances security and privacy in the studied scenario of V2S

Table 10. Security Risks Mitigated using proposed Blockchain-based DT Solution

Security Risks	**Threat**	**Impact**	**Blockchain-based DT Countermeasure**
SR1	**ST1**	Compromised Privacy and Confidentiality of V2X notifications, Integrity of sensory data, and Vehicle details	Authorization requests through Digital Twins of vehicles. Large transaction details, System logs are stored in **IPFS** through **Consensus**. The proposed solution also has TA which is used as a certificate authority validating vehicles joining network **(Fabric CA)**
SR2	**ST2**	Compromised Privacy and Confidentiality of vehicle details, Integrity of sensory data, and traffic details	Membership service provider **MSP** in HLF uses attribute-based access control
TR1	**TT1**	Compromised Privacy and Confidentiality of the vehicles, RSU, central authority, Integrity of sensory data and traffic details, vehicle notifications	Physical layer is unaffected as DTs need to be synchronized with **verification**. **Channels** in HLF partition the network to allow transaction visibility for stakeholders only. Only the channel members are involved in consensus, while other network members do not see the transactions on the channel.
TR2	**TT2**	Integrity of data packets communicated from vehicles to RSUs, Control Units, and cloud storage.	**ABAC** in HLF and three ordering mechanisms: SOLO, Kafka, and Simplified Byzantine Fault Tolerance (SBFT) to allow peer-to-peer communication
RR1	**RT1**	Compromises integrity of Vehicle Details, Confidentiality of System Activities	**Channels**, **Ordering service** and **MSP** allow secure peer-to-peer communication

RR2	**RT2**	Compromises Availability of resources, Integrity of Data analysis request	Ordering service node, along with Peers, handle a large number of requests nicely. **Encrypted** key authorization allows only **role-specific** resource allocation. **Consensus** allows resource retrieval of idle or affected nodes.
IR1	**IT1**	Compromises Integrity, Confidentiality of vehicle information, request contents	HLF allows **ABAC** using blockchain allows vehicles with correct **CA** to connect to IoV Network.
DR1	**DT1**	Compromises Availability of IoV Services, Confidentiality of Communication messages	**Channels** in HLF partition the network to allow transaction visibility for stakeholders only. Only the channel members are involved in **consensus**, while other network members do not see the transactions on the channel.
DR2	**DT2**	Compromised Privacy and Confidentiality of the system notifications and messages, Integrity of sensory data and traffic details, vehicle notifications	**ABAC** in HLF and three ordering mechanisms: SOLO, Kafka, and Simplified Byzantine Fault Tolerance (SBFT) to allow peer-to-peer communication. As the access to **DT layer** is given, other physical assets in the system are protected.
ER1	**ET1**	Compromises integrity of Vehicle Validation with an intention to compromise security criteria of other assets that use Vehicle Validation as a data source	**DT layer** sits as middleware between communication channels. Authorization by **MSP** allow only legitimate entities to communicate between them

communication. However, these two technologies also intensify the security issues and vulnerabilities of the scenario by using a plethora of external components. These components also become attack vendors targeted by adversaries. Figure 19 highlights the derived threat model for the V2S communication scenario in IoV and new attacks introduced by blockchain-based DTs. They surpass conventional security threat mitigation techniques in terms of mitigating risks but also append 14 more threats into the precarious threat landscape of the V2S communication scenario.

In the category of ***Spoofing*** attacks, the blockchain-based digital twin creation process might introduce malicious software attacks **(ST5)**. The physical to virtual layer mapping might use insecure APIs, enabling adversaries to launch covert attacks **(ST8)**. Primary key manipulation and delegate call attacks targeting blocks and the smart contract might also hamper required functionalities in the communication **(ST6, ST7)**. *Spoofing* attacks like tampering with the digital thread from the vehicle to its DT might infuse malicious data into the IoV system and affect other nodes **(TT5)**. Blockchain-based DTs are also prone to prominent blockchain attacks like 51%, Long-Range, and Nothing-at-stake. These act as means to tamper the consensus of the communication scenario used to verify and authorize vehicles into the system **(TT6)**. Smart contracts automating the broadcasting of sensory data highly depend on timestamp and hence its manipulation allows unverified operation of nodes involved in the communication scenario **(TT7)**. Vehicle DTs need to be synchronized in real-time for successfully mapping the V2S scenario in the virtual layer. So, synchronization from malicious or hampered vehicles affects the integrity of the system **(RT7)**. ***Repudiation*** is also caused by 51%, Longrange, Nothing-at-stake as described before **(RT8)**.

The confidentiality of sensitive data is also violated by malevolent vehicles. The physical location of business assets is also hampered which might lead to physical assault on the premises **(IT4)** under ***Information Disclosure***. Private keys assigned to authorized vehicles might also be leaked by adversaries once they gain access as a result of other attacks **(IT5)**. The creation of blockchain-based DTs for entities involved in the V2S communication scenario is already resource intensive. With attacks like MITM, affected entities might block more resources as false identities leading to transaction flooding, bottlenecks in the network, and so on **(DT4, DT5)**. Attackers with excess privileges might disrupt whole traffics if they are allowed to broadcast misinformation **(ET4)**. Smart contracts used for automating cloud storage and transactions need authorization and privileges to go through. Adversaries with gained privileges might alter contracts and send business-sensitive information to other nodes outside the network **(ET5)**. Table 11 summarizes the introduction of new threats as a result of introducing blockchain and digital twin technologies in the V2S communication scenario.

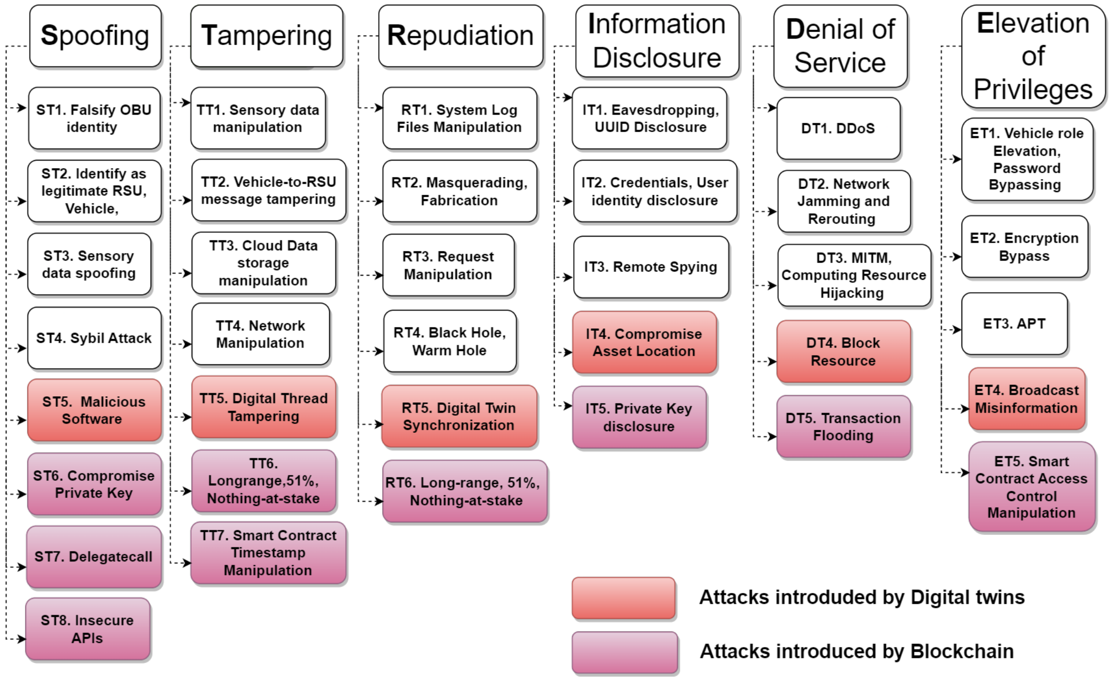

Spoofing
ST1. Falsify OBU identity
ST2. Identify as legitimate RSU, Vehicle,
ST3. Sensory data spoofing
ST4. Sybil Attack
ST5. Malicious Software
ST6. Compromise Private Key
ST7. Delegatecall
ST8. Insecure APIs
Tampering
TT1. Sensory data manipulation
TT2. Vehicle-to-RSU message tampering
TT3. Cloud Data storage manipulation
TT4. Network Manipulation
TT5. Digital Thread Tampering
TT6. Longrange,51%, Nothing-at-stake
TT7. Smart Contract Timestamp Manipulation
Repudiation
RT1. System Log Files Manipulation
RT2. Masquerading, Fabrication
RT3. Request Manipulation
RT4. Black Hole, Warm Hole
RT5. Digital Twin Synchronization
RT6. Long-range, 51%, Nothing-at-stake
Information Disclosure
IT1. Eavesdropping, UUID Disclosure
IT2. Credentials, User identity disclosure
IT3. Remote Spying
IT4. Compromise Asset Location
IT5. Private Key disclosure
Denial of Service
DT1. DDoS
DT2. Network Jamming and Rerouting
DT3. MITM, Computing Resource Hijacking
DT4. Block Resource
DT5. Transaction Flooding
Elevation of Privileges
ET1. Vehicle role Elevation, Password Bypassing
ET2. Encryption Bypass
ET3. APT
ET4. Broadcast Misinformation
ET5. Smart Contract Access Control Manipulation
Attacks introduded by Digital twins
Attacks introduced by Blockchain

Figure 19. Threat Model for Vehicle-To-Sensor Communication with BCT-based DT. The red color denotes threats introduced by DT, purple color denotes threats introduced by Blockchain.

5.6 Security Risk Management for Blockchain-based DT of IoV

The threat model in Section 5.5 illustrates the newly added security threat landscape added to the already precarious threat domain of V2S communication. Although, Hyperledger Fabric, IPFS, and DTs used in the proposed solution protect against major attacks but are still prone to attacks as mentioned above. The mitigation strategies taken into account are illustrated in Table 12

In this section, we have discussed and showcased the security threats involved in IoV and how blockchain-based countermeasures can help in mitigating such risks. Security analysis of the proposed solution was discussed as well. Blockchain and DT as a technology mitigate some risks in conventional IoV systems but also introduce some more threats.

5.7 Summary

In this section of the thesis, we have answered the research question **RQ4: What are the security risks in industrial systems, and how blockchain-based DTs can be used to mitigate them?**. We have used STRIDE and ISSRM, to first present the elaborative

Table 11. Threats caused by blockchain-based DTs in V2S communication

STRIDE	***Threats introduced by Digital Twins***	***Threats introduced by Blockchain***
Spoofing	**ST5**	**ST6, ST7, ST8**
Tampering	**TT5**	**TT6, TT7**
Repudiation	**RT5**	**RT6**
Information Disclosure	**IT4**	**IT5**
Denial of Service	**DT4**	**DT5**
Elevation of Privileges	**ET4**	**ET5**

Table 12. Threats introduced by blockchain-based DTs in V2S communication

Security Risks	**Countermeasure**
ST5	Verified trusted third-party software providers mitigated the risk of outdated software and malware injection
ST6, **IT5**	SSL certificate usage. Use of long passwords and frequent change of passwords mitigates access of malicious nodes
ST7	Authentication and authorization protocols which do not share passwords through P2P network
ST8	Reduction of attack surfaces i.e. less no of APIs. Privately verifying each transaction with the system
TT5, **RT5**	Automated Packet filtering and packet inspection before vehicle digital twins are synchronized with the physical vehicle
TT6	Use of Proof-of-Stake (PoS) consensus rather than Proof-of-Work (PoW) which is more secure than the latter.
TT7	Stronger access control mechanism. Use a stronger password with a combination of Uppercase, Lowercase, letters, symbols, and so on.
RT6	Use of Proof-of-Stake (PoS) consensus rather than Proof-of-Work (PoW) which is more secure than the latter.
IT4	Removal of affected nodes as soon as possible. Stronger RBAC allowing role-specific access to information about other vehicles
DT5, **DT4**	Monitoring and analysis unit for network activity management
ET4, **ET5**	Deep packet inspection and message filtering. Trojan Horse detection software

discussion on the threat landscape of IoV in general, the risk associated with it, and mitigation strategies using blockchain-based DTs. Secondly, this section also presents the threat model for blockchain-based DTs for IoV and suggests mitigation strategies that can be taken into account for such a system. This section aids in the goal of the thesis to provide a blockchain-based DT solution for IoV following the design science research methodology guidelines **DG3** mentioned in the Section 1.4. The artifacts produced in this section have been presented to answer the research question.

6 Solution Design

This section seeks to examine the design artifacts introduced in this thesis, ensuring their alignment with the guidelines (**DG7, DG5**) outlined in Table 1 for the design science research methodology. The primary focus of this section is on various inputs that will influence the development of the blockchain-based DT software solution, addressing the research question **RQ4: How to design V2S data communication in IoV using a blockchain-based DT solution?** To address this research question effectively, we present design artifacts that contribute to the enhancement of the research process. As shown in Table 13, for this research question we present, *a) user stories*, *b) use cases* for V2S communication in IoV, *c) sequence diagrams* explaining the interactions of components, and *d) class diagram.* All these artifacts contribute to a better understanding of the proposed blockchain-based DT solution for IoV.

6.1 Design Artifacts

The design artifact is a construct, model, or method applied in the development of a software solution [38]. These artifacts provide a vocabulary or symbols that help to realize the problem and solutions. The list of artifacts we selected to produce, to identify the problems and solutions to these problems, in the V2S communication process is listed in Table 13.

6.2 User Stories

User stories are the functional requirements extracted from the case study in Section 3.3. These will help us to list out the core features and functionalities that need to be developed for the solution. The system response and the responsible user for performing those functionalities are also pointed out by these user stories. The table shows the user stories for the proposed blockchain-based DT solution for V2S communication.

However, these user stories only illustrate an overview of the core features and lack the preconditions, postconditions, and expectations. Use case diagrams will explain these details which are discussed in the Section 6.3.

Table 13. Design artifacts produced in this thesis.

Artefact	**Importance and its Purpose**
User Stories	User Stories will help to identify the core business requirements. One user story contains the responsible person who needs to do some action and specifies the reason why he needs to perform the specific action.
Use cases	It helps to define the scope of the project by creating the system boundary and actors who are involved. With the help of user stories, we define the features and functions of the system and the flow of actions performed by the actors. This gives more precise information about the behavior of the developed system
Class Diagram	The class diagram is the advanced form of ERD. It helps to identify how many classes we require to create while developing the solution. How these classes interact with each other, what parameters are required by their member functions, and what access level we have to define for the member functions and attributes of the class. The developer can start writing code with the help of this diagram.
Sequence Diagrams	A sequence diagram will help us to understand the insights of the software solution. In these diagrams, we can see the internal flow of every single function, how these functions perform tasks, and how they create or destroy the data models. It helps to see what kind of exception can occur and what alternative response should generate. This diagram is technical in terms of software development.

6.3 Use Cases

The most prominent to visualize the user stories graphically with the broader view is to represent and transform the needs mentioned in user stories into use cases and use case diagrams, as shown in Figure 20. Each oval-shaped use case has its detailed technical functionality, which has been discussed individually.

The actors in the V2S communication are: *Vehicle*, *Analyzer*, *RSU*, and *TA*. The actors are drawn on the left side of the box in the diagram. The *Vehicle* tries to connect to the region cloud. The *RSU* with the help of the *TA* verifies the vehicle identity and allows it to upload sensory data and send an analysis request. Any free *Analyzer* picks up the analysis job, analyzes the new vehicle sensory data, and notifies the *RSU* about the result of the analysis. Next, the *RSU* sends the analysis result to the *Vehicle*. In the case of successful analysis, the *Vehicle*, when notified, synchronizes its DT and sends the notification message to all stakeholders in the region cloud. On the other hand, if the analysis is not successful, the *RSU* asks for the real identity of the vehicle from the *TA*. It also flags the *vehicle* as malicious and all other stakeholders in the cloud are notified. All

Table 14. User Stories for V2S Communication

Id	User Stories
US01	As a vehicle user, I want to request vehicle information verification, so that the vehicle can be registered into the RSU region cloud.
US02	As an authorized vehicle user, I want to upload sensory data to the RSU region cloud, so that the data can be analyzed and verified.
US03	As an authorized vehicle user, I want to send a sensory data analysis request, so that the request can be broadcasted to another already verified analyzer.
US04	As an authorized analyzer, I want to analyze new vehicle sensory data, so that the data can be used for services of the system.
US04	As the RSU, I want to send successful analysis result notifications, so that a new vehicle can synchronize its digital twin in the cloud.
US05	As the RSU, I want to send an unsuccessful analysis result notification, so that a new vehicle can be flagged as malicious and further requests from it are discarded.

future requests from the vehicle are ignored.

6.3.1 Use case #1 - Send Authorization Request

As shown in Table 15, any new vehicle user will request identity verification in this use case. The RSU will verify the real identity of the vehicle with the help of the TA.

6.3.2 Use case #2 - Upload Sensory data

In this use case, as shown in Table 16 the authorized vehicle user will start uploading aggregated sensory data. RSU will allow upload and notify other already verified vehicles about the newly uploaded data.

6.3.3 Use case #3 - Send Data Analysis Request

In this use case, as shown in Table 17 the authorized vehicle user will send a data analysis request. An analyzer will pick up the analysis job.

6.3.4 Use case #4 - Analyze Sensory Data

In this use case, as shown in Table 18 the authorized analyzer will start the analysis process. Compare new vehicle data with historical data. Sends analysis result to the RSU which later notifies the new vehicle and changes its status.

