Technical Progress Report

”Digital Twin Interoperability for Space Exploration Missions Using the Spatial Web”

Alicia Sanjurjo Barrio, Edward Chow, Thomas Lu

April 2025

ABSTRACT

As lunar exploration evolves into a multi-stakeholder endeavor involving government agencies, private companies, and international partners, the need for interoperable and secure digital twin systems has become critical. This thesis investigates the potential of the Spatial Web standard—specifically the Hyperspatial Modeling Language (HSML) and Hyperspatial Transaction Protocol (HSTP)—to enable real-time, cross-platform interoperability between digital twins in simulated space mission environments. Addressing key gaps in current digital twin architectures, the study develops and implements a unified framework that includes a shared ontology (HSML), a secure RESTful API, and decentralized identity verification using the DID:key method.

Two sets of experiments were conducted using Unity, Unreal Engine, and Omniverse Isaac Sim. The first demonstrated that structured HSML messages allow digital twins across different platforms to exchange data in real time. The second integrated authentication and authorization mechanisms, validating secure communication without compromising system performance. Results confirm that the Spatial Web can support decentralized, reliable, and efficient digital twin coordination, which is an essential capability for future lunar operations where agents from different organizations must interact and share mission-critical data.

This work offers a proof of concept for applying Spatial Web principles to space systems, and proposes recommendations for future development, including schema extensions, automated integration tools, and enhanced governance mechanisms.

Digital Twins, Interoperability, Spatial Web, Lunar Missions, Decentralized Identity, Web 3.0, HSML, HSTP

CONTENTS

1. INTRODUCTION 1

1.1. Introduction 1

1.2. Statement of the Problem 2

1.3. Background and Need 3

1.3.1. The Moon as the next frontier..4

1.3.2. The rise of Digital Twins..6

1.3.3. The path to interoperability with the Spatial Web 9

1.4. Purpose of the Study..10

1.5. Research Questions 11

1.6. Significance to the Field 12

1.7. Limitations 13

1.8. Ethical Considerations..13

1.9. Structure of the Thesis..14

2. REVIEW OF RELATED WORK 15

2.1. Space Exploration: Communication and Collaboration 15

2.1.1. Why is Interoperability needed and important now in the context of Moon Exploration 15

2.1.2. munication..17

2.2. Digital Twins in Space: Opportunities and Challenges 18

2.2.1. Digital Twins in the Space Industry..19

2.2.2. Challenges in Digital Twin Implementation for Space Exploration..20

2.2.3. Evaluating Digital Twins Interoperability..22

2.2.4. Interoperability Among Digital Twin Development Platforms..23

2.3. The Spatial Web: Interoperability, Security, and Implementation 25

2.3.1. The Value of the Spatial Web for Digital Twins 25

2.3.2. Implementation of the Spatial Web Architecture 27

2.3.3. Spatial Web Challenges for Digital Twins Interoperability 30

3. METHODS 32

3.1. Overview 32

3.2. HSML Schema Development 33

3.3. HSTP Communication Protocol Implementation 36

3.3.1. The SWID & the DID:key Method 36

3.3.2. MySQL Database for Storage 37

3.3.3. The Verification Logic 38

3.3.4. The HSML API 41

3.4. Experimental Setup 43

3.4.1. Simulation Environment & Tools 44

3.4.2. Simulation Scenarios 47

3.5. Data Collection & Analysis 51

3.5.1. Logging Mechanisms & Analysis Tools 51

3.5.2. Evaluation Metrics 52

3.6. Validation & Limitations 52

4. RESULTS AND ANALYSIS 54

4.1. HSML Schema Results and Discussion 54

4.2. Unit Tests & HSML API Evaluation 57

4.3. Experiment 1: Cross-Platform Interoperability Using HSML Schema 59

4.4. Experiment 2: HSTP-Based Cross-Platform Interoperability with HSML API 63

4.5. Summary & Insights 67

5. CONCLUSIONS AND RECOMMENDATIONS 69

5.1. Conclusions 69

5.2. Recommendations 71

BIBLIOGRAPHY...72

LIST OF FIGURES

1.1 NASA's Artemis Mission 5

1.2 Conceptual Flow Between Real Space and Digital Twins 7

1.3 The Spatial Web Protocols 10

2.1 NASA's 13 candidate landing regions for Artemis III mission. 17

2.2 Map of the Entity subtypes that conform the HSML ontology 29

3.1 HSML Ontology Classes Hierarchy and their correspondent Schema.org Types 34

3.2 Example of an HSML JSON File for an Agent 35

3.3 Minimum Entities Needed to Register to Test Verification Logic.....................40

3.4 HSML API Architecture Block Diagram...42

3.5 Experimental Setup Graph...47

3.6 Lunar rover collaboration use case with three simulated agents. 48

3.7 Architecture of Experiment 2 — Cross-Platform Interoperability Using HSML Schema 49

3.8 Architecture of the Preliminary Registration through the HSML Web App & HSML API...50

3.9 ability with HSML API...51

4.1 Experiment 1 Cross-platform Simulation Screenshot 59

4.2 Latency Between HSML Messages Sent Directly Thorugh Kafka 62

4.3 Experiment 2 Cross-platform Simulation with HSML API Screenshot............64

4.4 Latency Between HSML Messages Sent Thorugh HSML API To Kafka . 66

LIST OF TABLES

2.1 Table of challenges of implementation of Digital Twins, highlighting its relevance in the Space sector and key new technologies that could help overcome these difficulties. 21

2.2 Comparison Table of Digital Twin Platforms....24

3.1 Structure of the *did_keys* Table in the MySQL Registry....38

3.2 Hardware Configuration Used in Simulation Experiments....45

4.1 HSML Schema Statistics Overview....55

4.2 Usage of HSML Fields in the Cross-Platform Simulation....61

4.3 Usage of HSML Fields in the Verified Cross-Platform Simulation 66

ABBREVIATIONS

3D Three-Dimensional.

AI Artificial Intelligence.

API Application Programming Interface.

AR/VR Augmented and Virtual Reality.

BL Business Logic.

CLPS Commercial Lunar Payload Services.

CPS Cyber-Physical Systems.

DID Decentralized Identifier.

DT Digital Twin.

ESA European Space Agency.

FL Federated Learning.

HSML Hyperspatial Modeling Language.