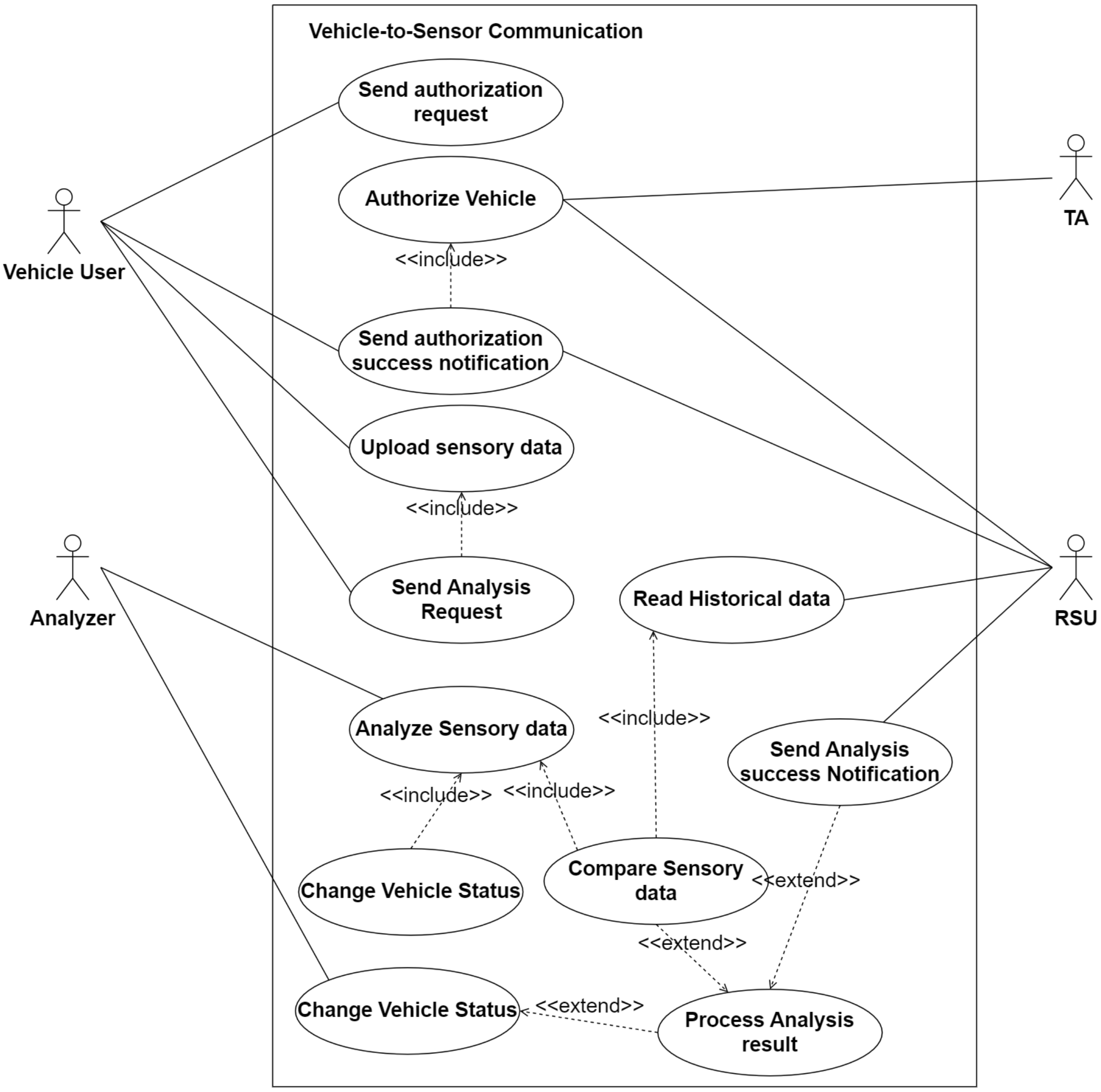


Figure 20. Vehicle-to-Sensor Communication Use Case Diagram.

6.3.5 Use case #5 - Send Unsuccessful Analysis Notification

In this use case, as shown in Table 19 the RSU sends an unsuccessful analysis result to the new vehicle and changes its status to be malicious.

6.4 Interaction - Sequence Diagrams

Specific actions are required to fulfill the functionality for each use case. Use cases lack technical details and merely provide a general explanation of the capability. Every

Table 15. Use case for the authorization request.

UC01	**Send Authorization Request**
Dependencies	User story US01
Description	The RSU should behave according to the use case when actor tries to get authorized into the RSU region cloud
Pre-condition	The new vehicle does not need to be authorized to send this request, any vehicle can send this request.
Ordinary Sequence	Step & Action 1. Actor Vehicle sends an authorization request to the region cloud 2. RSU verifies the real identity of the vehicle with the help of TA 3. RSU notifies the new vehicle about authorization result
Post-condition	The new vehicle gets authorized into the region cloud.
Exception	2. RSU fails to verify the identity of the vehicle and sends failed notification
Comments	

Table 16. Use case for uploading sensory data to region cloud.

UC02	**Upload Sensory Data**
Dependencies	User story US02
Description	The RSU should behave according to the use case when the actor try to get upload sensory data into the RSU region cloud
Pre-condition	The new vehicle needs to be authorized to send this request.
Ordinary Sequence	Step & Action 1. Actor Vehicle sends aggregated sensory data to the region cloud 2. RSU verifies the authorization of the vehicle 3. RSU notifies the already verified vehicles to take on this analysis job
Post-condition	The new vehicle successfully upload data into the region cloud.
Exception	2. RSU fails to verify the authorization of the vehicle.
Comments	

activity in the use case is a step. Classes interact with one another in a predetermined sequence as they generate, which accurately represents the action described in the use case. A sequence diagram provides more technical details regarding the behavior and

Table 17. Use case for sending sensory data analysis request.

UC03	**Send analysis request**
Dependencies	User story US03
Description	The RSU should behave according to the use case when the actor try to send a sensory data analysis request to the region cloud.
Pre-condition	The new vehicle needs to be authorized to send this request.
Ordinary Sequence	Step & Action 1. Actor Vehicle sends sensory data analysis request to the region cloud 2. RSU notifies the already verified analyzers to pick the analysis job. 3. The already verified analyzer successfully picks up the analysis job.
Post-condition	The analyzer starts the newly uploaded data analysis.
Exception	3. The analyzer fails to start the analysis.
Comments	

Table 18. Use case for sending sensory data analysis request.

UC04	**Analyze Sensory Data**
Dependencies	User story US04
Description	The analyzer should behave according to the use case when the actor picks up the analysis request.
Pre-condition	The analyzer needs to be free and have enough resources.
Ordinary Sequence	Step & Action 1. Actor analyzer picks the data analysis request. 2. The already verified analyzer analyzes the new sensory data by comparing it with its own historical data. 3. The analyzer successfully analyzes the sensory data. 4. The analyzer notifies the RSU about the analysis result.
Post-condition	The analyzer finishes the analysis and changes its status and frees up resources.
Exception	3. The analyzer fails to successfully analyze the new data.
Comments	

flow of the system by illustrating how things interact inside it. Sequence diagrams contribute to the construction of the class diagram. Once the class diagram is complete, it provides the programmer with further technical details about the system that has to be constructed. The sequence diagrams explained in this section are the sequence flow of

Table 19. Use case for sending sensory data analysis request.

UC05	**Send unsuccessful analysis notification**
Dependencies	User story US05
Description	The system acts according to the user story.
Pre-condition	The RSU needs to receive notification from the analyzer.
Ordinary Sequence	Step & Action 1. The analyzer could not successfully analyze the new sensory data. 2. The analyzer notifies the RSU about the analysis result. 3. RSU sends a fail notification to the new vehicle. 4. RSU flags the new vehicles as malicious and notifies all stakeholders.
Post-condition	The new vehicle is flagged as malicious and any further request from it is discarded.
Exception	3. RSU fails to send a fail notification to the new vehicle.
Comments	

V2S communication using the proposed blockchain-based solution.

6.4.1 Vehicle Authorization

The proposed solution for V2S communication begins with a new vehicle being authorized into the region cloud. The sequence flow of vehicle authorization starts with the vehicle user sending an authorization request to the RSU region cloud. The sequence is demonstrated using the use case UC01 described in Section 6.3.1. A new vehicle sends an authorization request upon entering the region cloud. The RSU responds by verifying the vehicle's real identity with the TA's help and notifying the vehicle. Figure 21 illustrates the sequence of vehicle authorization.

6.4.2 Vehicle Sensory Data Analysis Request

The sequence of vehicle sensory data analysis requests as shown in figure 22, is demonstrated using use case no UC02 described in Section 6.3.3. This sequence starts when an authenticated new vehicle sends a data analysis request via RSU. The request is broadcasted to the cloud by RSU. The region cloud responds with broadcast success once the analysis job is picked up by any analyzer.

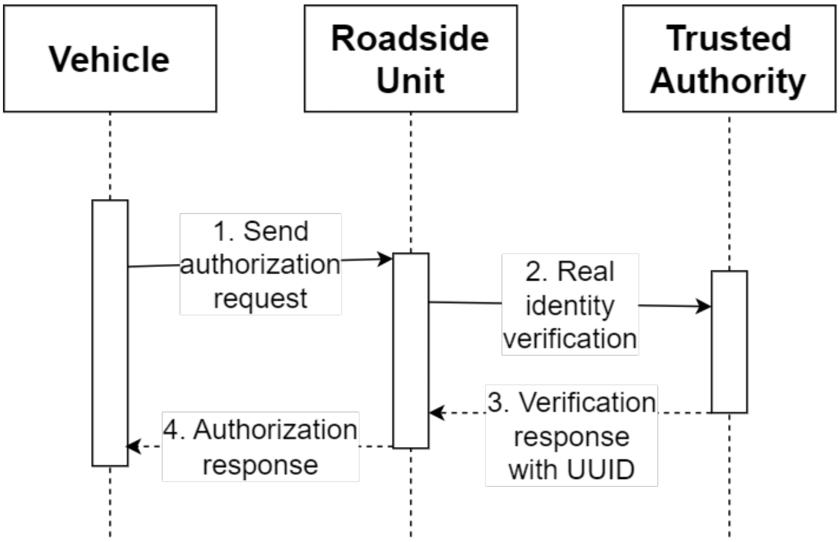

Vehicle
Roadside Unit
Trusted Authority
1. Send authorization request
2. Real identity verification
3. Verification response with UUID
4. Authorization response

Figure 21. Sequence of Successful Vehicle Authentication Request.

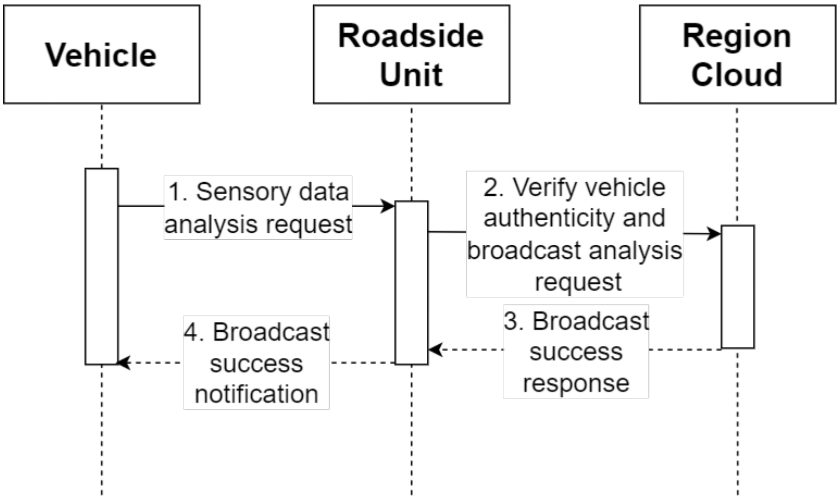
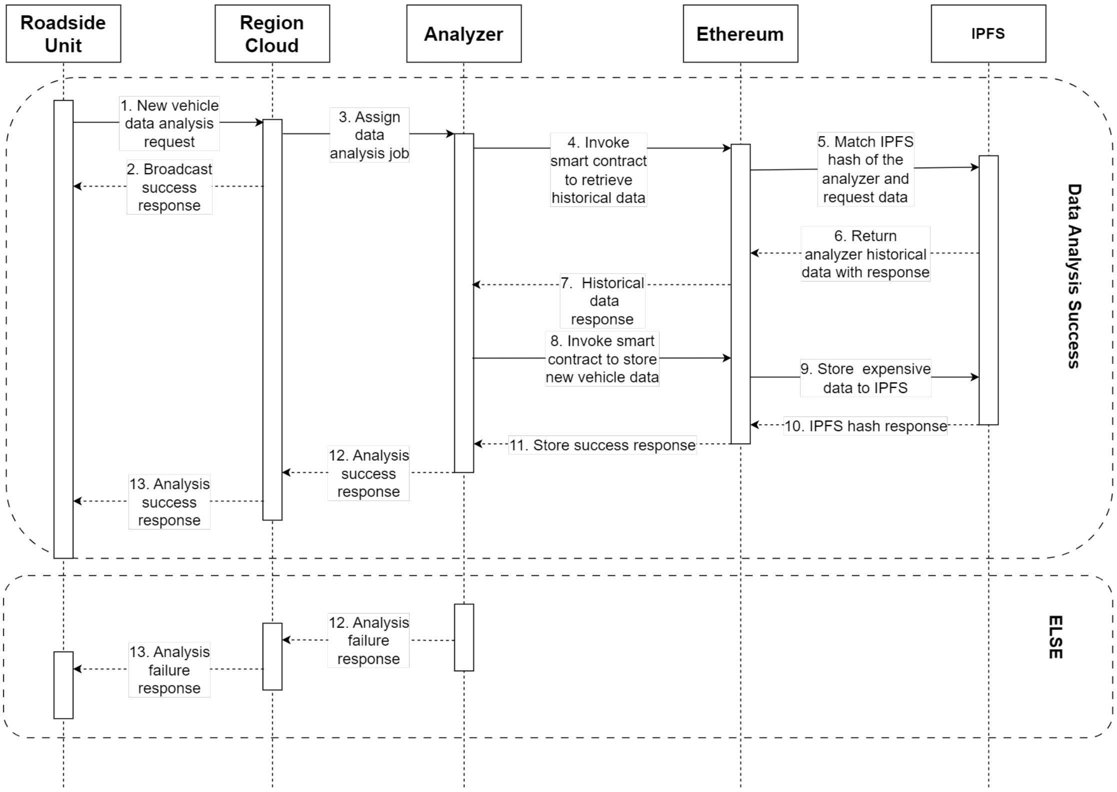

Vehicle
Roadside Unit
Region Cloud
1. Sensory data analysis request
2. Verify vehicle authenticity and broadcast analysis request
3. Broadcast success response
4. Broadcast success notification

Figure 22. Sequence of Successful Sensory Data Analysis Request.

6.4.3 Sensory Data Analysis and Storage

The main sequence in V2S communication is data analysis and storage. It is illustrated in figure 23. The sequence is demonstrated using use case no UC03 and described in Section 6.3.4. It begins when the analysis job is assigned to an already authorized analyzer in the region cloud. The analyzer then invokes the smart contract on the Ethereum blockchain to retrieve its historical data. Ethereum retrieves the historical data from IPFS storage using on-chain stored IPFS hash. Once the analyzer gets the data, it matches it with the new vehicle data stored in the request. After successful analysis, the analyzer stores the new vehicle UUID, analyzer UUID, and timestamp as on-chain storage. It also stores the newly analyzed data in the IPFS system and stores its hash on the blockchain. Thus, the analysis process is concluded.



Roadside Unit
Region Cloud
Analyzer
Ethereum
IPFS
1. New vehicle data analysis request
2. Broadcast success response
3. Assign data analysis job
4. Invoke smart contract to retrieve historical data
5. Match IPFS hash of the analyzer and request data
6. Return analyzer historical data with response
7. Historical data response
8. Invoke smart contract to store new vehicle data
9. Store expensive data to IPFS
10. IPFS hash response
11. Store success response
12. Analysis success response
13. Analysis success response
Data Analysis Success
12. Analysis failure response
13. Analysis failure response
ELSE

Figure 23. Sequence of Sensory Data Analysis & Storage.

6.4.4 Send Sensory Data Analysis Notification

The sequence of sending sensory data analysis notification begins as the RSU get a response from the analyzer about the analysis result. It notifies the vehicle about the result and changes its state to "LIVE" if the analysis is successful. When the analysis fails, the state of the new vehicle is changed to "Malicious" and the stakeholders in the region cloud are notified. Any further requests from the vehicle are discarded. The sequence is illustrated in figure 24.

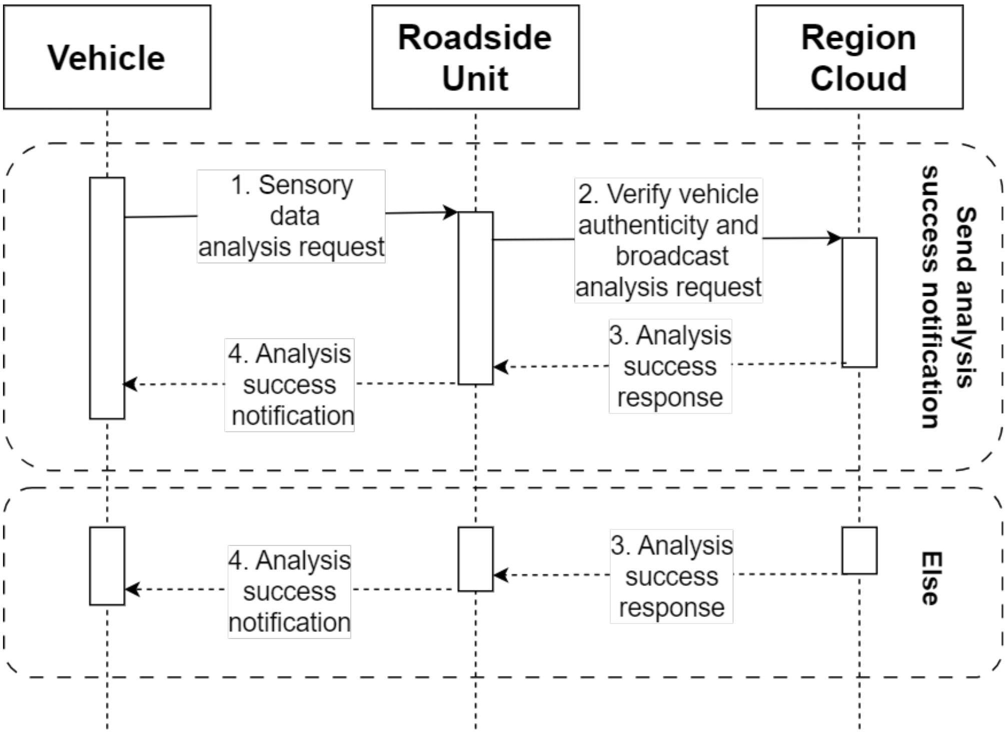

Vehicle
Roadside Unit
Region Cloud
1. Sensory data analysis request
2. Verify vehicle authenticity and broadcast analysis request
3. Analysis success response
4. Analysis success notification
Send analysis success notification
3. Analysis success response
4. Analysis success notification
Else

Figure 24. Sequence of Sending Sensory Data Analysis Notification.