HSQL Hyperspace Query Language.

HSTP Hyperspatial Transaction Protocol.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

IEEE Institute of Electrical and Electronics Engineers.

IoT Internet of Things.

ISRU In-Situ Resource Utilization.

JAXA Japan Aerospace Exploration Agency.

JPL Jet Propulsion Laboratory.

JSON JavaScript Object Notation.

LCIM Levels of Conceptual Interoperability Model.

LLM Large Language Model.

MAAP Multi-Mission Algorithm and Analysis Platform.

ML Machine Learning.

NASA National Aeronautics and Space Administration.

SQL Structured Query Language.

SSI Self-Sovereign Identity.

SWF Spatial Web Foundation.

SWID Spatial Web Identifier.

UDG Universal Domain Graph.

USD Universal Scene Description.

WWW World Wide Web.

1. INTRODUCTION

This chapter introduces the context of space exploration and the growing role of digital twins in supporting simulation, coordination, and mission operations. It presents the increasing complexity of multi-stakeholder lunar missions and highlights key challenges related to interoperability, communication, and data security. The chapter defines the research problem, outlines the purpose and significance of the study, and introduces the research questions that guide the work. It also discusses the study's limitations and ethical considerations, and concludes with an overview of the thesis structure.

1.1. Introduction

Space exploration has always been a driver of technological advancement, scientific discovery, and international cooperation. As humanity moves from short-term lunar missions to long-term exploration goals, such as the Artemis program and the Moon to Mars pathway (NASA, 2020), collaboration between space agencies, private companies, and international partners becomes increasingly critical. Among these technologies being developed stand Digital Twins (DTs), which are virtual representations of physical assets capable of collecting information from the real environment to represent, validate, and simulate the physical twin's present and future behavior (Botín-Sanabria et al., 2022). DTs have become increasingly important in the aerospace sector for simulating, testing, and monitoring spacecraft and rovers.

Unlike previous decades, where space exploration was dominated by a few major players, today's efforts involve a growing number of stakeholders, each bringing unique capabilities and interests (NASA, 2018). This expansion has led to new challenges in interoperability, communication, and governance, particularly in high-priority regions such as the lunar South Pole. At the same time, despite their growing relevance, DTs are often developed in isolation using domain-specific architectures and proprietary tools (Klar et al., 2024). This fragmentation makes it difficult to coordinate assets across organizations, highlighting the urgent need for standardized, secure, and interoperable systems to support future multi-stakeholder lunar missions.

This thesis explores the implementation of the Spatial Web standard (IEEE Computer Society, 2024), specifically the Hyperspatial Modeling Language (HSML) and the Hyperspatial Transaction Protocol (HSTP), as a framework for achieving digital twin interoperability in a simulated lunar mission scenario. Using three widely adopted platforms in the digital twin domain—Unity, Unreal Engine, and Omniverse Isaac Sim—the experiment evaluates how DTs can exchange structured data in real time while maintaining consistency and security. The simulation focuses on a collaborative rover use case, involving coordination and data sharing between agents using a Kafka-based streaming system and a secure authentication mechanism.

Building on this foundation, the next sections outline the specific research problem, the motivation for addressing it, and the broader context that highlights the importance of secure and interoperable digital twin systems in space exploration.

1.2. Statement of the Problem

The growing interest in long-term lunar exploration, particularly in the lunar south pole, has led to an increasing number of missions involving multiple space agencies, private companies, and international collaborators (U.S. Department of State, 2025; Jones, 2024). Unlike past missions managed by single entities, modern space exploration requires interoperability across diverse platforms and organizations (ESA, 2024*a*; Manning, 2023). However, existing communication and coordination frameworks are not designed for such a distributed environment, creating challenges in data exchange, real-time decision-making, and mission synchronization.

At the same time, digital twins are emerging as critical tools for space exploration, enabling real-time simulation, monitoring, and automation (Liu et al., 2024). However, their implementations remain fragmented, with varying architectures, data models, and protocols that limit interoperability (Botín-Sanabria et al., 2022; Klar et al., 2024). Without a unified framework, integrating DTs across different platforms is challenging, making it difficult to coordinate assets, share mission-critical data, and support collaboration between agencies and private sector entities.

Additionally, DTs require secure and trusted data environments, yet there is no universally adopted mechanism for standardization, authentication, and governance (Botín-Sanabria et al., 2022; IEEE Computer Society, 2024). The lack of clear interoperability standards creates security vulnerabilities, limits data privacy, and complicates access control in multi-stakeholder environments. While new interoperability standards have been proposed (IEEE Computer Society, 2024), their implementation remains an open challenge, particularly in complex domains like space exploration. Determining how to integrate such standards effectively into existing and emerging DT frameworks is an unresolved issue that must be addressed.

In summary, achieving digital twin interoperability for space exploration is hindered by three key challenges: (1) the growing need for multi-actor collaboration and cross-organization communication in space missions, (2) the lack of a unified framework for digital twin interoperability, and (3) the challenge of implementing emerging standards to establish security, governance, and interoperability in digital twin ecosystems. Addressing these challenges is critical for ensuring real-time coordination, efficient data exchange, and trusted digital twin operations in future space exploration missions.

1.3. Background and Need

Space exploration has entered a new era, with the Moon becoming a focal point for both governmental and private missions. Initiatives like the Artemis program aim to establish a sustainable human presence on the lunar surface, driving advancements in various technologies essential for long-term space exploration. Among these technologies, Digital Twins (DTs) play a crucial role by providing accurate, real-time virtual replicas of physical assets, enabling better decision-making and predictive capabilities. As the integration of Artificial Intelligence (AI) and Machine Learning (ML) continues to evolve, DTs are positioned to enhance mission planning, resource allocation, and risk management in space missions. However, to fully realize the potential of DTs, effective interoperability across different systems and platforms is required, especially as the complexity of space missions increases.

1.3.1. The Moon as the next frontier

The Moon has become the new hotspot for space exploration, with over 400 missions scheduled to target either the lunar orbit or the lunar surface over the next decade (ESA, 2024*b*). The future lunar market is projected to reach a value of 137 Billion USD (NSR, 2023). The Moon, being the closest celestial body to Earth, serves as the first stepping-stone in the roadmap of space exploration. Although its proximity is not the only factor driving interest.

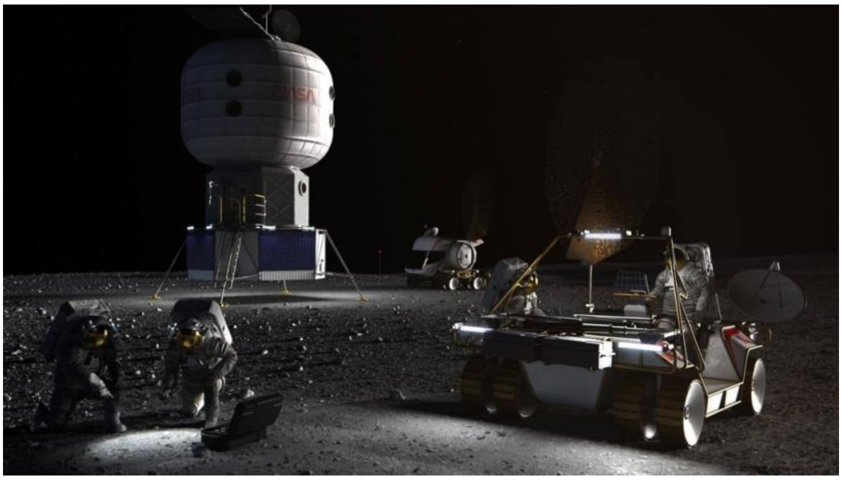
The Moon offers a wealth of valuable resources, making it a highly attractive destination. It holds abundant building materials, water, metals, fuel, oxygen, and solar power. Extensive research is underway to utilize the Moon's predominant surface material, regolith, for In-Situ Resource Utilization (ISRU), enabling the production of essential supplies like oxygen, water, and construction materials directly on the lunar surface (NASA, 2023). This capability is seen as a key step toward supporting long-term lunar habitation and exploration.

NASA's Artemis program aims to return humans to the Moon, including landing the first woman and the next man by 2027, and to establish a sustainable human presence through crewed missions to the lunar South Pole region, as illustrated in Figure 1.1. To achieve this, Artemis is driving advancements in spacecraft, habitats, surface mobility, and communication technologies, laying the groundwork for long-term exploration and habitation (NASA, 2020).

However, reaching the surface of the Moon remains a challenge. To date, the list of countries to have landed successfully on the Moon is reduced to only five: USA, USSR, China, India, and Japan.

Figure 1.1

NASA's Artemis Mission



Note. Illustration of NASA astronauts working on the lunar South Pole. From NASA (2020).

The private space sector has started to play an important role in space exploration. This private landscape is not limited to big players like SpaceX working on Starship to enable humans on the Moon again faster than ever before. Intuitive Machines, an American startup became the first private-sector company to land a spacecraft on the Moon (Kawahara and Hanafusa, 2024).

Other startups are also planning on reaching the lunar surface, such as the Japanbased iSpace and the American Firefly Aerospace, whose landers were both launched on the same Falcon 9 rocket at the beginning of 2025. Resilience is the second attempt for the Japanese startup to land on the Moon, as a culmination of its HAKUTO-R program, in which the lander will also carry a small rover for surface exploration and data collection. Firefly Aerospace's lander is called Blue Ghost under the mission named Ghost Riders in the Sky, which is part of NASA's Commercial Lunar Payload Services (CLPS) program to deliver scientific payloads to the surface of the Moon (Dinner, 2025). These developments reinforce the idea that the Moon could become the flight layover for future solar system travel.

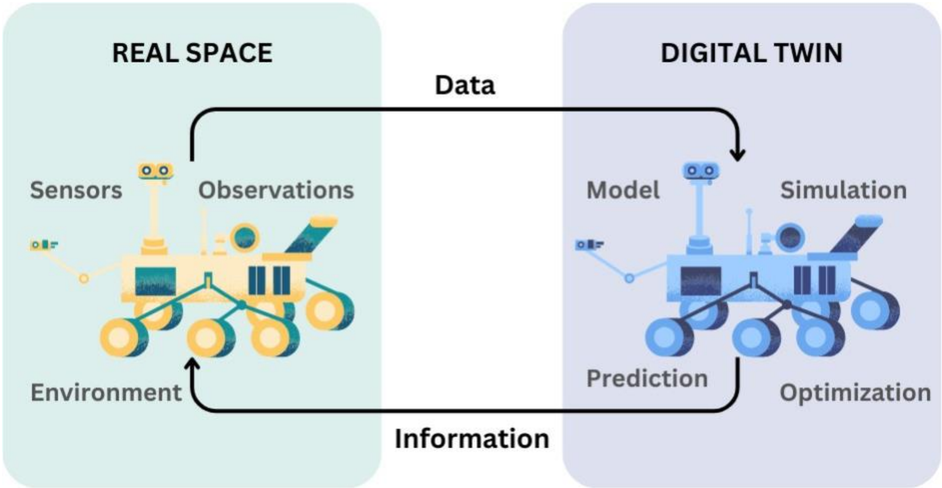
1.3.2. The rise of Digital Twins

The origin of digital twins dates back to the 1960s and the era of the Apollo missions, when NASA used simulators to evaluate the failure of Apollo 13's oxygen tanks. Although the term Digital Twin would not be formally introduced until 2010 in the NASA roadmap report by John Vickers (Guo and Lv, 2022). Since 2014, digital twins have been part of the marketing campaigns of well-known companies like Siemens, ANSYS or Dessault (Guo and Lv, 2022). However, despite their popularity, nowadays there is still a disparity on what exactly constitutes a digital twin.

According to the Spatial Web, a Digital Twin is a 3D virtual replica or representation of the data associated with a physical asset, process, or system. It serves as a highly detailed model that acts as the digital counterpart of a real-world object (René and Mapes, 2019, p. 158). By utilizing computer vision and connected sensors on the physical asset, data is continuously gathered and synchronized with the virtual model. This data exchange, illustrated in Figure 1.2 ,enables the Digital Twin to provide valuable insights into the physical asset's performance in the real world, offering both real-time and historical information on its state and operations (René and Mapes, 2019, p. 158).

Figure 1.2

Conceptual Flow Between Real Space and Digital Twins



Note. Illustration of the bidirectional relationship between real-world physical systems and their digital twin counterparts. Information flows from sensors and observations in the physical environment to simulations and models in the digital domain, while feedback and control can be sent back to optimize real-world operations. Inspired by Guo and Lv (2022).