6.5 Class Diagram - V2S Communication in IoV

The class diagram represents classes and attributes of the system that will interact with each other. The class diagram derives from the Sequence Diagram. Each action message in the sequence diagram converts into the class diagram methods, and the object converts into attributes. The class diagram for our proposed solution is shown in 25.

6.6 Summary

The implementation of the proposed solution in this thesis has seven classes mentioned in the class diagram i.e. *Analyzer*, *Vehicle*, *sensory Data*, *Notification*, *RSU Contract*, *Analyzer Contract*, and *Vehicle Contract*. The details about these classes are given in the Section 7.2. This section exhibits the artifacts produced in this thesis following our research method and answers **RQ4: How to design secure V2S data communication in IoV using Blockchain-based DT solution?** mentioned in Section 1.3. The artifact produced in this section aids in the Section 7 and takes into consideration the threats and mitigation strategies presented in Section 5.

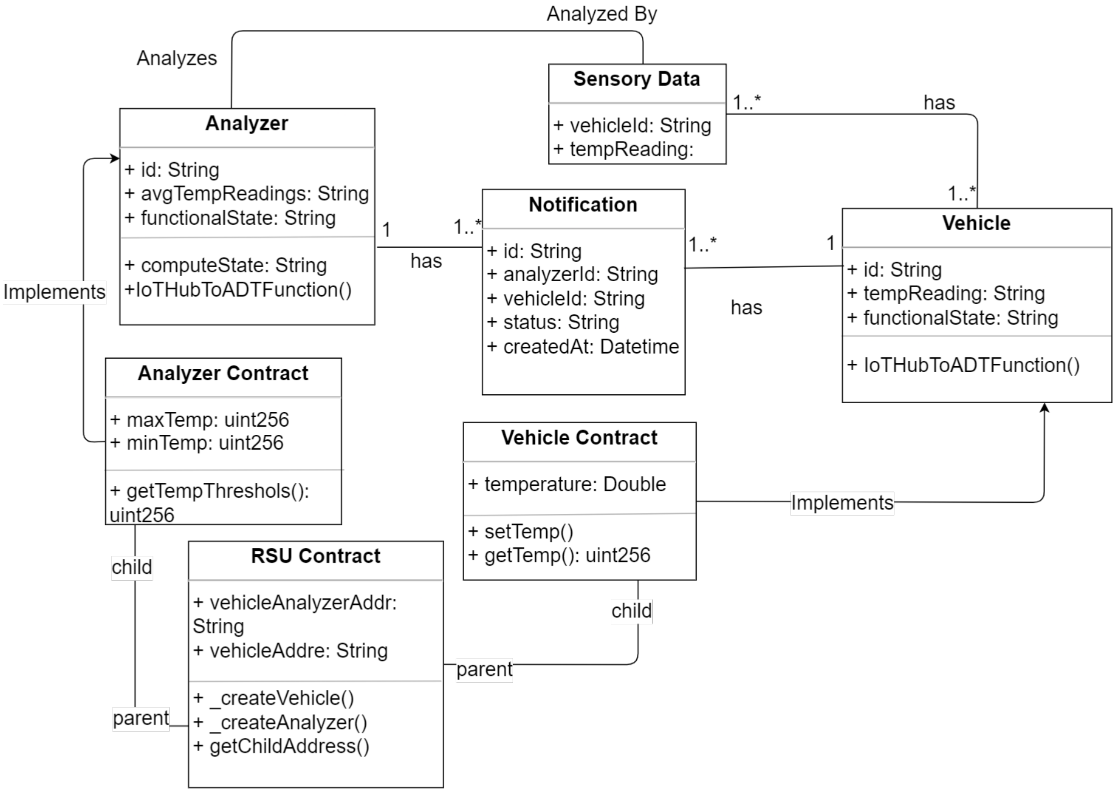


Figure 25. Class diagram illustrating a list of methods and attributes of smart contract and system for V2S communication

7 Implementation

According to the design science guidelines mentioned in Table 1, it is essential to provide design implementation of the thesis contribution for the research to be considered acceptable. Hence, this section showcases the implementation of the proposed framework in this thesis. Implementation of the proposed solution conforms to the research guideline **DG6. Design as a search process**. Implementation is also necessary to prove the claims in this thesis according to the research guideline **DG3. Design Evaluation**. This implementation section also aids in the evaluation of the proposed blockchain-based DT solution illustrated in the Section 7.8. Implementation can be used as a boilerplate for future researchers. This section answers the research question **RQ6. How to implement a blockchain-based DT solution to enhance the security of sensory data communication in V2S for IoV?**. To showcase the implementation of the proposed blockchain-based DT solution for the security of V2S sensory data communication in IoV, a simulation-based approach has been used to simulate vehicles, RSUs, and IoV region clouds and interactions between each component. This section describes the approach to deploying a DT instance and developing the components based on the theory and the research done in the previous sections.

7.1 Proposed Architecture

This section discusses a prototypical implementation of the proposed conceptual framework by generating a virtual environment to demonstrate how physical assets, i.e., vehicular sensors, and RSUs can be modeled and analyzed through the twinned environment. The Microsoft Azure Digital Twins[6] service (ADT) is used which models DTs in a cloud environment following Platform-as-a-Service (PaaS) architecture and enables physical twin monitoring in a virtual environment. Figure 26 shows the architecture used for the simulation. We decided to go with the Ethereum blockchain based on the decision flowchart shown in Figure 9.

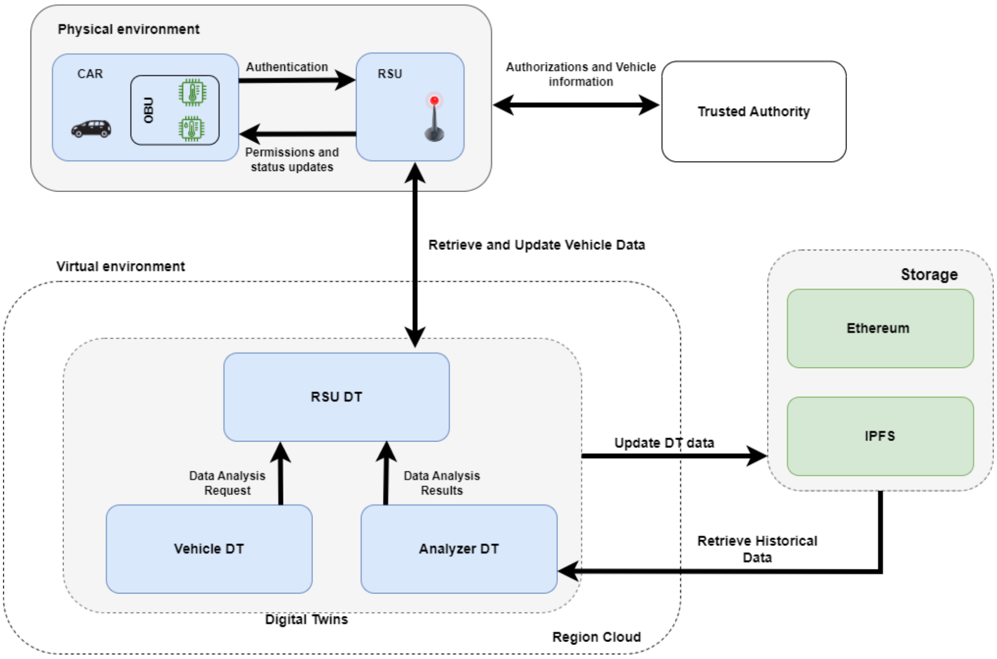


Figure 26. Implementation of the proposed architecture.

In the implementation, the architecture consists of four major parts:

- ***Physical Environment (Region Cloud)***: It consists of the actual physical components of the proposed architecture. These components like the physical RSU, vehicle entering the range of the RSU forms the region cloud.

[6] https://azure.microsoft.com/en-us/products/digital-twins

- ***Trusted Authority***: It is responsible for verifying the identity of the vehicles entering the region cloud and initializing the deployment of the contracts on the blockchain. For the simulated implementation, this was done manually.

- ***Virtual Environment***: This is the virtual counterpart of the physical entities. The digital twins of the RSU, Vehicle, and analyzer are its components. These DTs are continuously synchronized with their physical counterparts. Analysis phase operations are mostly done in this part. The components in this part are in continuous communication with the blockchain and depend on smart contracts for the analysis of vehicular data.

- ***Storage***: Blockchain and the IPFS system used by the simulated implementation are responsible for the storage of vehicular data. The ingestor and simulator applications interact with this storage part for data used in the analysis process done by the analyzer.

7.2 Digital Twin Models

In ADT, the user can define the vocabulary for building twins of the physical layer components. This is allowed by creating models for each twin representing the physical component. ADT models are represented in the JSON-LD-based Definition Language (DTDL) [7]. For the V2S scenario, four models were created to represent the involved components: **(a)** Vehicle, **(a)** RSU, **(a)** Region Cloud, and **(a)** Analyzer Vehicle,

7.2.1 Vehicle

To represent a vehicle in the twin model, as Listing 1 shows, three properties have been defined, namely *Id*, *Functional State*, and *Temperature Readings*. Here Id represents the address of the vehicle contract deployed on the blockchain. Functional State will be charged based on the analysis done in the ingestion process in an Azure function app. The Temperature Reading is simulated as an OBU temperature sensor reading.

```
{
    "@id": "dtmi:iov:digitaliovregioncloud:rsu:vehicle;1",
    "@context": "dtmi:dtdl:context;2",
    "@type": "Interface",
    "displayName": "Vehicle Interface Model",
    "contents": [
        {
            "@type": "Property",
```

[7] https://learn.microsoft.com/en-us/azure/digital-twins/concepts-models

```
            "name": "Id",
            "schema": "string",
            "description": "Vehicle Id",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "FunctionalState",
            "schema": "string",
            "description": "Functional and non-functional state of Vehicle",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "TemperatureReading",
            "schema": "double",
            "description": "Temperature Reading from OBU",
            "writable": true
        }
    ]
}
```

Listing 1. Vehicle model defined in JSON-LD using DTDL specifications.

7.2.2 RSU

In the RSU model, the address property represents its contract address. VehicleAddress and AnalyzerAddresses represent their contract addresses, respectively. The State property represents the health of the RSU. Also, two relationship properties are added for the vehicle and the analyzer, respectively, like One-to-Many relationships in conventional relational databases. That means one RSU can have multiple vehicles and analyzers. The detailed model structure is illustrated in Listing 2.

```
{
    "@id": "dtmi:iov:digitaliovregioncloud:rsu;1",
    "@context": "dtmi:dtdl:context;2",
    "@type": "Interface",
    "displayName": "Road-Side Unit Interface Model",
    "contents": [
        {
            "@type": "Property",
            "name": "Address",
            "schema": "string",
            "description": "RSU address",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "State",
            "schema": "string",
            "description": "Funcational and non-functional state of RSU",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "VehicleAddress",
            "schema": "string",
            "description": "Smart Contract Address For vehicle",
            "writable": true
        },
        {
            "@type": "Property",
            "name": "AnalyzerVehicleAddress",
            "schema": "string",
            "description": "Smart Contract Address For analyzer vehicle",
            "writable": true
        },
        {
            "@type": "Relationship",
            "@id": "dtmi:iov:digitaliovregioncloud:rsu:rel_has_vehicles;1",
            "name": "rel_has_vehicles",
            "displayName": "Has Vehicles",
            "target": "dtmi:iov:digitaliovregioncloud:rsu:vehicle;1"
        },
        {
            "@type": "Relationship",
            "@id": "dtmi:iov:digitaliovregioncloud:rsu:rel_has_analyzervehicles;1",
            "name": "rel_has_analyzervehicles",
            "displayName": "Has Incentivized Analyzer Vehicles",
            "target": "dtmi:iov:digitaliovregioncloud:rsu:analyzer_vehicle;1"
        }
    ]
}
```

Listing 2. RSU model defined in JSON-LD using DTDL specifications.

7.2.3 Region Cloud

The region cloud is the entity encapsulating an RSU, vehicles, and analyzers under this region. The property Id is the unique identifier for the region cloud in the IoV network. Location is an object-type property representing its longitude and latitude values. It has a One-to-One relationship property with its RSU (as Listing 3 shows).

```
{
  "@id": "dtmi:iov:digitaliovregioncloud;1",
  "@type": "Interface",
  "displayName": "IoV Region Interface Model",
  "@context": "dtmi:dtdl:context;2",
  "contents": [
      {
          "@type": "Property",
          "name": "Id",
          "schema": "string",
          "description": "IoV Region ID",
          "writable": true
      },
      {
          "@type": "Property",
          "name": "IoVRegionLocation",
          "schema": {
              "@id": "dtmi:iov:digitaliovregioncloud:custom_schema:iovregionlocation;1",
              "@type": "Object",
              "fields": [
                  {
                      "@id": "dtmi:iov:digitaliovregioncloud:custom_schema:iovregionlocation:lat;1",
                      "name": "Latitude",
                      "schema": "double"
                  },
                  {
                      "@id": "dtmi:iov:digitaliovregioncloud:custom_schema:iovregionlocation:lon;1",
                      "name": "Longitude",
                      "schema": "double"
                  }
              ]
          }
      },
      {
          "@type": "Relationship",
          "@id": "dtmi:iov:digitaliovregioncloud:rel_has_road_side_units;1",
          "name": "rel_has_road_side_units",
          "displayName": "Has Road-Side Units",
          "target": "dtmi:iov:digitaliovregioncloud:rsu;1"
      }
  ]
}
```

Listing 3. Region Cloud Model defined in JSON-LD using DTDL specifications.

7.2.4 Analyzer

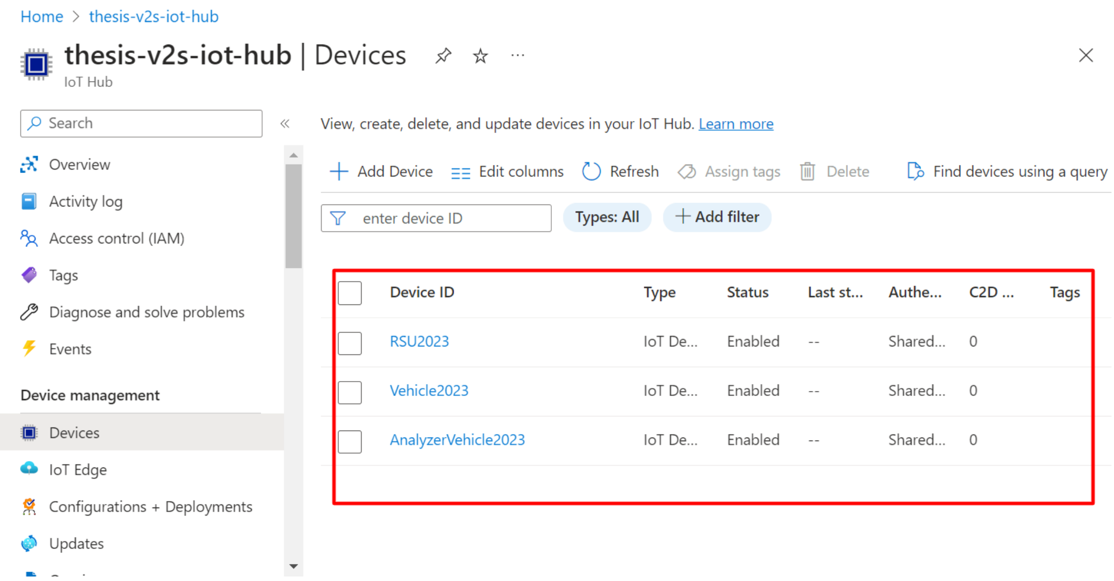
The analyzer mimics any non-malicious vehicle already present in the cloud and validates the sensory data from a newly joined vehicle in the network. Id property represents the unique identifier, FunctionalState is the health of the analyzer, and ComputeState is the computational state of the analyzer ("BUSY", "FREE" etc.). Lastly, AverrageOfLastThreeReadings is the property for the average temperature reading. The JSON structure of the model can be seen in Listing 4.

```
{
  "@id": "dtmi:iov:digitaliovregioncloud:rsu:analyzer_vehicle;1",
  "@context": "dtmi:dtdl:context;2",
  "@type": "Interface",
  "displayName": "Analyzer Vehicle",
  "contents": [
      {
          "@type": "Property",
          "name": "Id",
          "schema": "string",
          "description": "Analyzer Vehicle Id",
          "writable": true
      },
      {
          "@type": "Property",
          "name": "FunctionalState",
          "schema": "string",
          "description": "Funcational and non-functional state of Vehicle",
          "writable": true
      },
      {
          "@type": "Property",
          "name": "ComputeState",
          "schema": "string",
          "description": "Computational or Non-computational state of Vehicle",
          "writable": true
      },
      {
          "@type": "Property",
          "name": "AverageOfLastThreeReadings",
          "schema": "double",
          "description": "Average of the last three temperature readings",
          "writable": true
      }]}
```

Listing 4. Analyzer model defined in JSON-LD using DTDL specifications.

7.3 Simulators

To simulate the real entities (cars, analyzers, RSUs, etc.), *C#* console applications have been used. The applications interact with the resources created in **Azure IoT Hub** [8]. Three devices have been created under an instance to represent the vehicle, analyzer, and RSU. The devices (RSU2023, Vehicle2023, and AnalyzerVehicle2023) as seen in Figure 27 are subscribers to the data events pushed by the simulators to the Azure IoT Hub instance. They simulate real devices and interact with the device instances in the Azure portal cloud (as shown in Figure 27). The vehicle's simulator application is described below. Codebase for all other Data simulators is publicly available on GitHub [9].