In the past, with only physical assets available, real-life testing was the only option for space manufacturers. However, with the introduction of AI and other emerging technologies, it has become impossible to account for all possible scenarios just by live testing. This is why Digital Twins have become key for testing today, as they bridge the gap between the physical and digital worlds. Additionally, when running simulations, we cannot consider all individual external elements, especially when other players are also involved. To make simulations more accurate, we must connect with others and consider their assets.

However, the Aerospace sector is not the only benefited by this technology, digital twins are driving change in many other industries. DTs are essential for Industry 4.0, bridging the physical and digital worlds to enable smarter, more efficient industrial systems. As industrial Internet of Things (IoT) and 5G advance, DT technology is expected to scale globally, driving innovation in high-value sectors like manufacturing and energy (Guo and Lv, 2022). For instance, in the automotive industry, DTs are applied throughout the entire lifecycle of the vehicle, allowing designers and engineers to test and verify vehicle performance in a virtual environment, such as crash tests and climate adaptability, thus reducing the cost and time of physical testing (Liu et al., 2024).

DT technology is also progressively being used in the healthcare sector for personalized medical treatment and surgical planning (Liu et al., 2024). Additionally, DTs help in the development of smart cities, as they integrate multiple disciplines, multiple scales, and multiple probabilities, which can well meet the virtual mapping of smart cities to the real world (Guo and Lv, 2022). In the business world, digital twins can drive sales by creating digital models of products or even entire stores online, transforming the customer shopping experience (Guo and Lv, 2022).

Nevertheless, the use of DT technology not only transforms the operational models of traditional industries but also significantly influences the economic structure of society. By optimizing production processes and improving resource efficiency, digital twins contribute to achieving sustainable development goals (Liu et al., 2024).

As DTs continue to evolve, their integration with current technologies like ML is shaping their future potential, opening up new possibilities for innovation and efficiency. Future developments will focus on cross-disciplinary innovation, such as incorporating Augmented and Virtual Reality (AR/VR) to enhance data interaction and visualization (Liu et al., 2024). Additionally, new technologies like Cyber-Physical Systems (CPS), edge computing, blockchain, AI/ML, big data analytics, and AR/VR are facilitating diverse integration across systems, including digital twins, Federated Learning (FL), and enhanced cybersecurity (Acharya et al., 2024). These technologies are driving the evolution of digital twins and creating opportunities for smarter, more responsive systems in various industries.

In space exploration, AI and ML are significantly enhancing DT development, optimizing operations and automating processes. By processing vast amounts of environmental data, such as monitoring Earth's magnetic field, solar activity, and cosmic rays, DTs support mission planning, asset protection, and fault prediction (Liu et al., 2024). DTs can also integrate historical data with real-time sensor data, providing a 3D model that evolves with its physical counterpart, allowing for simulations, predictive analysis, and dynamic process monitoring (René and Mapes, 2019, p. 158). In the context of space, digital twins have the capacity to improve orbit prediction, resource allocation, and risk assessment, contributing to mission efficiency, safety, and reliability, while enabling new scientific and commercial applications in space exploration (Liu et al., 2024).

1.3.3. The path to interoperability with the Spatial Web

The Spatial Web Foundation (SWF) and the IEEE are setting the global standards for AI, IoT, and robotic interoperability and governance. The Spatial Web standard was approved by the IEEE Working Group in the summer of 2024, and is now awaiting publication. Given its role in establishing interoperability, this project selected the Spatial Web standard from among the various protocols being developed for Digital Twins.

The Spatial Web standard aims to set the guidelines that will define the Web 3.0, a computing environment that will blur the lines between the physical and digital worlds. The Spatial Web is an evolution of the internet, where we move from a 2D environment to a 3D virtual space that will allow people to interact with places, things, and each other (René and Mapes, 2019). The Spatial Web standard enables agent interactivity and interoperability across physical, virtual, and hybrid domains through HSML, HSTP, HSQL, and a Universal Domain Graph (UDG). It integrates with IoT, Web of Things, Sensor Web, and Industry 4.0 to represent people, places, and objects as Internet entities, supporting interactions via sensors, actuators, and autonomous systems (IEEE Computer Society, 2024).

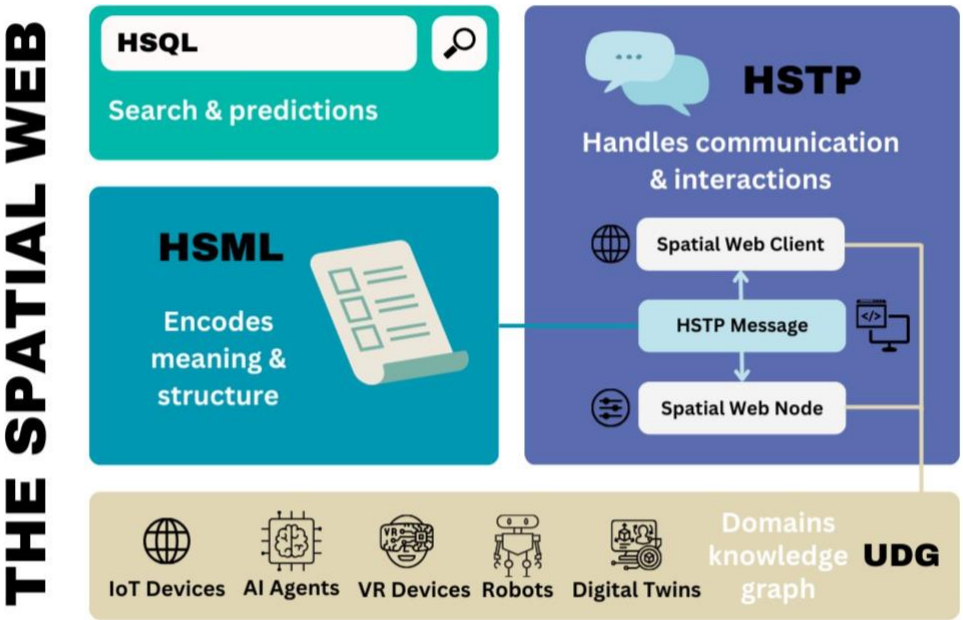
To be able to place information spatially and contextually, we require a detailed map that explains how all the different data is related to each other in the system. This is where the Hyperspace Modeling Language (HSML) comes into play. HSML is a multi-dimensional ontology for encoding fundamental elements and the relationships between them (IEEE Computer Society, 2024). HSML does for the Spatial Web as HTML does for the WWW.