View, create, delete, and update devices in your IoT Hub. Learn more

□	Device ID	Type	Status	Last st...	Authe...	C2D ...	Tags
□	RSU2023	IoT De...	Enabled	--	Shared...	0	
□	Vehicle2023	IoT De...	Enabled	--	Shared...	0	
□	AnalyzerVehicle2023	IoT De...	Enabled	--	Shared...	0	

Figure 27. Instances of components in Azure IoT Hub.

The simulator application mimics real-life sensory data in JSON format and pushes it into respective Azure IoT hub devices which are then used to update the digital twins. For the vehicle simulation in the main function, we call the *SimulateDeviceToSendD2cAndReceiveD2c* function. The function (as shown in Listing 5) takes no parameters but calls two asynchronous functions from the class named *AzureIoTHub*. The first function *SendDeviceToCloudMessageAsync* as shown in Listing 7 sends telemetry to IoT Hub and the latter *ReceiveMessagesFromDeviceAsync* as shown in Listing 8 receives any reply from it.

```
private static async Task SimulateDeviceToSendD2cAndReceiveD2c()
```

[8] https://azure.microsoft.com/en-us/products/iot-hub
[9] https://github.com/ashfaqshuvo007/thesis-adt-data-simulators

```
        {
            var tokenSource = new CancellationTokenSource();

            Console.CancelKeyPress += (s, e) =>
            {
                e.Cancel = true;
                tokenSource.Cancel();
                Console.WriteLine("Exiting...");
            };
            Console.WriteLine("Press CTRL+C to exit");

            await Task.WhenAll(
                AzureIoTHub.SendDeviceToCloudMessageAsync(tokenSource
    .Token),
                AzureIoTHub.ReceiveMessagesFromDeviceAsync(
    tokenSource.Token));

            tokenSource.Dispose();
        }
```

Listing 5. Main function to communicate with the Vehicle created in IoT Hub.

The function *SendDeviceToCloudMessageAsync* performs the task in three steps:

```

private static string iotHubConnectionString = @$"HostName={hubName}.
    azure devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey
    ={hubSharedAccessKey}";

private static string deviceConnectionString = $"HostName={hubName}.
    azure-devices.net;DeviceId={deviceName};SharedAccessKey={
    deviceSharedAccessKey}";
```

Listing 6. Connection strings to connect to Azure cloud.

- ***Instantiate IoT Hub*:** First, it creates an instance of the virtual device in IoT Hub using a connection string *deviceConnectionString* as shown in Listing 6. In this string, *hubName* refers to the name of the IoT Hub instance created in Azure Portal. *deviceName* is the unique name set while creating the device in the IoT hub (e.g. Vehicle2023) and **sharedDeviceAcessKey** is the private key assigned automatically to the device to make it accessible by external systems.

- ***Create fake telemetry data*:** Now, for this simulation the fake telemetry data object is created. In this case, a temperature reading (of double data type). The message data is converted to a JSON object and sent to the device on IoT Hub.

- ***Wait for completion and repeat*:** Finally, it waits for the asynchronous request to complete and repeats every minute. The delay time can be manipulated using *Task.Delay()*.

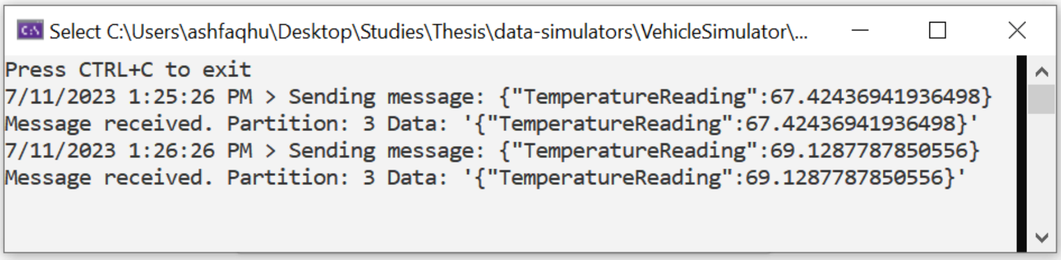
```
public static async Task SendDeviceToCloudMessageAsync(CancellationToken cancelToken)
  {
      var deviceClient = DeviceClient.CreateFromConnectionString(deviceConnectionString);
      string id = deviceId;

      double temperature = 70.0D;
      var rand = new Random();

      while (!cancelToken.IsCancellationRequested)
      {
          double currentTemperature = temperature + rand.NextDouble() * 4 - 3;

          var telemetryDataPoint = new
          {
              TemperatureReading = currentTemperature,
          };
          var messageString = JsonSerializer.Serialize(telemetryDataPoint);
          var message = new Microsoft.Azure.Devices.Client.Message(Encoding.UTF8.GetBytes(messageString))
          {
              ContentType = "application/json",
              ContentEncoding = "utf-8"
          };
          await deviceClient.SendEventAsync(message);
          Console.WriteLine($"{DateTime.Now} > Sending message: {messageString}");

          await Task.Delay(60000);
      }
  }
```

Listing 7. Function Sending telemetry to virtual Vehicle in IoT Hub.

However, the second function *ReceiveMessagesFromDeviceAsync* creates a consumer client to consume any message (JSON Object) coming from the virtual Vehicle2023 device in IoT Hub. It waits for any incoming messages from IoT Hub, converts them to string, and logs it to the console window. This goes on as long as the process is not exited. Figure 28 shows typical message logs for the application.

```
  public static async Task ReceiveMessagesFromDeviceAsync(CancellationToken cancelToken)
      {
          try
          {
              string eventHubConnectionString = await IotHubConnection.GetEventHubsConnectionStringAsync(iotHubConnectionString);
              await using var consumerClient = new EventHubConsumerClient(
                  EventHubConsumerClient.DefaultConsumerGroupName,
                  eventHubConnectionString);

              await foreach (PartitionEvent partitionEvent in consumerClient.ReadEventsAsync(cancelToken))
              {
                  if (partitionEvent.Data == null) continue;

                  string data = Encoding.UTF8.GetString(partitionEvent.Data.Body.ToArray());
                  Console.WriteLine($"Message received. Partition: {partitionEvent.Partition.PartitionId} Data: '{data}'");
              }
          }
          catch (TaskCanceledException) { } // do nothing
          catch (Exception ex)
          {
              Console.WriteLine($"Error reading event: {ex}");
          }
      }
```

Listing 8. Receiving response from IoT Hub.


Select C:\Users\ashfaqhu\Desktop\Studies\Thesis\data-simulators\VehicleSimulator\...

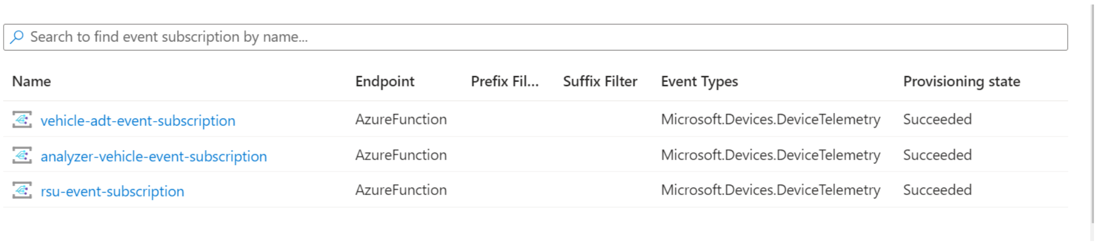
```
Press CTRL+C to exit
7/11/2023 1:25:26 PM > Sending message: {"TemperatureReading":67.42436941936498}
Message received. Partition: 3 Data: '{"TemperatureReading":67.42436941936498}'
7/11/2023 1:26:26 PM > Sending message: {"TemperatureReading":69.1287787850556}
Message received. Partition: 3 Data: '{"TemperatureReading":69.1287787850556}'
```

Figure 28. Vehicle simulator sending telemetry data.

7.4 Data Ingestors

Data ingestors for the proposed solution are Azure function applications that retrieve data sent to IoT Hub as events from Data simulator applications. They analyze the data and update the digital twins for components in the V2S sensory data communication in the proposed architecture. These function applications act as event subscribes with *Azure Data Owner* roles on the IoT hub for the resource group created as the simulation layer on Azure cloud. Azure function applications for the data ingestors in the proposed architecture are available on GitHub [10]. Implementation for vehicle ingestor application is illustrated here.

All the events coming to the IoT Hub for our proposed solution have respective subscriptions (as Figure 29 shows) attached to them. Each subscription has an Azure function to process data coming to the devices and update their digital twin counterpart. For example, the (vehicle-adt-event-subscription) is for our **Vehicle2023** device on IoT Hub. The highlighted Azure function *IoTHubToADTFunction* in Figure 30 is the ingestor function updating the vehicle digital twin based on the telemetry data update events from the vehicle simulator. For developing the ingestor Azure functions for the proposed architecture references have been taken from Azure's sample Github [11] repository.



Search to find event subscription by name...

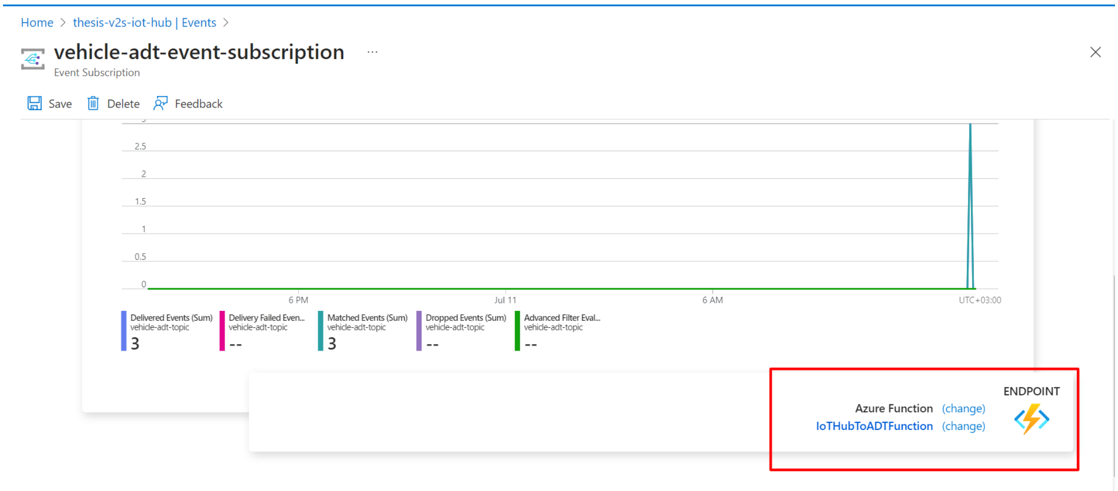
Name	Endpoint	Prefix Fil...	Suffix Filter	Event Types	Provisioning state
vehicle-adt-event-subscription	AzureFunction			Microsoft.Devices.DeviceTelemetry	Succeeded
analyzer-vehicle-event-subscription	AzureFunction			Microsoft.Devices.DeviceTelemetry	Succeeded
rsu-event-subscription	AzureFunction			Microsoft.Devices.DeviceTelemetry	Succeeded

Figure 29. Event subscriptions in Azure IoT Hub for the devices.

To demonstrate the workflow of the Vehicle Ingestor Azure function application, we first run the vehicle simulator to create fake telemetry data and send it to Azure IoT Hub as events. The ingestor application is directly deployed in the Azure cloud under our resource group and a subscription to an analyzer device **(vehicle-event-subscription)** is assigned to it. The Azure function *IoTHubToADTFunction* catches these event invocations and processes the data in them. And finally, update the DT on ADT Explorer. Successful DT update logs can be seen in the metrics for the ingestor Azure function *IoTHubToADTFunction*. Figure 31 illustrates the data ingestion flow diagram.

Moreover, the vehicle ingestor azure function application is a C# class named *IoTHubToADTFunction* with a single asynchronous task function that handles the logic

[10] https://github.com/ashfaqshuvo007/thesis-adt-data-ingestors
[11] https://github.com/Azure-Samples/digital-twins-samples/



Home > thesis-v2s-iot-hub | Events >

vehicle-adt-event-subscription

Event Subscription

Delivered Events (Sum) vehicle-adt-topic 3

Delivery Failed Even... vehicle-adt-topic --

Matched Events (Sum) vehicle-adt-topic 3

Dropped Events (Sum) vehicle-adt-topic --

Advanced Filter Eval... vehicle-adt-topic --

ENDPOINT

Azure Function (change)

IoTHubToADTFunction (change)

Figure 30. Azure function for the vehicle in Azure IoT Hub.

of catching event telemetry data from the vehicle simulator, interacts with the analyzer smart contract data and updates its digital twin counterpart in ADT explorer. The function has three tasks (as Listing 9 shows):

- ***Instantiate ADT client***: First, it creates an instance of the virtual device event subscription already created in Azure. It uses a configuration variable "*ADT_SERVICE_URL*". This configuration variable is set in the Azure cloud after function application deployment and is basically the Azure instance URL for our resource group. It showcases the use case **UC01**.
- ***Retrieve telemetry data***: Next, it checks for EventGrid Data, if present, catches telemetry data by deserializing it into JSON object. For the vehicle ingestor, we get the temperature from the simulator as telemetry. It showcases the use case **UC02**.
- ***Analyzer data from Smart contract***: Then it communicates with the analyzer smart contract to get the temperature readings and compares them with data coming from the vehicle simulator. In the case of vehicle data falling inside the range, it proceeds to the next step by updating the vehicle state as "functional". In the other case, it marks the vehicle as "non-functional". It showcases the use case **UC03** and **UC04**.
- ***Update ADT explorer twin***: Finally, with the new values for the vehicle properties, it updates the vehicle ADT data. We can verify the update in the ADT explorer (as Listing 32 shows). It showcases the use case **UC05**.

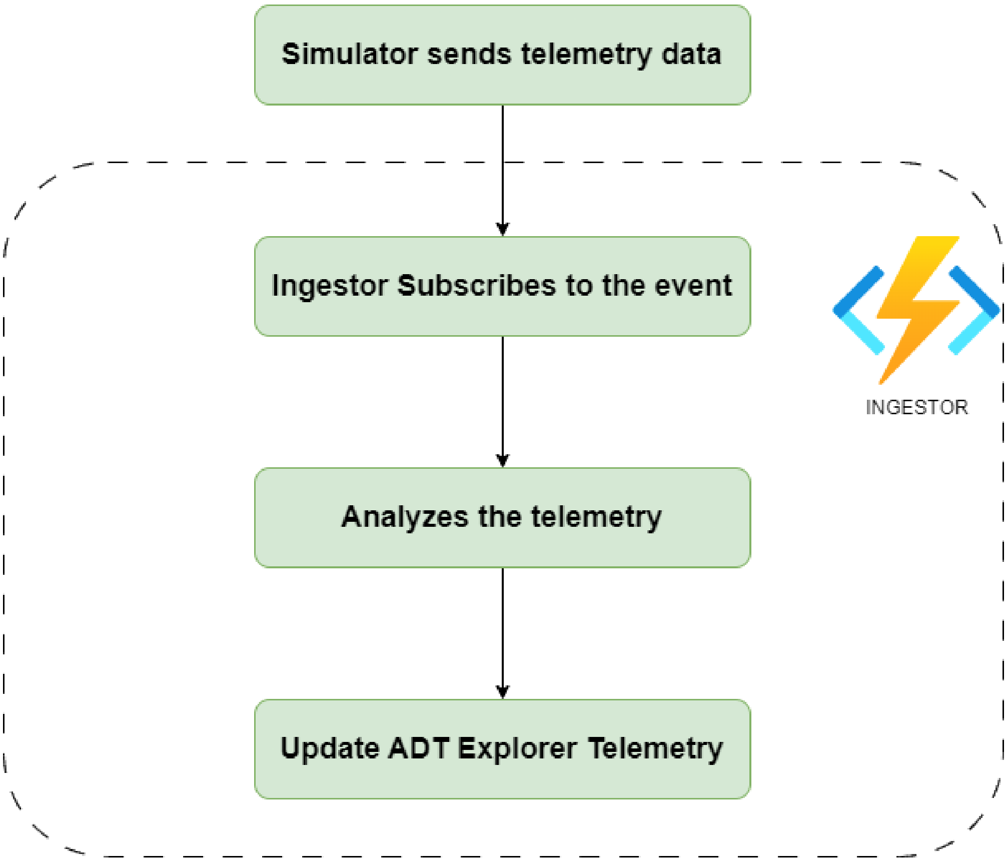
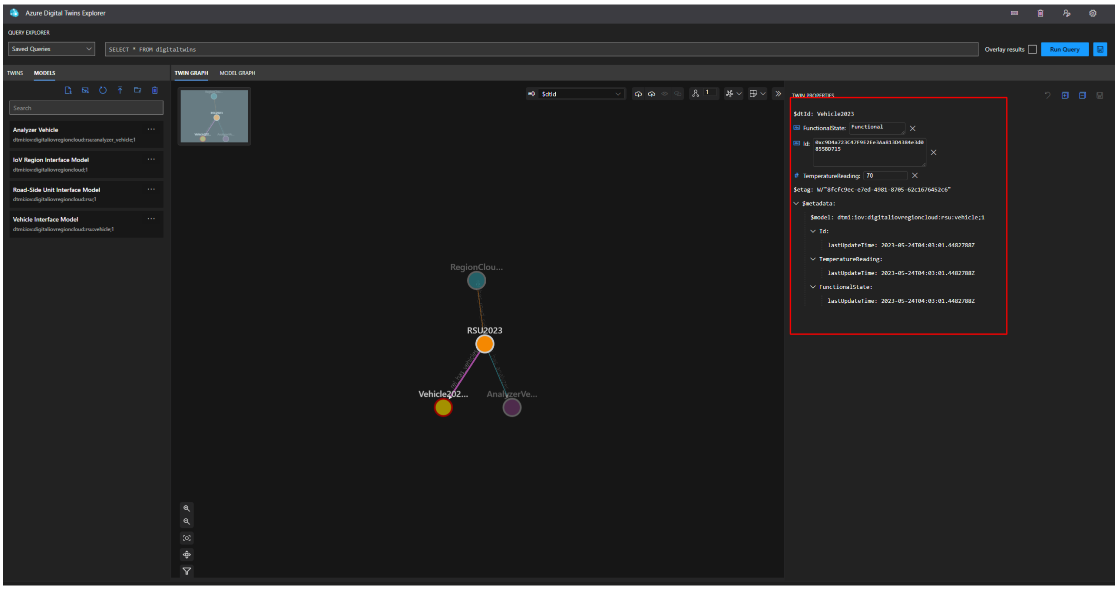