Additionally, a system needs to be put in place to communicate and exchange information. This is the role of the Hyperspace Transaction Protocol (HSTP), which is a multi-dimensional range query for governing the interactions between actors and assets. This system is designed to enable smooth, decentralized, secure, and private communication between the different Spatial Web nodes, focusing on transactions in the form of HSML messages (IEEE Computer Society, 2024). Thus, HSTP does for the Spatial Web as HTTP does for the WWW.

Finally, although this work addresses it to a lesser extent, the Spatial Web also defines Hyperspace Query Language (HSQL), which is a syntax for performing predictive queries on a vector graph document database which combines comparative, relational, and similarity searching (IEEE Computer Society, 2024). All these protocols together shape the Spatial Web. Their correct implementation is key to ensuring the efficiency and functionality of this standard, particularly in enabling interoperability between digital twins for future space exploration missions.

Figure 1.3

The Spatial Web Protocols



Note. This figure illustrates how the Spatial Web protocols work together to enable semantic understanding (HSML), communication (HSTP), and intelligent querying (HSQL). Together, they facilitate interaction across devices and agents (IoT, AI, VR, robots, digital twins), all grounded on a shared knowledge base: the Universal Domain Graph (UDG). Inspired by concepts from the IEEE P2874 Spatial Web standard (IEEE Computer Society, 2024).

1.4. Purpose of the Study

The purpose of this study is to evaluate the effectiveness of the Hyperspatial Modeling Language (HSML) and the Hyperspatial Transaction Protocol (HSTP), developed as part of the Spatial Web standard, in achieving digital twin interoperability while ensuring secure data exchange among stakeholders. As lunar exploration expands with increased international and private sector involvement, seamless collaboration between DTs is essential. However, the lack of standardized interoperability frameworks presents challenges in data sharing, mission coordination, and security.

This study aims to implement and extend HSML as a common ontology for digital twins and test its integration across three widely used simulation platforms: Omniverse, Unity, and Unreal Engine. Additionally, it will focus on leveraging HSTP to establish a secure communication framework for real-time data exchange between digital twins. By simulating a lunar surface collaboration scenario where rovers exchange data in real time, the study will assess the feasibility of HSML for interoperability and determine the architectural requirements for connecting various platforms using HSTP to facilitate trusted, real-time data transactions.

The expected outcomes include demonstrating real-time synchronous simulation with latency under 50 milliseconds, identifying necessary modifications for platform compatibility, and validating HSML's adaptability to exchanged data types. Furthermore, the study will evaluate the implementation of HSTP in securing data transactions through Decentralized Identifiers (DIDs) and cryptographic methods. The findings will contribute to advancing digital twin development and interoperability in space exploration while assessing the applicability of the Spatial Web standard in this domain.

1.5. Research Questions

To investigate the standardization of digital twins for space exploration and the role of the Spatial Web in enabling interoperability and secure communication, this study proposes a core research question (RQ) and refines it into several subquestions (SQx) for further exploration:

RQ: How can standardization be applied to achieve digital twin interoperability in space exploration?

- SQ1: Can different simulation platforms (Omniverse, Unity, and Unreal Engine) effectively exchange data using the HSML ontology?

- SQ2: How does the implementation of HSML and HSTP impact interoperability, data integrity, and communication latency across digital twins?
- SQ3: Does the authentication framework based on HSTP and DIDs ensure secure and trusted data sharing in a multi-stakeholder space environment?
- SQ4: What are the challenges of integrating standardized digital twins into real-time space operations, and how can they be addressed?

These research questions aim to assess the feasibility and impact of adopting HSML and HSTP as standardization mechanisms for digital twin interoperability, while also identifying technical and operational challenges in their implementation.

1.6. Significance to the Field

The main contributions of this study to the scientific and engineering communities are summarized below:

- **Demonstration of a standardized interoperability framework for digital twins in space exploration:** This thesis presents one of the first implementations of the Spatial Web standard—through HSML and HSTP—as a foundation for achieving digital twin interoperability. By conducting simulations across three major platforms (Omniverse, Unity, and Unreal Engine), the study provides a concrete demonstration of how heterogeneous digital environments can communicate using a shared ontology and secure protocol architecture.
- **Development and evaluation of a secure, real-time communication protocol for distributed digital twins:** The study introduces and tests a lightweight authentication and data exchange system using HSTP and decentralized identifiers (DIDs). This contributes to the emerging body of research on trusted multi-agent systems by showing how secure, scalable, and real-time communication can be achieved without compromising latency or data integrity—critical for collaborative missions in space environments.
- **Advancement of spatial computing standards through applied testing of HSML:** By extending and implementing the Hyperspatial Modeling Language (HSML) to

support semantic descriptions of agents, activities, and assets across platforms, this work contributes to validating the Spatial Web standard's applicability in practical settings. More importantly, by applying HSML within the domain of space exploration and multi-agent simulation, this study provides a concrete example that can serve as a foundation for others seeking to implement or build upon the HSML schema. It offers a domain-specific reference that helps advance the development, adaptation, and scalability of HSML for future applications across other sectors.

1.7. Limitations

While this study aims to explore the applicability of the Spatial Web standard for achieving digital twin interoperability in space exploration, several limitations must be acknowledged. First, the experimental framework is limited to three well-established digital twin platforms—Omniverse, Unity, and Unreal Engine—which already incorporate certain features that facilitate interoperability. As a result, the findings may not fully extend to proprietary or internal digital twin platforms used by private companies, where integration could pose greater challenges.

Second, the scope of the experiment is restricted to a limited set of data types exchanged between platforms, which may not capture the full complexity of data encountered in real-world space operations. Additionally, the implementation of the Spatial Web standard in this study is partial; certain components, such as the identity verification mechanism, still rely on centralized databases for key management and validation. This limits the ability to fully assess the decentralized potential envisioned by the standard.

These constraints may affect both the internal validity—by simplifying some system interactions—and the external validity, as generalizability to more complex or highly proprietary environments remains limited.

1.8. Ethical Considerations

This study did not involve human subjects, personal data, or any form of animal testing. All work was conducted using virtual simulations and synthetic data within digital environments. Ethical considerations were primarily focused on responsible data handling, transparency, and reproducibility.

The implementation of decentralized identifiers (DIDs) and secure communication protocols was carried out with attention to privacy and data protection principles, aligning with the ethical goals of building secure and trustworthy systems. All software tools and libraries used in the project were open-source or properly licensed, and all contributions were appropriately credited.

Artificial intelligence (AI) tools were used in limited capacities to support automation and code generation tasks, such as assisting with file conversion and script development. These tools were used to enhance productivity but did not influence the research design, data interpretation, or schema development. Their use was carefully monitored to maintain accuracy and academic integrity.

Finally, as this research involves emerging technologies for space exploration, care was taken to present findings accurately and transparently, without overstating the capabilities or readiness of the proposed frameworks.

1.9. Structure of the Thesis

This thesis is organized into five chapters. **Chapter 1** introduces the research topic, presents the problem statement, outlines the research questions, and defines the objectives and significance of the study. **Chapter 2** provides a review of the existing literature, covering the current state of digital twin technologies, challenges in interoperability, collaboration in space exploration, and the emerging Spatial Web standard. **Chapter 3** describes the methodology, including the experimental setup, the development and extension of HSML and HSTP, the simulation platforms used, and the overall evaluation framework. **Chapter 4** presents and analyzes the results of the simulation experiments, focusing on interoperability performance, communication latency, security, and platform-specific integration challenges. Finally, **Chapter 5** summarizes the key contributions of the study, discusses its limitations, and outlines directions for future work.

2. REVIEW OF RELATED WORK

The chapter discusses the challenges of interoperability in Digital Twin (DT) systems, particularly when integrating platforms like NVIDIA Omniverse, Unity, and Unreal Engine. It highlights the issues of proprietary formats, differing APIs, and incompatible simulation tools, making seamless communication difficult. The chapter emphasizes the role of the Spatial Web, particularly the Hyperspatial Modeling Language (HSML) and Hyperspatial Transport Protocol (HSTP), in providing a standardized framework for interoperability, security, and governance. It also discusses how these advancements can benefit space exploration, improving communication and collaboration between different space agencies and missions through interconnected digital twins.

2.1. Space Exploration: Communication and Collaboration

Space exploration increasingly relies on international and private-sector collaboration, especially with missions like Artemis paving the way for a sustained Moon to Mars presence. With more actors involved, ensuring interoperability, secure communication, and coordinated operations is essential. This section examines the need for collaboration in lunar exploration, the role of digital twins, and emerging communication frameworks like LunaNet and Moonlight.

2.1.1. Why is Interoperability needed and important now in the context of Moon Exploration

On Mars, NASA remains the primary operator, meaning that when the JPL mission operations team plans the next move of the Perseverance rover, they only need to consider environmental risks rather than interference from other space agencies or private entities (NASA JPL, 2025). In contrast, lunar exploration is becoming increasingly competitive. The current space race for the Moon differs significantly from that of the 1960s, when the USA and the USSR were the sole competitors. Today, the rise of the private space sector and the growing number of participating countries have led to a considerable increase in the number of players.

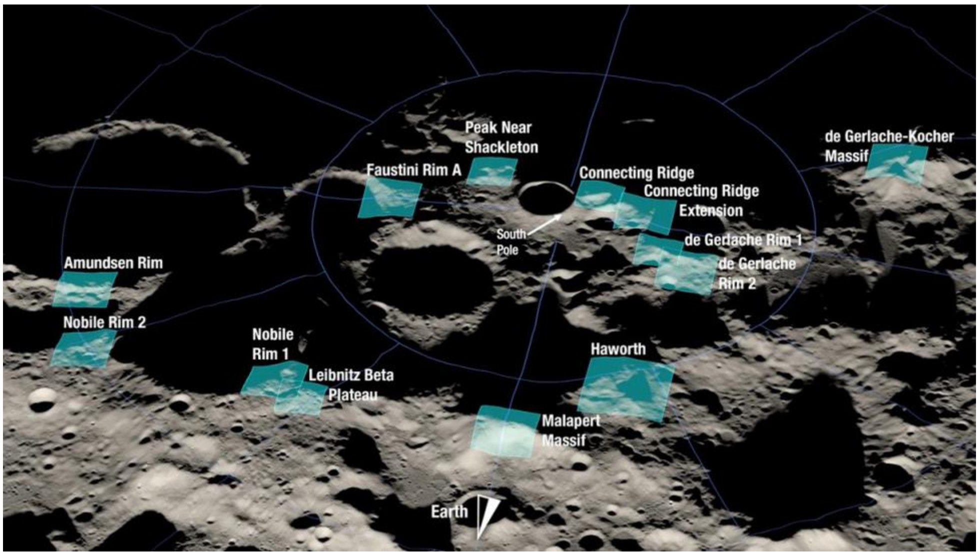
NASA's Artemis program is proof of the future vision for lunar exploration and highlights the increasing need for collaboration on the Moon. Unlike in the past, today's efforts are built on partnerships, as seen with the Artemis Accords, launched in 2020, bringing countries together in a shared agreement for peaceful, sustainable, and transparent cooperation in space. The Artemis Accords, with 53 signatory countries as of January 2025, are a non-binding set of principles designed to guide civil space exploration and use in the 21st century, rooted in the Outer Space Treaty of 1967 (U.S. Department of State, 2025). Among these principles is interoperability, showcasing its importance for space exploration, as it enhances the potential for safe and robust missions.

However, the real challenge lies in applying these principles in practice. How can these nations cooperate on the lunar surface? And this is where digital twins could play a significant role. The global and multi-stakeholder nature of these programs needs a renewed focus on complex infrastructure across various technical fields, planetary environmental stewardship, and legal and collaborative frameworks (Ehrenfreund and Christensen, 2024).

Another layer of difficulty lies in the location of these missions, as the general interest is in the lunar South Pole, where ice has been found and other attractive resources are suspected to exist, based on observations from orbit (Barnett, 2019). NASA has identified 13 candidate landing sites in this region for the Artemis III mission, which will be the first crewed mission to the Moon's surface since 1972. These are scientifically significant due to their proximity to permanently shadowed areas rich in resources, like water ice, which could support long-term lunar exploration (Lloyd et al., 2022). These regions are also considered attractive because of their potential for long-duration sunlight access, direct communication with Earth, and more favorable terrain for landing (NASA, 2020).