Simulator sends telemetry data
Ingestor Subscribes to the event
INGESTOR
Analyzes the telemetry
Update ADT Explorer Telemetry

Figure 31. Data Ingestion by Azure Function Flow Diagram.

```

 public static class IoTHubToAzureDigitalTwinsFunction
 {
     // ADT Instance
     private static readonly string adtInstanceUrl = Environment.GetEnvironmentVariable("ADT_SERVICE_URL");
     private static readonly HttpClient httpClient = new HttpClient();

     // Contract addresses
     private static string selfAddress = "0xc9D4a723C47F9E2Ee3Aa813D4384e3d0855BD715";
     private static string avAddress = "0x8f4410d9cD82b5D4108d2b6ADc95e1375e4c75eF";
     private static string sepoliaApiKey = "87deefc774f04134bb6afb66d6bf3cb5";

```

```
        [FunctionName("IoTHubToADTFunction")]
        public static async Task Run([EventGridTrigger]EventGridEvent eventGridEvent, ILogger log)
        {
            log.LogInformation(eventGridEvent.Data.ToString());

            if (adtInstanceUrl == null) log.LogError("Application setting \"ADT_SERVICE_URL\" not set");

            try
            {
                //Managed Identity Credentials
                var cred = new DefaultAzureCredential();

                //Instantiate ADT Client
                var adtClient = new DigitalTwinsClient(new Uri(adtInstanceUrl), cred);

                // Log successful connection creation
                log.LogInformation($"Vechile ADT client connection created!");

                //if we receive data
                if (eventGridEvent != null && eventGridEvent.Data != null)
                {
                    //Log the data
                    log.LogInformation(eventGridEvent.Data.ToString());

                    //Covert to json
                    JObject vehicleMessage = (JObject)JsonConvert.DeserializeObject(eventGridEvent.Data.ToString());

                    //Get device data from object
                    string vehicleId = (String)vehicleMessage["systemProperties"]["iothub-connection-device-id"];
                    int deviceCurrentTemperature = (int) vehicleMessage["body"]["TemperatureReading"];
                    string functionalState = "Functional";

                    //Log the telemetry
                    log.LogInformation($"Device: {vehicleId} Temperature is: {deviceCurrentTemperature}, Functional state is: {functionalState}");

                    // Smart Contract or Blockchain code goes here

                    // Get Temperature Range from Analyzer Vehicle
                var web3 = new Web3($"https://sepolia.infura.io/
v3/{sepoliaApiKey}");

                var avABI = @"[{""inputs"": [],""stateMutability""
": ""nonpayable"",""type"": ""constructor""},
                {""inputs"": [],""name"": ""getTempThresholds"","
"outputs"": [{""internalType"": ""uint256"",""name"": """",""type"
": ""uint256""},
                {""internalType"": ""uint256"",""name"": """",""
type"": ""uint256""}],""stateMutability"": ""view"",""type"": ""
function""}]";

                // Initialize AV Contract
                var contract = web3.Eth.GetContract(avABI,
avAddress);
                var tempRange = contract.GetFunction("
getTempThresholds");

                //Deserialize
                var tempValues = await tempRange.
CallDeserializingToObjectAsync<GetTemperatureOutputDTO>();

                //Log the values
                log.LogInformation($"Min temp: {tempValues.
MinTemp}");
                log.LogInformation($"Max temp: {tempValues.
MaxTemp}");

                int maxAvgTemp = (int)tempValues.MaxTemp;
                int minAvgTemp = (int)tempValues.MinTemp;

                if (deviceCurrentTemperature < minAvgTemp)
                {
                    functionalState = "Non-functional";
                }

                if (deviceCurrentTemperature > maxAvgTemp)
                {
                    functionalState = "Non-functional";
                }

                log.LogInformation($"Vehicle: {vehicleId}
Temperature is: {deviceCurrentTemperature}, Vehicle state is: {
functionalState}");

                //Update Digital Twin
                var updateTwinData = new JsonPatchDocument();
                updateTwinData.AppendAdd("/Id", selfAddress);
                updateTwinData.AppendAdd("/TemperatureReading",
deviceCurrentTemperature);
                    updateTwinData.AppendAdd("/FunctionalState",
functionalState);
                    await adtClient.UpdateDigitalTwinAsync(vehicleId,
 updateTwinData).ConfigureAwait(false);
                }
            }
            catch (Exception e)
            {

                log.LogError($"Error in Vehicle Ingest Function: {e.
Message}");
            }}}

```

Listing 9. Azure function class for vehicle ingestor.



SELECT * FROM digitaltwins

Analyzer Vehicle
dtmi:iov:digitaliovregioncloud:rsu:analyzer_vehicle;1

IoV Region Interface Model
dtmi:iov:digitaliovregioncloud;1

Road-Side Unit Interface Model
dtmi:iov:digitaliovregioncloud:rsu;1

Vehicle Interface Model
dtmi:iov:digitaliovregioncloud:rsu:vehicle;1

```
$dtId: Vehicle2023
FunctionalState: Functional
Id: 0xc9D4a723C47F9E2Ee3Aa813D4384e3d0
    855BD715
TemperatureReading: 70
$etag: W/"8fcfc9ec-e7ed-4981-8705-62c1676452c6"
$metadata:
    $model: dtmi:iov:digitaliovregioncloud:rsu:vehicle;1
    Id:
        lastUpdateTime: 2023-05-24T04:03:01.4482788Z
    TemperatureReading:
        lastUpdateTime: 2023-05-24T04:03:01.4482788Z
    FunctionalState:
        lastUpdateTime: 2023-05-24T04:03:01.4482788Z
```

Figure 32. Verifying vehicle DT update on ADT explorer.

7.5 Solidity Smart Contracts

For the analysis and governing the V2S communication, Ethereum smart contracts have been developed using solidity and deployed in the Ethereum Sepolia test network. [12]. The contracts are developed, compiled, and deployed using Hardhat[13].

[12] https://sepolia.dev/

[13] https://hardhat.org/docs

7.5.1 Vehicle Contract

As Listing 10 shows, there are two functions: **(a)** ***setTemp*** and **(a)** ***getTemp*** to interact with in the contract for a vehicle. The system might trigger the *TemeratureSet* event by calling the *setTemp* function which takes an integer parameter (temperature) to be set after successful analysis of the temperature from the analyzer in the cloud. To retrieve the current temperature of the vehicle.

```
    //SPDX-License-Identifier: UNLICENSED
    pragma solidity ^0.8.9;

    contract Vehicle {

        int private temperature;

        event TemperatureSet(address setter, int
temperature);

        function setTemp(int _temp) public {
            temperature = _temp;
            emit TemperatureSet(msg.sender,
temperature);
        }

        function getTemp() public view returns (int) {
            return temperature;
        }

    }

```

Listing 10. Smart Contract used for Vehicle Digital Twin.

7.5.2 RSU Contract

The RSU contract is responsible for deploying the other contracts as well. Since the vehicle and the analyzer are child components of the RSU digital twin, RSU controls the deployment of them on process initialization explained in Section 3.5. *getChildContractAddresses* returns the vehicle and analyzer contract addresses, which can be used by the client application to interact with them. RSU contract is illustrated in Listing 11.

```
    //SPDX-License-Identifier: UNLICENSED
    pragma solidity ^0.8.9;


    import ”./AnalyzerVehicle.sol”;
    import ”./Vehicle.sol”;


    contract RoadSideUnit {

       address public vehicleAnalyzerAddr;
       address public vehicleAddr;


       constructor() {

          Vehicle vehicle = new Vehicle();
          vehicleAddr = address(vehicle);

          VehicleAnalyzer vehicleAnalyzer = new
VehicleAnalyzer();
          vehicleAnalyzerAddr = address(
vehicleAnalyzer);
       }

       function getChildContractAddresses() public
view returns (address, address) {

          return (vehicleAddr, vehicleAnalyzerAddr);
       }

    }

```

Listing 11. Smart Contract used for Road Side Unit Digital Twin.

7.5.3 Vehicle Analyzer Contract

The analyzer returns the maximum and minimum temperature thresholds in the current scenario in the region cloud. The vehicle data ingestor explained in Section 7.4 used these threshold values to validate new vehicle sensor readings and flag the state of the vehicle as malicious or not. Listing 12 illustrates the analyzer contract.

```
        //SPDX-License-Identifier: UNLICENSED
        pragma solidity ^0.8.9;

        contract VehicleAnalyzer{
            uint256 minTemp;
            uint256 maxTemp;

            constructor () {
                minTemp = 65;
                maxTemp = 75;
            }

            function getTempThresholds() public view returns (uint256, uint256) {

                return (minTemp, maxTemp);
            }

        }

```

Listing 12. Smart Contract used for Vehicle Analyzer Digital Twin.

7.5.4 Deployment Script

To deploy the smart contracts, the script shown in Listing 13 is used. It contains an asynchronous function that collects the contract, i.e., the RSU contract for this case, and deploys the contract and the child contracts in it. After deployment, the contract address can be searched in **Etherscan Sepolia**[14] and verify that it has been deployed with two other internal contracts by going into transaction details (as shown in Figure 33).

```

    async function main() {

      // get the road side unit factory
```

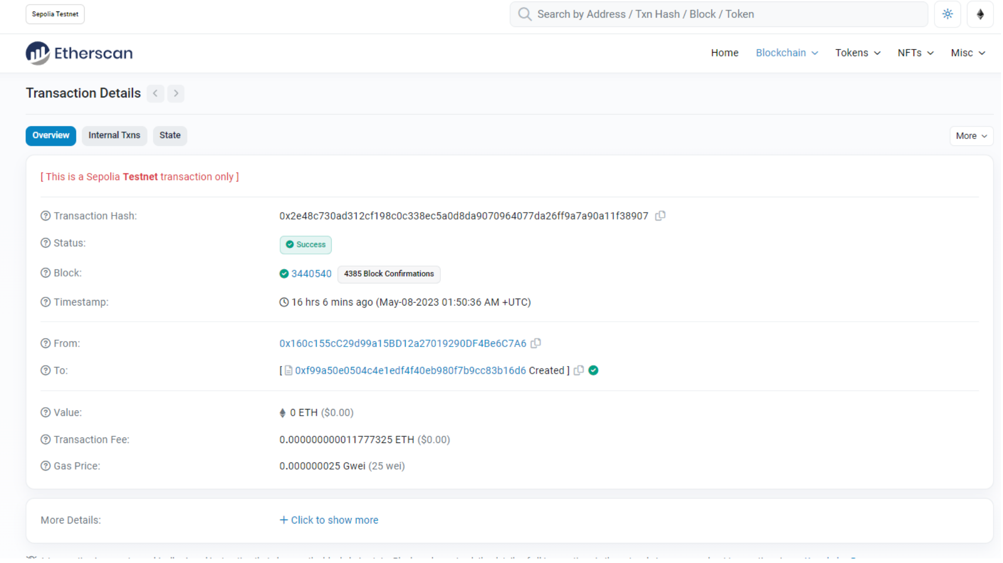
[14] https://sepolia.etherscan.io/

```
    const RoadSideUnit = await hre.ethers.getContractFactory("RoadSideUnit");

    //deploy the rsu contract which also deploys vehicle and analyzer vehicle contracts
    const rsu = await RoadSideUnit.deploy();

    // wait for the deployment completion
    await rsu.deployed();

    //Get all the addresses

    const [vehcileAddr, vehicleAnalyzerAddr] = await rsu.getChildContractAddresses();


    console.log(`RSU deployed to:  ${rsu.address}`);
    console.log(`Vehcle Contract deployed to:  ${vehcileAddr}`);
    console.log(`Analyzer Contract deployed to:  ${vehicleAnalyzerAddr}`);
  }

  // We recommend this pattern to be able to use async/await everywhere
  // and properly handle errors.
  main().catch((error) => {
    console.error(error);
    process.exitCode = 1;
  });


```

Listing 13. Script used for deploying the contracts to Ethereum.

To get more details about the proposed solution's smart contract component, please refer to this public GitHub repository[15].

[15] https://github.com/ashfaqshuvo007/thesis-adt-smart-contracts/



Transaction Details

[This is a Sepolia **Testnet** transaction only]

Transaction Hash: 0x2e48c730ad312cf198c0c338ec5a0d8da9070964077da26ff9a7a90a11f38907

Status: Success

Block: 3440540 4385 Block Confirmations

Timestamp: 16 hrs 6 mins ago (May-08-2023 01:50:36 AM +UTC)

From: 0x160c155cC29d99a15BD12a27019290DF4Be6C7A6

To: [0xf99a50e0504c4e1edf4f40eb980f7b9cc83b16d6 Created]

Value: 0 ETH ($0.00)

Transaction Fee: 0.000000000011777325 ETH ($0.00)

Gas Price: 0.000000025 Gwei (25 wei)

More Details: + Click to show more

Figure 33. Etherscan showing successful deployment of contracts.

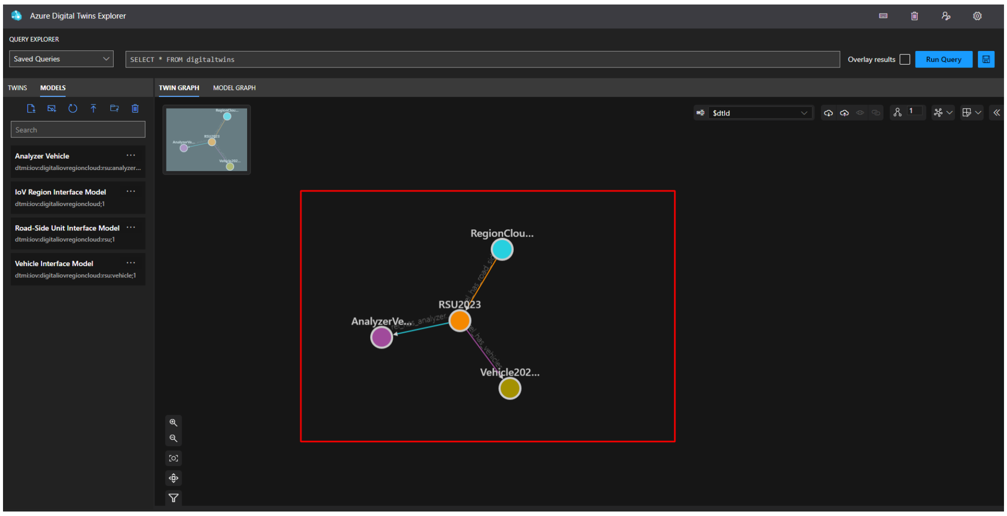
7.6 Azure Digital Twins Explorer

A locally run ADT explorer has been used for the proposed architecture simulation to create digital twins from models described in Section 7.2. All the JSON-LD-based DTDL files were uploaded to the ADT Explorer[16] application running on localhost. DTs were created using the models in the explorer window. The relationships between the twins have to be assigned manually in the explorer explicitly. From the uploaded models, multiple twins for the devices can be created. Since a very small IoV network is used for the simulation, the created DT had one Region Cloud twin, one RSU twin, and a single vehicle and analyzer. The structure of the DT used for this simulation can be seen in Figure 34.

7.7 Practical Significance

The IoV has garnered significant attention due to its promising benefits in enhancing safety, efficiency, and convenience. However, the widespread adoption of IoV has raised concerns about the various security vulnerabilities associated with this technology. While a basic level of security can be achieved through authorization and access control, IoV remains susceptible to major security risks, including Privacy Concerns, Data Integrity, V2X communication channels security, and DDoS attacks. Addressing these complex

[16] `https://learn.microsoft.com/en-us/samples/azure-samples/digital-twins-explorer/digital-twins-explorer/`



```
SELECT * FROM digitaltwins
```

Azure Digital Twins Explorer
QUERY EXPLORER
Saved Queries
Overlay results
Run Query
TWINS
MODELS
TWIN GRAPH
MODEL GRAPH
Search
Analyzer Vehicle
dtmi:iov:digitaliovregioncloud:rsu:analyzer...
IoV Region Interface Model
dtmi:iov:digitaliovregioncloud;1
Road-Side Unit Interface Model
dtmi:iov:digitaliovregioncloud:rsu;1
Vehicle Interface Model
dtmi:iov:digitaliovregioncloud:rsu:vehicle;1
$dtId
RegionClou...
RSU2023
AnalyzerVe...
Vehicle202...
rel_has_road_sid...
rel_has_analyzer...
rel_has_vehicles

Figure 34. Digital twins with relationships in the ADT explorer.

security challenges necessitates exploring more robust and sophisticated solutions. This thesis investigates the application of Blockchain-based DTs as a potential means to mitigate security risks in IoV. While current research primarily focuses on traditional security measures, the potential of Blockchain technology remains relatively unexplored in this domain. By leveraging the inherent properties of Blockchain, such as immutability, decentralized consensus, and cryptographic mechanisms, a more resilient and secure IoV ecosystem has been established.

Insights gained through a comprehensive analysis of existing security concerns and vulnerabilities in IoV in the Section 4, this implementation highlights the limitations of conventional security approaches and underscores the need for innovative solutions. It further delves into the fundamental principles of Blockchain-based DTs and their potential applicability in IoV security. Drawing upon insights from related industries and applications, the implemented framework for incorporating Blockchain-based DTs to bolster the security posture of IoV sheds light on the pressing security challenges facing IoV and advocates for a paradigm shift towards exploring the uncharted territory of Blockchain-based DTs as a promising avenue for safeguarding the integrity and confidentiality of IoV systems.

The implemented solution of integrating blockchain and DT technology into the IoV ecosystem holds real-life applicability in various aspects. For instance, when considering Privacy Concerns, implementing Ethereum's cryptographic mechanisms can ensure that sensitive data related to vehicle owners, location, and sensory data remains encrypted and accessible only to authorized parties. This would protect user privacy and prevent unauthorized tracking or misuse of personal information. Likewise, for ensuring sensory data integrity, the Ethereum and IPFS component of the implemented framework can safeguard critical information from tampering and unauthorized modifications. Historical sensory data can be securely stored on the Blockchain, providing an auditable and trustworthy source of information. In the context of V2S communication security, Ethereum's decentralized consensus protocol can prevent attackers from intercepting or manipulating communication. This would ensure the authenticity and accuracy of data and messages exchanged, enhancing the reliability of IoV applications like cooperative driving and traffic management.

Furthermore, the resilience offered by the Ethereum blockchain and IPFS can help mitigate DDoS attacks, as the decentralized nature of the network would make it harder for attackers to overload a single point of entry.

7.8 Evaluation

This section refers to our research method's third guideline, the evaluation (**DG3. Design Evaluation**). It describes the evaluation criteria and discusses the results by mapping them on the outcome of the developed blockchain-based DT solution. This thesis implemented the case study of the V2S communication business process. The primary purpose of the process is to eliminate fraudulent sensory data broadcasts by preventing malicious vehicles from manipulating the original data, getting real-time data for accurate analysis results, and faster detection of adversaries and attacks, which improves the data integrity and overall health of the IoV network.

7.8.1 Security

Ensuring the security of blockchain-based DT implementation in the V2S communication in the IoV context is of paramount importance due to the sensitive nature of the data and the potential risks associated with vehicle communication and control. We implemented a comprehensive security evaluation of blockchain-based DT technology in the IoV, focusing on privacy concerns, data integrity, smart contract vulnerabilities, scalability, and potential attacks. We have used Ethereum as a blockchain distributed ledger and IPFS for larger off-chain storage.

For our use case of V2S communication in IoV, we used the CIA triad (Confidentiality, Integrity, Availability) to first identify the assets which need protection. This approach ensures the security of valuable data without hindering the other security parameters. Table 20 illustrates the components in the proposed solution, which map to enforce the CIA triad. The **Confidentiality** principle assures that only authorized bodies can communicate with the system and access the data. It is mostly about the access control mechanism for the system's users. The **Integrity** principle contributes to data reliability by guaranteeing tamper-proof and correct state needed for assets.

Table 20. Mapping of CIA triad with study

CIA triad	Addressing component
Confidentiality	New vehicle is only allowed to communicate with the IoV region cloud after authorization by comparing its data from TA. Cryptographic keys for each vehicle are stored in Ethereum. RBAC is also maintained in the system. All historical sensory data used in the analysis phase is stored in blockchain and IPFS which are distributed and immutable.
Integrity	The integrity principle implements security protocols to ensure the authenticity of data, such as digital signatures or by hashing the data states. Only authorized vehicles are allowed to update and synchronize their DTs. Each data block stored in Ethereum can not be updated, only a new updated block is attached to the chain and hence, immutable data ensures the integrity of sensory data used by IoV services.
Availability	Ethereum blockchain network and IPFS aid to remove the deficiency of conventional IoV applications' dependency on central storage as they both are decentralized and distributed in nature.

Moreover, the **Availabilty** principle ensures that authorized bodies have access to data at any time they need it. Compared to other principles, availability is more critical because, without real-time sensory data, the whole system and resources go to waste. It is also dependent on physical resources, and hence it is harder to maintain as physical resources may get damaged, need maintenance, and require protection from natural phenomena.

7.8.2 Performance

Ethereum uses a gas-based fee system, where users pay for the computational resources required to execute their transactions or smart contracts. The cost of gas can be highly volatile, especially during times of network congestion, making it expensive for users to perform even simple operations. For our simple implementation, Ethereum has performed well with high scalability and fast transaction.

While smart contracts provide powerful functionality, they are executed sequentially on the Ethereum network. As a result, complex or resource-intensive smart contracts can significantly slow down the network and increase costs. Although in our case, we have used Ethereum 2.0. It is an extensive upgrade from Ethereum 1.0 aimed at transitioning from the current proof-of-work (PoW) consensus mechanism to proof-of-stake (PoS), introducing shard chains and other optimizations to enhance scalability. However, the full implementation and adoption of Ethereum 2.0 have taken time and are still ongoing. More limitations of the proposed solution are discussed in Section 8.

7.9 Summary

In this section, we answer the research question **RQ6. How to implement a blockchain-based DT solution to enhance the security of sensory data communication in V2S for IoV?**. We first implemented the proposed solution to answer the question using Azure DT services, Ethereum blockchain, and IPFS technology. In this section, we also defined the evaluation criteria on which the developed blockchain-based solution will evaluate. We searched and could not find any specific evaluation criteria for the blockchain-based enterprise software solution. Hence, we defined our criteria, considering the need for the solution and pain points mentioned in Section 7.8. Here we evaluated the solution outcome with the defined criteria and discussed how the proposed blockchain-based DT solution aids in mitigating the security risks associated with IoV. This section refers to the third guideline of the design science research method. According to the design science methodology, we can iterate repeatedly to improve the solution based on findings and measures achieved by evaluation after evaluating the solution. However, we considered only one iteration and concluded the study results.

8 Discussion

This thesis discusses the concept of DTs and their capabilities beyond 3D modeling. DTs assimilate real-time data and enable two-way communication with physical objects in virtual spaces. They find applicability throughout the lifecycle of products or systems, and blockchain-based DTs enhance these processes by ensuring data integrity and traceability, promoting collaboration and standardization. The thesis also addresses the context of the Internet of Vehicles (IoV) and the importance of securing sensory data and stakeholder information from potential threats. The lack of extensive research in securing IoV is highlighted, and the potential of blockchain-based DTs in enhancing IoV's sensory data communication security is discussed. By exploring security issues before deployment, vulnerabilities can be identified and risk mitigation strategies can be developed.

However, the thesis acknowledges that both blockchain and DTs have their security discrepancies. The thesis aims to find solutions to address these issues and improve the security of blockchain-based DTs for IoV sensory data communication. In this section, I discuss the results and outcomes of the study in the context of our research questions. Table 21 explains the general mapping of our study to our defined research questions.

Table 21. Mapping of Research Questions with Study

RQs	**Description**	**Section addressing RQ**
RQ1	What are the key building blocks of IoV?	Section 3 Building blocks of IoV
RQ2	What is the state-of-the-art in securing industrial systems using blockchain-based DT solutions?	Section 4 Literature Review
RQ3	What are the security risks in industrial systems and how blockchain-based DTs can be used to mitigate them?	Section 5 Security Risk Analysis
RQ4	How to design secure V2S data communication in IoV using Blockchain-based DT solution?	Chapter 6 Solution Design
RQ5	How to implement a blockchain-based DT solution to enhance the security of sensory data communication in V2S for IoV?	Section 7 Implementation

In this thesis, we first initialized the research with a generic research question **How to securely build a digital twin solution for the Internet of Vehicles using blockchain?** To answer this research question, for the thesis we divided it into five research questions each represented and addressed using their respective sections. The **RQ1** focuses on understanding the basic building blocks of IoV in Section 3. This section aims to provide the reader with a clear understanding of the necessity behind its pursuit, as well as the specific focus on V2S communication in the context of IoV as the chosen use case for this thesis. After the generic understanding of the building blocks of IoV, we focused on **RQ2** in Section 4. Since design science research methodology was utilized in this thesis, ensuring its contribution's viability necessitated alignment with the state-of-the-art. To achieve this, in this section, an SLR was conducted, serving as a guiding reference for our proposed solution. Section 5 answers the research question **RQ3**. This section plays a crucial role in achieving the thesis's objective of presenting a blockchain-based DT solution for IoV, following the guidelines of design science research methodology **DG3**, as mentioned in Section 1.4. The artifacts developed in this section are presented to address the research question effectively.

To address **RQ4: How to design secure V2S data communication in IoV using a Blockchain-based DT solution?**, in Section 6, we showcase the artifacts resulting from our research, which align with our chosen research method. These artifacts directly address the research question stated in Section 1.3. Furthermore, the artifact produced here plays a vital role in Section 7 and takes into account the threats and mitigation strategies discussed in Section 5.

Finally, in Section 7, we address **RQ5. How to implement a blockchain-based DT solution to enhance the security of sensory data communication in V2S for IoV?**. We first implemented the proposed solution to answer the question using Azure DT services, Ethereum blockchain, and IPFS technology. In this Section, we also defined our evaluation criteria on which the developed blockchain-based solution will evaluate. We evaluated the solution outcome with the defined criteria and discussed how the proposed blockchain-based DT solution aids in mitigating the security risks associated with IoV. This Section also refers to the third guideline of the design science research method.

8.1 Limitations

The main limitation is that the simulated implementation of the proposed solution is done using only the Azure platform which provides the IoT Hub and the ADT. Implementing other platforms and developing solutions to tackle various scenarios require much more time and resources. Ethereum as a blockchain used for the simulation is also a limitation. Complex channels, membership services, and robust role-based and attribute-based access control are required for the precarious threat landscape of the V2S communication scenario. One of the learnings from the implementation is that HLF as a blockchain would have been a better option for the scenario. The reason for using Ethereum was the unavailability of C# SDK for HLF. Building a full-fledged Decentralized Application (dApp) would require much more time and resources.

Moreover, most of the IPFS services are paid subscriptions. Local development of IPFS requires more time and computing resources. Another limitation is the number of region clouds, RSUs, vehicles, and analyzers with their DTs. A real-life scenario is much more complex, with many entities interacting concurrently in an IoV. There is also latency and time constraint due to the vehicles' high mobility, which is not considered in the simulation. The reason for that is the huge experiments, where we can test the solution's scalability, require more resources like funding, people to help with the integration and the configurations, and time.

However, The Azure documentation states that people use ADT to simulate complex physical environments like smart cities and so on [14]. This is quite an intriguing challenge to explore the scope of blockchain-based DTs as security enablers in such systems; thus, this is another future opportunity for other researchers with more resources. Azure also has an open code repository with smart city ontology [2].

8.2 Future Work

The proposed blockchain-based DT implementation for the IoV V2S communication scenario can be a stepping stone for future researchers. Tools like Eclipse Ditto [1] can be considered for future experiments. Local development of IPFS with HLF blockchain implementation is another aspect that can be explored.

For future work, this study and software solutions can help map the real-world data and provide results and insights to the readers to help in their research fieldwork. Also, according to the design research guidelines, we need multiple iterations to properly evaluate research, but we used one iteration. In the future, one can reiterate the research with complex data to properly evaluate the proposed blockchain-based DT solution for V2S communication in IoV.

Furthermore, various scalability issues (e.g., storage, throughput, latency) can be addressed by implementing the solution using Hyperledger Fabric (HLF) as the chosen blockchain technology, in combination with a local IPFS system.

9 Conclusion

The efficiency of security and data communication integrity in facile-secured Internet of Vehicles (IoV) networks is limited. There is a dearth of comprehensive research focused on securing IoV systems. This study investigates the potential of blockchain-based Digital Twins (DTs) in enhancing the security of sensory data communication within the IoV context. Data exchange security is critical within the communication channels of IoV. Utilizing blockchain-based DTs allows for pre-deployment security assessment, revealing potential security gaps and vulnerabilities. Moreover, this approach enables the exploration of risk mitigation strategies before implementation. By employing blockchain-based DTs, sensory data communication in IoV gains much-needed credibility. However, blockchain and DTs exhibit some security discrepancies, which this thesis aims to address by proposing solutions.

Nevertheless, in this thesis, We conducted the SLR where blockchain-based DTs have enhanced security in industrial systems. In the review, only seventeen papers were found where the authors combined blockchain and DT technology as security enablers in the systems. Only five papers related to the context of IoV with such cases exist. During this review, we also found social and technological issues facing the realization of using blockchain-based DTs as a countermeasure against security threats facing IoV. Additionally, we have proposed a novel blockchain-based DT framework for IoV and implemented it using the V2S communication scenario. The proposed solution proves the concept but has a lot of room for improvement and constraints. This was mainly due to the lack of time and resources for this thesis scope.

Moreover, we have done a security analysis on Blockchain-based DTs for IoV and the V2S communication scenario, where we have used STRIDE threat modeling to point out the threats, vulnerabilities, and impact of the security loopholes in the IoV system. Security risk management using the ISSRM model has also been done. Countermeasures using blockchain-based DTs have also been suggested. As a result of the introduction of blockchain and DT technologies, systems are also exposed to new security threats, which have been analyzed using STRIDE and ISSRM. Countermeasures were also discussed. We implemented the proposed framework application with self-compiled sample telemetry data for simulation. However, this solution can be used to calculate and benchmark the performance of blockchain-based DTs with the help of different tools and resources.

In conclusion, our proposed system strives to close the divide between theoretical comprehension and the actual application of blockchain-based DTs in ensuring security within the realm of the IoV. Through offering concrete remedies to real-world security complexities, this research endeavors to inspire various stakeholders, ranging from researchers and industry experts to policymakers, to adopt inventive methodologies and foster a safer and more secure environment for the IoV.

Acknowledgement

Grammarly

During the composition of this thesis, a digital writing assistant called Grammarly [17] was utilized. Grammarly is an AI-powered tool designed to assist users in enhancing their writing skills by offering suggestions for grammar, spelling, punctuation, clarity, and style. This tool is accessible through a web-based service, a browser extension, and a desktop application compatible with both Windows and Mac operating systems. Grammarly is capable of analyzing various types of writing, such as emails, essays, and reports. By employing sophisticated algorithms and natural language processing, it provides real-time suggestions, enabling users to make immediate corrections and enhance the overall quality of their writing as they progress. The key features of Grammarly are:

- *Spell and grammar checks*
- *Suggestions for proper punctuation*
- *Analysis of readability and clarity*
- *Enhancement of vocabulary*

[17] https://app.grammarly.com/

Furthermore, Grammarly provides a premium version that includes supplementary features, such as plagiarism detection, advanced grammar checks, and a more extensive array of style recommendations.

ChatGPT

In the course of writing this thesis, we have used ChatGPT [3] version V3.5. It is a type of AI language model developed by OpenAI [18]. It has only been used as a brainstorming assistant. It is based on the GPT (Generative Pre-trained Transformer) architecture, specifically GPT-3.5 is one of the most advanced language models, capable of understanding and generating human-like text based on the input it receives.

Additionally, ChatGPT is pre-trained on a massive amount of text data from the internet, which allows it to learn grammar, vocabulary, and various linguistic patterns. It can be fine-tuned for specific tasks, making it versatile and adaptable to a wide range of applications. People use ChatGPT for tasks such as customer support, content generation, programming help, language translation, and even just for having interactive conversations with an AI.

References

[1] Eclipse ditto™. https://www.eclipse.org/ditto/intro-overview.html, 2017.

[2] Digital twins definition language (dtdl) ontology for smart cities. https://github.com/Azure/opendigitaltwins-smartcities/tree/main/Ontology, 2021.

[3] Chatgpt v3.5, openai. https://https://chat.openai.com/, 2023.

[4] Sasikumar A., Subramaniyaswamy Vairavasundaram, Ketan Kotecha, Indragandhi V., Logesh Ravi, Ganeshsree Selvachandran, and Ajith Abraham. Blockchain-based trust mechanism for digital twin empowered industrial internet of things. *Future Generation Computer Systems*, 141:16–27, 2023.

[5] Wissam Abbass, Amine Baina, and Mostafa Bellafkih. Improvement of information system security risk management. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 182–187, 2016.

[6] Wissam Abbass, Amine Baina, and Mostafa Bellafkih. Survey on information system security risk management alignment. In *2016 International Conference on Information Technology for Organizations Development (IT4OD)*, pages 1–6, 2016.

[18] https://chat.openai.com/

[7] Kazi Masudul Alam and Abdulmotaleb El Saddik. C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE access*, 5:2050–2062, 2017.

[8] Cristina Alcaraz and Javier Lopez. Digital twin: A comprehensive survey of security threats. *IEEE Communications Surveys & Tutorials*, 24(3):1475–1503, 2022.

[9] Syed Anas Ansar, Archita Singh, Shruti Aggrawal, Amitabha Yadav, Prabhash Chandra Pathak, and Raees Ahmad Khan. Modernizing cps with blockchain: Applications, challenges & future directions. In *2022 Second International Conference on Interdisciplinary Cyber Physical Systems (ICPS)*, pages 124–129, 2022.

[10] Lixia Bao, Qiulan Wang, and Yan Jiang. Review of digital twin for intelligent transportation system. In *2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT)*, pages 309–315, 2021.

[11] Barbara Rita Barricelli, Elena Casiraghi, and Daniela Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE access*, 7:167653–167671, 2019.

[12] F. Rizal Batubara, Jolien Ubacht, and Marijn Janssen. Challenges of blockchain technology adoption for e-government: A systematic literature review. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, dg.o '18, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Maurizio Bevilacqua, Eleonora Bottani, Filippo Emanuele Ciarapica, Francesco Costantino, Luciano Di Donato, Alessandra Ferraro, Giovanni Mazzuto, Andrea Monteriù, Giorgia Nardini, Marco Ortenzi, Massimo Paroncini, Marco Pirozzi, Mario Prist, Elena Quatrini, Massimo Tronci, and Giuseppe Vignali. Digital twin reference model development to prevent operators' risk in process plants. *Sustainability*, 12(3), 2020.

[14] Olivier Bloch. Smart cities ontology for digital twins. `https://learn.microsoft.com/en-us/shows/internet-of-things-show/smart-cities-ontology-for-digital-twins`, 2021.

[15] Theodor Borangiu, Ecaterina Oltean, Silviu Răileanu, Florin Anton, Silvia Anton, and Iulia Iacob. Embedded digital twin for arti-type control of semi-continuous production processes. In *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future: Proceedings of SOHOMA 2019 9*, pages 113–133. Springer, 2020.

[16] Jesus David Chaux, David Sanchez-Londono, and Giacomo Barbieri. A digital twin architecture to optimize productivity within controlled environment agriculture. *Applied Sciences*, 11(19):8875, 2021.

[17] JiuJun Cheng, JunLu Cheng, MengChu Zhou, FuQiang Liu, ShangCe Gao, and Cong Liu. Routing in internet of vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2339–2352, 2015.

[18] Juan Contreras-Castillo, Sherali Zeadally, and Juan Antonio Guerrero-Ibañez. Internet of vehicles: Architecture, protocols, and security. *IEEE Internet of Things Journal*, 5(5):3701–3709, 2018.

[19] Tianhu Deng, Keren Zhang, and Zuo-Jun (Max) Shen. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. *Journal of Management Science and Engineering*, 6(2):125–134, 2021.