Figure 2.1

NASA's 13 candidate landing regions for Artemis III mission.



Note. The figure shows a rendering of the 13 candidate landing regions for NASA's Artemis III mission. Each region is approximately 9.3 by 9.3 miles (15 by 15 kilometers). A landing site is a location within those regions with an approximate 328-foot (100-meter) radius. From NASA (2022).

Another mission targeting this area is the Chandrayaan-5 mission, also known as LUPEX; a collaboration between ISRO and The Japan Aerospace Exploration Agency (JAXA). It is planned to land around 2028 on an elevated ridge near Shackleton crater, carrying a rover, whose objective is to take in-situ measurements, including determining potential water-ice deposits (Jones, 2024). Thus, with the growing interest in this region, the available space for free exploration and settlement may become limited, making collaboration and interoperability essential to ensure the success and safety of missions operating in close proximity.

2.1.2. Collaboration on the Moon towards achieving Interoperability & Communication

In this new era of lunar exploration, nations and private entities are not just competitors but also collaborators, working together to ensure the success and sustainability of missions on the lunar surface. Efforts like lunar traffic management frameworks and resource utilization agreements are being explored to mitigate conflicts as more actors target the same high-value regions (Bilka et al., 2024). NASA and ESA are jointly developing lunar communications and navigation standards, including LunaNet and Moonlight, to support interoperability between international missions (ESA, 2024*a*; Manning, 2023). The ESA-led Lunar Pathfinder mission aims to establish reliable lunar telecommunications, facilitating coordinated operations (ESA, 2024*a*). Additionally, the Moon Village concept, promoted by ESA, envisions a long-term, multinational lunar presence, emphasizing cooperation in infrastructure, science, and resource management (Wörner, 2016).

Establishing effective communication and data-sharing frameworks is crucial for ensuring smooth collaboration in space exploration. NASA's Multi-Mission Algorithm and Analysis Platform (MAAP) has been developed to facilitate collaborative Earth science data analysis by providing a cloud-based environment for international partners to share algorithms, datasets, and computational resources, ensuring seamless cooperation (NASA, 2025). However, sharing data on the Moon becomes a challenge, particularly due to the competitive nature of some missions, especially among private companies. The concept of "trustful coopetition" has been proposed as a model for balancing competition and collaboration among international space actors, ensuring that cooperation does not compromise strategic independence (Lima Baima et al., 2024). Recent work from our lab, in collaboration with VERSES AI, explores how Spatial Web technologies and AI-driven digital twins can enable secure, interoperable frameworks for future lunar operations (Carrillo et al., 2025). These efforts emphasize the growing need for continued research and practical demonstrations to validate strategies for communication, coordination, and data trust in collaborative space missions.

2.2. Digital Twins in Space: Opportunities and Challenges

This section explores the role of digital twins in the space sector, where they were originally developed, and their growing importance in the industry. It also examines the platforms currently used for creating DTs and the challenges these platforms present. Special attention is given to the interoperability issues that arise when integrating different DTs, which is a key focus of our project. By understanding these challenges, we can better address the future of DTs and their relevance in the years to come.

2.2.1. Digital Twins in the Space Industry

The origin of digital twins is in the space industry, and space is still nowadays one of the most focused applications of digital twins. Mostly due to their harsh conditions, small room for error and high testing costs, digital twins present themselves as the optimal solution to tackle all kinds of space endeavors, from rocket manufacturing to exploration rovers or the coordination of large constellations.

At NASA Jet Propulsion Laboratory (JPL), for the Mars 2020 mission, the concept of twinning was taken to a new level with the creation of a complete physical replica of the Perseverance Rover, called OPTIMISM, which allowed for testing in the Mars Yard at JPL's facilities (Brennan, 2021). However, creating physical replicas is not a viable solution for every space asset due to cost and logistical constraints, making digital twins an essential tool. In fact, space manufacturers are increasingly adopting digital twins for mission planning and testing.

SpaceX is proof of the acceleration that using DTs can bring to the entire design, manufacturing, and testing process. For example, the company has integrated DT technology into various aspects of its operations, notably in designing and testing its Starlink satellite constellation. These digital twins enable detailed simulations, such as assessing the impact of solar flares on network performance (Freedman, 2024). Additionally, SpaceX employs DTs to monitor and optimize the performance of its rockets and spacecraft through real-time data analytics. As it is the case with the digital replica of SpaceX's Dragon vehicle, which allows operators in Mission Control to monitor its status—trajectory, loads, propulsion systems, etc.—from data received from the hundreds of sensors integrated into the spacecraft (Carlos, 2021). Starship also leverages DT technology to simulate its entire lifecycle, from design to performance under extreme space conditions, allowing engineers to predict and address potential issues without physical prototypes (Liu et al., 2024). This approach allows for continuous improvement and efficient management of their aerospace systems.

Other major satellite constellations, such as OneWeb and Amazon's Project Kuiper, are also adopting digital twin technologies to enhance production, coordination, and operational efficiency. As DTs become more advanced, we can expect their integration to improve satellite constellation management, enhance analysis and simulation capabilities, and help predict and prevent network outages (Freedman, 2024).

2.2.2. Challenges in Digital Twin Implementation for Space Exploration

Despite the potential of digital twin technology across industries, its implementation and development still face significant challenges. While some challenges are universal, the space sector introduces unique constraints, such as extreme environmental conditions, high-cost barriers, and strict data security requirements. Table 2.1 presents key challenges and their specific implications for space missions, along with emerging technologies that could address these issues.

Table 2.1

Table of challenges of implementation of Digital Twins, highlighting its relevance in the Space sector and key new technologies that could help overcome these difflculties.