[20] Marietheres Dietz and Günther Pernul. Digital twin: Empowering enterprises towards a system-of-systems approach. *Business & Information Systems Engineering*, 62:179–184, 04 2020.

[21] Éric Dubois, Patrick Heymans, Nicolas Mayer, and Raimundas Matulevičius. A systematic approach to define the domain of information system security risk management. *Intentional perspectives on information systems engineering*, pages 289–306, 2010.

[22] Matthias Eckhart and Andreas Ekelhart. *Digital Twins for Cyber-Physical Systems Security: State of the Art and Outlook*, pages 383–412. 11 2019.

[23] Philip Empl and Günther Pernul. Digital-twin-based security analytics for the internet of things. *Information*, 14(2), 2023.

[24] Philip Empl, Daniel Schlette, Daniel Zupfer, and Günther Pernul. Soar4iot: Securing iot assets with digital twins. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, ARES '22, New York, NY, USA, 2022. Association for Computing Machinery.

[25] John Ahmet Erkoyuncu, Maryam Farsi, Dedy Ariansyah, et al. An intelligent agent-based architecture for resilient digital twins in manufacturing. *CIRP annals*, 70(1):349–352, 2021.

[26] Tolga Erol, Arif Furkan Mendi, and Dilara Doğan. The digital twin revolution in healthcare. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–7, 2020.

[27] Maryam Farsi, Dedy Ariansyah, John Ahmet Erkoyuncu, and Andrew Harrison. A digital twin architecture for effective product lifecycle cost estimation. *Procedia CIRP*, 100:506–511, 2021.

[28] Hailin Feng, Dongliang Chen, and Zhihan Lv. Blockchain in digital twins-based vehicle management in vanets. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19613–19623, 2022.

[29] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8:108952–108971, 2020.

[30] Daniel Ganji, Christos Kalloniatis, Haris Mouratidis, and Saeed Malekshahi Gheytassi. Approaches to develop and implement iso/iec 27001 standard - information security management systems: A systematic literature review. 12:228–238, 12 2019.

[31] Chaouki Ghenai, Lama Alhaj Husein, Marwa Al Nahlawi, Abdul Kadir Hamid, and Maamar Bettayeb. Recent trends of digital twin technologies in the energy sector: A comprehensive review. *Sustainable Energy Technologies and Assessments*, 54:102837, 2022.

[32] Mezzour Ghita, Benhadou Siham, and Medromi Hicham. Digital twins development architectures and deployment technologies: Moroccan use case. *International Journal of Advanced Computer Science and Applications*, 11(2), 2020.

[33] Edward Glaessgen and David Stargel. The digital twin paradigm for future nasa and u.s. air force vehicles. 04 2012.

[34] Michael Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.

[35] Michael Grieves and John Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems: New findings and approaches*, pages 85–113, 2017.

[36] Jiajie Guo, Muhammad Bilal, Yuying Qiu, Cheng Qian, Xiaolong Xu, and Kim-Kwang Raymond Choo. Survey on digital twins for internet of vehicles: Fundamentals, challenges, and opportunities. *Digital Communications and Networks*, 2022.

[37] Haya R. Hasan, Khaled Salah, Raja Jayaraman, Mohammed Omar, Ibrar Yaqoob, Saša Pesic, Todd Taylor, and Dragan Boscovic. A blockchain-based approach for the creation of digital twins. *IEEE Access*, 8:34113–34126, 2020.

[38] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *Management Information Systems Quarterly*, 28(1):6, 2008.

[39] Michael Howard and Steve Lipner. *The security development lifecycle*, volume 8. Microsoft Press Redmond, 2006.

[40] Mubashar Iqbal and Raimundas Matulevičius. Exploring sybil and double-spending risks in blockchain systems. *IEEE Access*, 9:76153–76177, 2021.

[41] Baofeng Ji, Xueru Zhang, Shahid Mumtaz, Congzheng Han, Chunguo Li, Hong Wen, and Dan Wang. Survey on the internet of vehicles: Network architectures and applications. *IEEE Communications Standards Magazine*, 4(1):34–41, 2020.

[42] Klementina Josifovska, Enes Yigitbas, and Gregor Engels. Reference framework for digital twins within cyber-physical systems. In *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*, pages 25–31. IEEE, 2019.

[43] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11):1016–1022, 2018. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[44] Nir Kshetri. Blockchain's roles in strengthening cybersecurity and protecting privacy. *Telecommunications policy*, 41(10):1027–1038, 2017.

[45] Siyi Liao, Jun Wu, Ali Kashif Bashir, Wu Yang, Jianhua Li, and Usman Tariq. Digital twin consensus for blockchain-enabled intelligent transportation systems in smart cities. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22619–22629, 2022.

[46] Jun Liu, Lei Zhang, Chunlin Li, Jingpan Bai, Haibin Lv, and Zhihan Lv. Blockchain-based secure communication of intelligent transportation digital twins system. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22630–22640, 2022.

[47] Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58:346–361, 2021.

[48] Yunlong Lu, Xiaohong Huang, Ke Zhang, Sabita Maharjan, and Yan Zhang. Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks. *IEEE Transactions on Industrial Informatics*, 17(7):5098–5107, 2021.

[49] Nicolas Mayer, Eric Dubois, Raimundas Matulevicius, and Patrick Heymans. Towards a measurement framework for security risk management. In *MODSEC@ MoDELS*, 2008.

[50] Gary McGraw. Software security. *IEEE Security & Privacy*, 2(2):80–83, 2004.

[51] Ritesh Modi. *Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain*. Packt Publishing Ltd, 2018.

[52] Dimitris Mourtzis, Thodoris Togias, John Angelopoulos, and Panos Stavropoulos. A digital twin architecture for monitoring and optimization of fused deposition modeling processes. *Procedia CIRP*, 103:97–102, 2021.

[53] Satoshi Nakamoto and A Bitcoin. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4(2), 2008.

[54] Chukwudi Nwogu, Giovanni Lugaresi, Anastasia Anagnostou, Andrea Matta, and Simon J.E. Taylor. Towards a requirement-driven digital twin architecture. *Procedia CIRP*, 107:758–763, 2022. Leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022.

[55] Morgen E Peck. Blockchains: How they work and why they'll change the world. *IEEE spectrum*, 54(10):26–35, 2017.

[56] Iakovos Pittaras and George C. Polyzos. Secure and efficient web of things digital twins using permissioned blockchains. In *2022 7th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–5, 2022.

[57] AJH Redelinghuys, Anton Herman Basson, and Karel Kruger. A six-layer architecture for the digital twin: a manufacturing case study implementation. *Journal of Intelligent Manufacturing*, 31:1383–1402, 2020.

[58] Matthew Sadiku, Mahamadou Tembely, and Sarhan Musa. Internet of vehicles: An introduction. *International Journal of Advanced Research in Computer Science and Software Engineering*, 8:11, 02 2018.

[59] Radhya Sahal, Saeed H. Alsamhi, Kenneth N. Brown, Donna O'Shea, Conor McCarthy, and Mohsen Guizani. Blockchain-empowered digital twins collaboration: Smart transportation use case. *Machines*, 9(9), 2021.

[60] Mikail Mohammed Salim, Alowonou Kowovi Comivi, Tojimurotov Nurbek, Heejae Park, and Jong Hyuk Park. A blockchain-enabled secure digital twin framework for early botnet detection in iiot environment. *Sensors*, 22(16), 2022.

[61] Priyank Sharma, Meet Patel, and Apoorva Prasad. A systematic literature review on internet of vehicles security, 2022.

[62] Surbhi Sharma and Baijnath Kaushik. A survey on internet of vehicles: Applications, security issues & solutions. *Vehicular Communications*, 20:100182, 2019.

[63] Maulshree Singh, Evert Fuenmayor, Eoin P. Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. Digital twin: Origin to future. *Applied System Innovation*, 4(2), 2021.

[64] Daniela Soares Cruzes, Martin Gilje Jaatun, Karin Bernsmed, and Inger Anne Tøndel. Challenges and experiences with applying microsoft threat modeling in agile development projects. In *2018 25th Australasian Software Engineering Conference (ASWEC)*, pages 111–120, 2018.

[65] Vinicius Souza, Robson Cruz, Walmir Silva, Sidney Lins, and Vicente Lucena. A digital twin architecture based on the industrial internet of things technologies. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2. IEEE, 2019.

[66] Gernot Steindl, Martin Stagl, Lukas Kasper, Wolfgang Kastner, and Rene Hofmann. Generic digital twin architecture for industrial energy systems. *Applied Sciences*, 10(24):8903, 2020.

[67] Sabah Suhail, Rasheed Hussain, Raja Jurdak, and Choong Seon Hong. Trustworthy digital twins in the industrial internet of things with blockchain. *IEEE Internet Computing*, 26(3):58–67, 2022.

[68] Sabah Suhail, Rasheed Hussain, Raja Jurdak, Alma Oracevic, Khaled Salah, Choong Seon Hong, and Raimundas Matulevičius. Blockchain-based digital twins: Research trends, issues, and future challenges. *ACM Comput. Surv.*, 54(11s), sep 2022.

[69] Sabah Suhail and Raja Jurdak. Towards trusted and intelligent cyber-physical systems: A security-by-design approach, 2022.

[70] Sabah Suhail, Saif Ur Rehman Malik, Raja Jurdak, Rasheed Hussain, Raimundas Matulevičius, and Davor Svetinovic. Towards situational aware cyber-physical systems: A security-enhancing use case of blockchain-based digital twins. *Computers in Industry*, 141:103699, 2022.

[71] Chenchen Tan, Xinghao Li, Tom H. Luan, Bruce Gu, Youyang Qu, and Longxiang Gao. Digital twin based remote resource sharing in internet of vehicles using consortium blockchain. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 1–6, 2021.

[72] Karl Wüst and Arthur Gervais. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54, 2018.

[73] Ibrar Yaqoob, Khaled Salah, Mueen Uddin, Raja Jayaraman, Mohammed Omar, and Muhammad Imran. Blockchain for digital twins: Recent advances and future research challenges. *IEEE Network*, 34(5):290–298, 2020.

[74] Chao Zhang, Guanghui Zhou, Han Li, and Yan Cao. Manufacturing blockchain of things for the configuration of a data- and knowledge-driven digital twin manufacturing cell. *IEEE Internet of Things Journal*, 7(12):11884–11894, 2020.

Appendix

A Resources

Code Repository: `https://github.com/ashfaqshuvo007/thesis-iov-v2s-adt`

Demo Videos

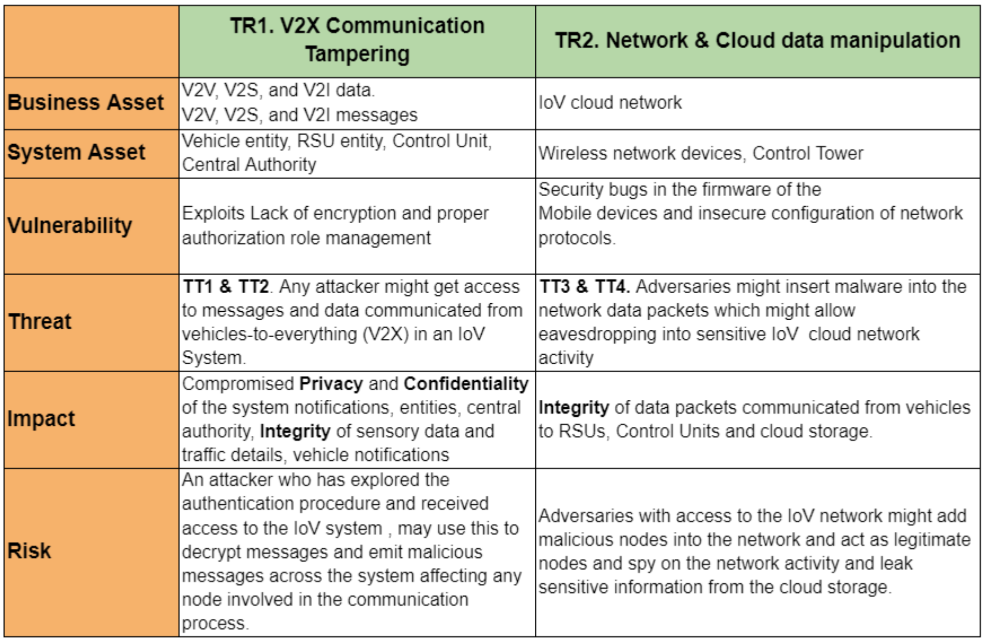
Data Simulators - Simulators Update digital twin in ADT explorer
`https://youtu.be/9dYQHXv6FgE`

Data Ingestors - Interaction with blockchain and ADT explorer
`https://youtu.be/R-YqUxDT-Vk`

Creating Digital Twins from DTDL - Microsoft ADT Explorer
`https://youtu.be/2sKrl3EReFM`

B Detailed Security Risk Model

Figure 35. Risk model IoV - Tampering Risks

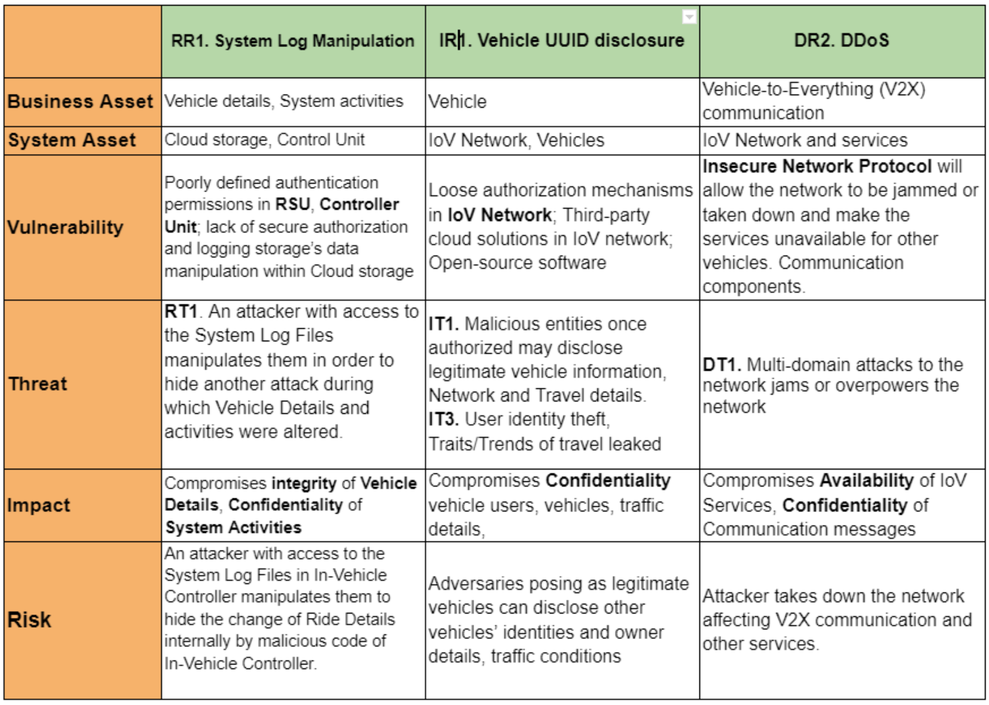


	TR1. V2X Communication Tampering	**TR2. Network & Cloud data manipulation**
Business Asset	V2V, V2S, and V2I data. V2V, V2S, and V2I messages	IoV cloud network
System Asset	Vehicle entity, RSU entity, Control Unit, Central Authority	Wireless network devices, Control Tower
Vulnerability	Exploits Lack of encryption and proper authorization role management	Security bugs in the firmware of the Mobile devices and insecure configuration of network protocols.
Threat	**TT1 & TT2**. Any attacker might get access to messages and data communicated from vehicles-to-everything (V2X) in an IoV System.	**TT3 & TT4.** Adversaries might insert malware into the network data packets which might allow eavesdropping into sensitive IoV cloud network activity
Impact	Compromised **Privacy** and **Confidentiality** of the system notifications, entities, central authority, **Integrity** of sensory data and traffic details, vehicle notifications	**Integrity** of data packets communicated from vehicles to RSUs, Control Units and cloud storage.
Risk	An attacker who has explored the authentication procedure and received access to the IoV system , may use this to decrypt messages and emit malicious messages across the system affecting any node involved in the communication process.	Adversaries with access to the IoV network might add malicious nodes into the network and act as legitimate nodes and spy on the network activity and leak sensitive information from the cloud storage.

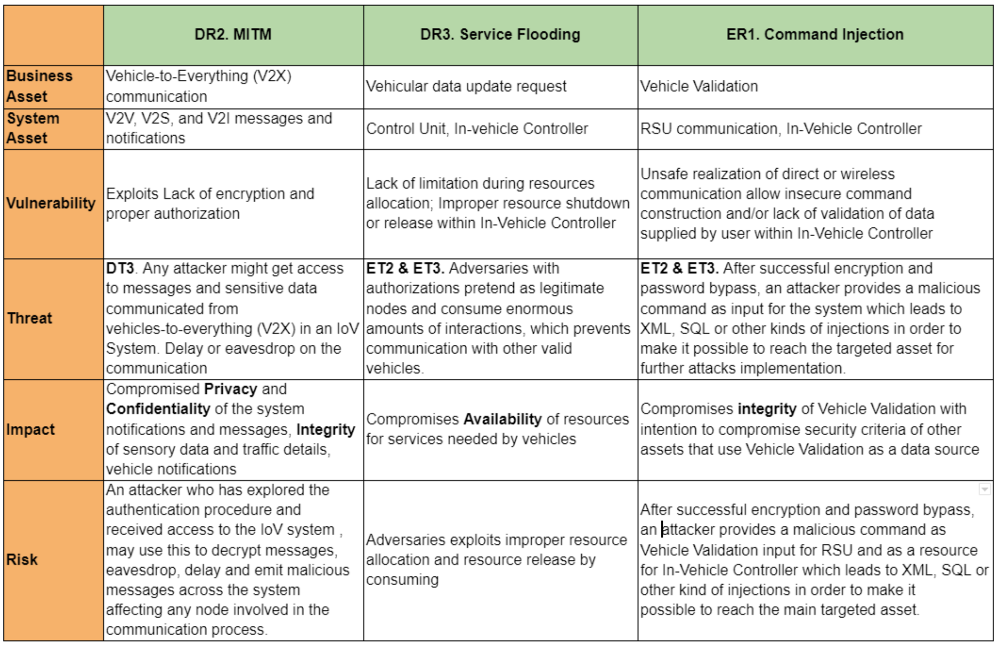
Table 22. Risk Model IoV - Spoofing Risks

	SR1. Vehicle Identity Spoofing	**SR2**.Vehicle Position (GPS) Spoofing
Business Asset	V2X (vehicle-to-everything) notifications, Vehicle details, Sensory data Update	Vehicle Routes, Sensory data Update
System Asset	Vehicle entity	Vehicle entity, RSU entitiy
Vulnerability	Lack of keeping logs of every authentication of vehicles in the IoV Network	Exploits Lack of keeping logs of every vehicle route, Insecure storage of travel data
Threat	**ST1.** Any attacker might pose as a legitimate vehicle after gaining access to the network	**ST2.** Any attacker might get access to travel routes, patterns, or daily schedule
Impact	Compromised **Privacy** and **Confidentiality** of V2X notifications, **Integrity** of sensory data and Vehicle details	Compromised **Privacy** and **Confidentiality** of the vehicle owner, **Integrity** of sensory data and traffic details
Risk	An attacker who has explored the authentication procedure and received access to the IoV system uses obtained credentials to authenticate himself as a legitimate Vehicles to the network that compromises the confidentiality of Passenger Notification and as a consequence Ride Details as well as loss of reliability (i.e. integrity) of any Ride Update Request.	An attacker who has explored the authentication procedure and received access to the IoV system may change the route of the travel. Traffic update might lose their integrity and confidentiality

Figure 36. Risk model IoV - Repudiation & DDoS



	RR1. System Log Manipulation	**IR1. Vehicle UUID disclosure**	**DR2. DDoS**
Business Asset	Vehicle details, System activities	Vehicle	Vehicle-to-Everything (V2X) communication
System Asset	Cloud storage, Control Unit	IoV Network, Vehicles	IoV Network and services
Vulnerability	Poorly defined authentication permissions in **RSU**, **Controller Unit**; lack of secure authorization and logging storage's data manipulation within Cloud storage	Loose authorization mechanisms in **IoV Network**; Third-party cloud solutions in IoV network; Open-source software	**Insecure Network Protocol** will allow the network to be jammed or taken down and make the services unavailable for other vehicles. Communication components.
Threat	**RT1**. An attacker with access to the System Log Files manipulates them in order to hide another attack during which Vehicle Details and activities were altered.	**IT1.** Malicious entities once authorized may disclose legitimate vehicle information, Network and Travel details. **IT3.** User identity theft, Traits/Trends of travel leaked	**DT1.** Multi-domain attacks to the network jams or overpowers the network
Impact	Compromises **integrity** of **Vehicle Details**, **Confidentiality** of **System Activities**	Compromises **Confidentiality** vehicle users, vehicles, traffic details,	Compromises **Availability** of IoV Services, **Confidentiality** of Communication messages
Risk	An attacker with access to the System Log Files in In-Vehicle Controller manipulates them to hide the change of Ride Details internally by malicious code of In-Vehicle Controller.	Adversaries posing as legitimate vehicles can disclose other vehicles' identities and owner details, traffic conditions	Attacker takes down the network affecting V2X communication and other services.



	DR2. MITM	DR3. Service Flooding	ER1. Command Injection
Business Asset	Vehicle-to-Everything (V2X) communication	Vehicular data update request	Vehicle Validation
System Asset	V2V, V2S, and V2I messages and notifications	Control Unit, In-vehicle Controller	RSU communication, In-Vehicle Controller
Vulnerability	Exploits Lack of encryption and proper authorization	Lack of limitation during resources allocation; Improper resource shutdown or release within In-Vehicle Controller	Unsafe realization of direct or wireless communication allow insecure command construction and/or lack of validation of data supplied by user within In-Vehicle Controller
Threat	**DT3**. Any attacker might get access to messages and sensitive data communicated from vehicles-to-everything (V2X) in an IoV System. Delay or eavesdrop on the communication	**ET2 & ET3.** Adversaries with authorizations pretend as legitimate nodes and consume enormous amounts of interactions, which prevents communication with other valid vehicles.	**ET2 & ET3.** After successful encryption and password bypass, an attacker provides a malicious command as input for the system which leads to XML, SQL or other kinds of injections in order to make it possible to reach the targeted asset for further attacks implementation.
Impact	Compromised **Privacy** and **Confidentiality** of the system notifications and messages, **Integrity** of sensory data and traffic details, vehicle notifications	Compromises **Availability** of resources for services needed by vehicles	Compromises **integrity** of Vehicle Validation with intention to compromise security criteria of other assets that use Vehicle Validation as a data source
Risk	An attacker who has explored the authentication procedure and received access to the IoV system , may use this to decrypt messages, eavesdrop, delay and emit malicious messages across the system affecting any node involved in the communication process.	Adversaries exploits improper resource allocation and resource release by consuming	After successful encryption and password bypass, an attacker provides a malicious command as Vehicle Validation input for RSU and as a resource for In-Vehicle Controller which leads to XML, SQL or other kind of injections in order to make it possible to reach the main targeted asset.

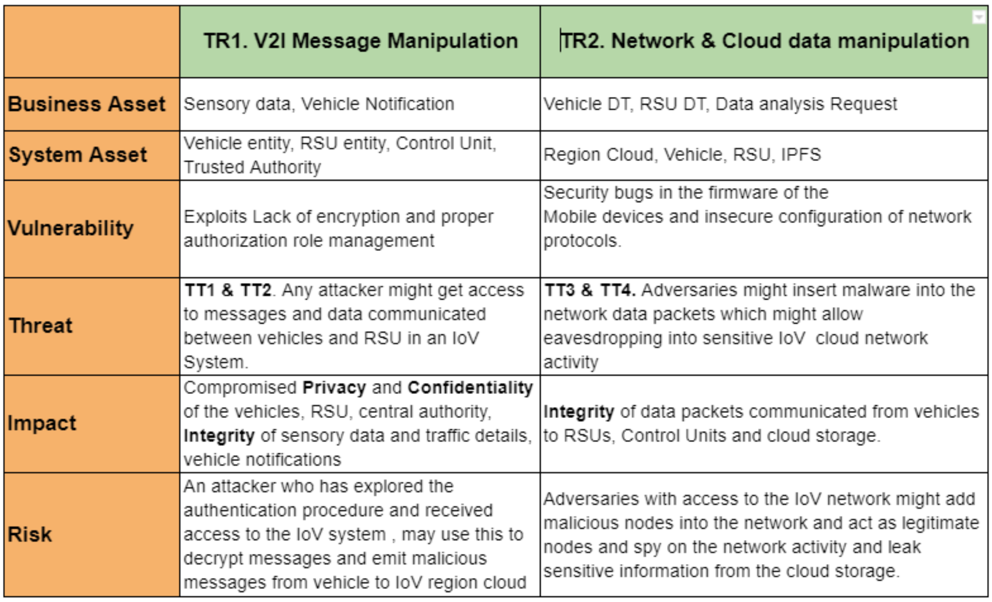
Figure 37. Risk model IoV - DDoS & Escalation of Privileges

C Detailed Security Risk Model - V2S communication

Table 23. Security Risk Model - Vehicle-To-Sensor communication

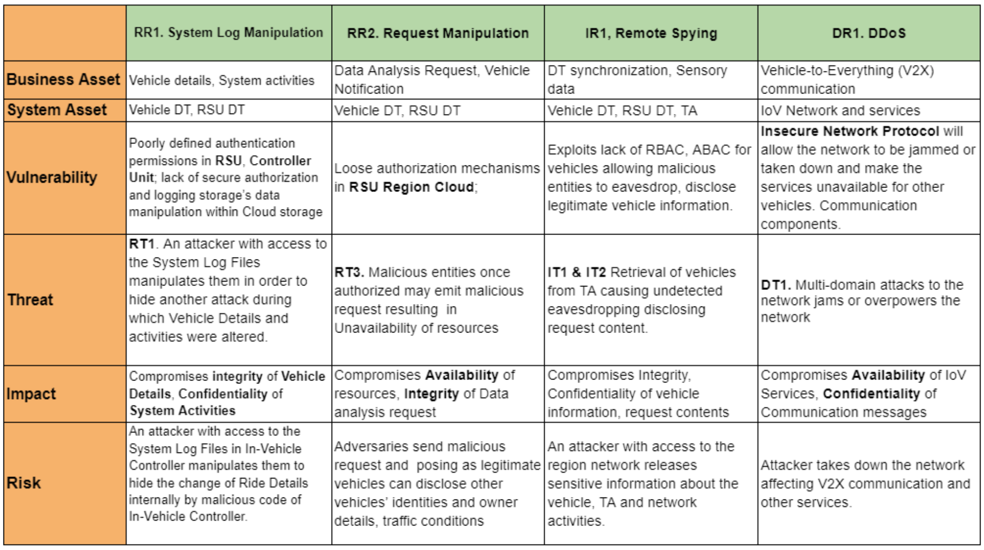
	SR1. Vehicle Identity Spoofing	**SR2**.Sybil Attack
Business Asset	V2X (vehicle-to-everything) notifications, Vehicle details, Sensory data Update	Vehicle Details, On-board Unit, Road-side Unit
System Asset	Vehicle entity	Vehicle entity, RSU
Vulnerability	Lack of keeping logs of every authentication of vehicles in the IoV Network	Exploits Lack of proper authorization, role-specific attributes, Insecure storage
Threat	**ST1.** Any attacker might pose as a legitimate vehicle after gaining access to the network	**ST2.** Any attacker might pose as legitimate vehicles and get access to travel routes, patterns, or daily schedule
Impact	Compromised **Privacy** and **Confidentiality** of V2X notifications, **Integrity** of sensory data and Vehicle details	Compromised **Privacy** and **Confidentiality** of vehicle details, **Integrity** of sensory data and traffic details
Risk	An attacker who has explored the authentication procedure and received access to the IoV system uses obtained credentials to authenticate himself as a legitimate Vehicles to the network that compromises the confidentiality of Passenger Notification and as a consequence Ride Details as well as loss of reliability (i.e. integrity) of any Ride Update Request.	An attacker who has explored the authentication procedure and received access to the IoV system may change the route of the travel. Traffic updates might lose their integrity and confidentiality

Figure 38. Detailed Security Risk Model - V2S communication



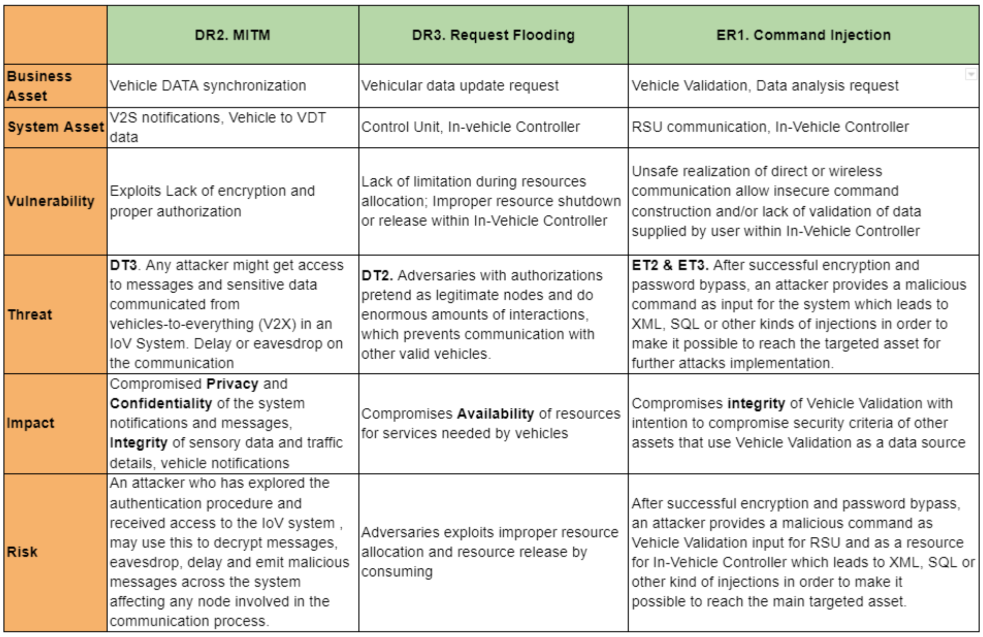
	TR1. V2I Message Manipulation	**TR2. Network & Cloud data manipulation**
Business Asset	Sensory data, Vehicle Notification	Vehicle DT, RSU DT, Data analysis Request
System Asset	Vehicle entity, RSU entity, Control Unit, Trusted Authority	Region Cloud, Vehicle, RSU, IPFS
Vulnerability	Exploits Lack of encryption and proper authorization role management	Security bugs in the firmware of the Mobile devices and insecure configuration of network protocols.
Threat	**TT1 & TT2**. Any attacker might get access to messages and data communicated between vehicles and RSU in an IoV System.	**TT3 & TT4.** Adversaries might insert malware into the network data packets which might allow eavesdropping into sensitive IoV cloud network activity
Impact	Compromised **Privacy** and **Confidentiality** of the vehicles, RSU, central authority, **Integrity** of sensory data and traffic details, vehicle notifications	**Integrity** of data packets communicated from vehicles to RSUs, Control Units and cloud storage.
Risk	An attacker who has explored the authentication procedure and received access to the IoV system , may use this to decrypt messages and emit malicious messages from vehicle to IoV region cloud	Adversaries with access to the IoV network might add malicious nodes into the network and act as legitimate nodes and spy on the network activity and leak sensitive information from the cloud storage.

Figure 39. Detailed Security Risk Model - V2S communication



	RR1. System Log Manipulation	**RR2. Request Manipulation**	**IR1, Remote Spying**	**DR1. DDoS**
Business Asset	Vehicle details, System activities	Data Analysis Request, Vehicle Notification	DT synchronization, Sensory data	Vehicle-to-Everything (V2X) communication
System Asset	Vehicle DT, RSU DT	Vehicle DT, RSU DT	Vehicle DT, RSU DT, TA	IoV Network and services
Vulnerability	Poorly defined authentication permissions in **RSU**, **Controller Unit**; lack of secure authorization and logging storage's data manipulation within Cloud storage	Loose authorization mechanisms in **RSU Region Cloud**;	Exploits lack of RBAC, ABAC for vehicles allowing malicious entities to eavesdrop, disclose legitimate vehicle information.	**Insecure Network Protocol** will allow the network to be jammed or taken down and make the services unavailable for other vehicles. Communication components.
Threat	**RT1**. An attacker with access to the System Log Files manipulates them in order to hide another attack during which Vehicle Details and activities were altered.	**RT3.** Malicious entities once authorized may emit malicious request resulting in Unavailability of resources	**IT1 & IT2** Retrieval of vehicles from TA causing undetected eavesdropping disclosing request content.	**DT1.** Multi-domain attacks to the network jams or overpowers the network
Impact	Compromises **integrity** of **Vehicle Details**, **Confidentiality** of **System Activities**	Compromises **Availability** of resources, **Integrity** of Data analysis request	Compromises Integrity, Confidentiality of vehicle information, request contents	Compromises **Availability** of IoV Services, **Confidentiality** of Communication messages
Risk	An attacker with access to the System Log Files in In-Vehicle Controller manipulates them to hide the change of Ride Details internally by malicious code of In-Vehicle Controller.	Adversaries send malicious request and posing as legitimate vehicles can disclose other vehicles' identities and owner details, traffic conditions	An attacker with access to the region network releases sensitive information about the vehicle, TA and network activities.	Attacker takes down the network affecting V2X communication and other services.

Figure 40. Detailed Security Risk Model - V2S communication



	DR2. MITM	**DR3. Request Flooding**	**ER1. Command Injection**
Business Asset	Vehicle DATA synchronization	Vehicular data update request	Vehicle Validation, Data analysis request
System Asset	V2S notifications, Vehicle to VDT data	Control Unit, In-vehicle Controller	RSU communication, In-Vehicle Controller
Vulnerability	Exploits Lack of encryption and proper authorization	Lack of limitation during resources allocation; Improper resource shutdown or release within In-Vehicle Controller	Unsafe realization of direct or wireless communication allow insecure command construction and/or lack of validation of data supplied by user within In-Vehicle Controller
Threat	**DT3**. Any attacker might get access to messages and sensitive data communicated from vehicles-to-everything (V2X) in an IoV System. Delay or eavesdrop on the communication	**DT2.** Adversaries with authorizations pretend as legitimate nodes and do enormous amounts of interactions, which prevents communication with other valid vehicles.	**ET2 & ET3.** After successful encryption and password bypass, an attacker provides a malicious command as input for the system which leads to XML, SQL or other kinds of injections in order to make it possible to reach the targeted asset for further attacks implementation.
Impact	Compromised **Privacy** and **Confidentiality** of the system notifications and messages, **Integrity** of sensory data and traffic details, vehicle notifications	Compromises **Availability** of resources for services needed by vehicles	Compromises **integrity** of Vehicle Validation with intention to compromise security criteria of other assets that use Vehicle Validation as a data source
Risk	An attacker who has explored the authentication procedure and received access to the IoV system , may use this to decrypt messages, eavesdrop, delay and emit malicious messages across the system affecting any node involved in the communication process.	Adversaries exploits improper resource allocation and resource release by consuming	After successful encryption and password bypass, an attacker provides a malicious command as Vehicle Validation input for RSU and as a resource for In-Vehicle Controller which leads to XML, SQL or other kind of injections in order to make it possible to reach the main targeted asset.

D Licence

Non-exclusive license to reproduce thesis and make thesis public

I, **Ashfaq Hussain Ahmed**,

1. herewith grant the University of Tartu a free permit (non-exclusive license) to

   reproduce, for preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Harnessing Blockchain and Digital Twin for Security Risk Assessment in Internet of Vehicles,

   supervised by Dr. Mubashar Iqbal and Dr. Sabah Suhail.

2. we grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons license CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. we are aware of the fact that the author retains the rights specified in p. 1 and 2.

4. we certify that granting the non-exclusive license does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Ashfaq Hussain Ahmed
11/08/2023