Challenge	Description	Impact on Space Sector	Key Technologies & Solutions
Data Security & Privacy	Trust, privacy, cybersecurity, governance, and large-scale data analysis pose significant hurdles (Botín-Sanabria et al., 2022).	Space missions generate large amounts of data, including sensitive astronaut health data and critical system operation parameters, making cybersecurity a top priority (Liu et al., 2024).	Zero-trust architectures, and blockchain for secure data exchange (Guo and Lv, 2022).
Lack of Standards & Interoperability	The absence of unified frameworks and interoperability regulations limits DT adoption, particularly in manufacturing (Botín-Sanabria et al., 2022). Ensuring interoperability and compatibility between different systems is a complex engineering task (Liu et al., 2024).	DTs require seamless integration with existing space assets, communications networks, and ground control centers (Liu et al., 2024). Inconsistent data formats and isolated systems obstruct collaboration between space agencies, private companies, and international partners. This limits DT model reusability across missions.	IEEE P2874 (interoperability for DTs), Universal Scene Description (USD) for data sharing (Carrillo et al., 2025), ontology-based frameworks (Böning, 2021), and developing intuitive and efficient user interfaces (Liu et al., 2024)
High Implementation Costs	The need for extensive sensors and computational resources makes DT adoption expensive, restricting accessibility (Botín-Sanabria et al., 2022).	The industry´s complexity makes implementation difficult and costly. Key technical hurdles include fault diagnosis with big data, modeling under uncertainty, real-time analysis, and resilience to extreme conditions (Guo and Lv, 2022).	Edge computing for onboard processing, AI-driven optimization of simulation resources (Acharya et al., 2024), cloud-based DTs for shared infrastructure.
AI & Big Data Processing	Managing and analyzing vast data streams require advanced AI and IoT technologies, with challenges in standardization and policy development (Botín-Sanabria et al., 2022).	Large-scale space missions generate vast amounts of telemetry data, which need real-time analysis despite communication delays. Efficient processing remains a challenge (Liu et al., 2024). Additionally, the framework for DTs remains immature, particularly in reasoning and intelligence (Guo and Lv, 2022)	Developing integrated platforms for autonomous DT development and operation (Guo and Lv, 2022). Quantum computing for faster simulations, neuromorphic chips for low-power AI, on-orbit processing (Acharya et al., 2024).
Communication Network Constraints	Faster, more efficient networks (e.g., 5G) are needed to support real-time data connectivity and ensure operational efficiency (Botín-Sanabria et al., 2022).	Deep-space networks suffer from latency and bandwidth limitations (Elewaily et al., 2024), affecting real-time synchronization and control of digital twins in space.	Optical communication, 6G for interplanetary networks (Kuruvatti et al., 2022), delay-tolerant networking (DTN) (Elewaily et al., 2024).

Note. This table summarizes key challenges in implementing DTs within the space sector, including data security, interoperability, high implementation costs, AI and big data processing, and communication network constraints. It also highlights relevant technologies such as blockchain, edge computing, AI-driven optimization, and advanced communication networks that can help address these challenges.

Among these, data security and privacy are key concerns, as digital twins involve the real-time transmission and processing of a large amount of sensitive information. Integrating diverse technologies also poses difficulties, particularly in balancing customization with standardization across different industries and applications (Liu et al., 2024). Additionally, interoperability issues frequently cause inefficiencies, increased costs, and project delays (Acharya et al., 2024). The lack of interoperability between these systems hinders DT development and reusability, making this challenge the focus of our research.

2.2.3. Evaluating Digital Twins Interoperability

Interoperability refers to the ability to discover, connect, and interact with other entities within a broader application context. This is a fundamental principle in modern distributed computer systems, where ensuring interoperability across all levels of the technology stack is essential to maximize the benefits of interconnected systems (Budiardjo and Migliori, 2021). However, one of the key issues in current DTs is that most remain twinned in isolation and very domain-specific (Klar et al., 2024), preventing them from reaching their full potential.

Klar et al. (2024) define 6 DTs maturity levels, being the last one Interoperability, as they consider Interoperable DTs to be the most mature twinning platform. But trust poses a challenge, particularly in securely sharing data across multiple digital twin systems while ensuring privacy, ownership, and protection between different institutions (Klar et al., 2024).

Additionally, interoperability is not a single achievable state but consists of multiple layers. The Levels of Conceptual Interoperability Model (LCIM) framework defines seven interoperability levels, ranging from L0 (no interoperability) to L6 (conceptual interoperability), where systems are fully aware of each other's information, context, and modeling assumptions, resulting in coherent, collective reasoning faculties (David et al., 2025). The key insight from the panel discussion at the 2023 Annual Simulation Conference is that, unlike the current state of digital twinning, which often restricts interoperability to lower levels, there is a pressing need to achieve higher levels of interoperability, particularly in distributed digital twin scenarios. Additionally, the panel highlighted emerging solutions to address this challenge, focusing on three approaches: co-simulation, standardization, and infonomics (David et al., 2025).

Interoperability standards, where the focus is on ensuring communication protocol compatibility, file format interoperability, and managing both technological obsolescence and backward/forward compatibility, are fundamental for DTs (Acharya et al., 2024). As Guo and Lv (2022) also indicates, establishing these standards is crucial for integrating DTs into real-world production and accelerating economic and social digitization. However, one of the biggest barriers for the widespread adoption of Digital Twin (DT) technology is the development of international standards applicable across industries (Guo and Lv, 2022). This challenge is being tackled by the Spatial Web, which is currently working on the standards that will define the future Web 3.0, enabling seamless connectivity across all systems and industries.

2.2.4. Interoperability Among Digital Twin Development Platforms

The widespread adoption of digital twins has resulted in a large number of frameworks and platforms from different vendors. However, the absence of technological convergence and common Application Programming Interfaces (APIs) complicates their integration and compositionality (David et al., 2025). Proprietary solutions like Siemens Xcelerator, Ansys Twin Builder, and Autodesk Tandem offer specialized tools tailored to specific domains, often relying on vendor-specific formats and ecosystems. In contrast, there are other platforms like NVIDIA Omniverse, designed to facilitate interoperability by supporting industry-standard formats like Universal Scene Description (USD) and integrating with third-party tools (NVIDIA, 2025). Despite these capabilities, interoperability remains a key challenge, as varying data formats, modeling standards, and communication protocols hinder seamless integration. Addressing these issues is essential for enabling cross-platform collaboration and maximizing the potential of digital twin technology.

For the purpose of this project, three of these platforms were selected as a proof of concept for achieving interoperability among digital twins. These platforms are NVIDIA Omniverse with Isaac Sim, Unity, and Unreal Engine, which are widely used across different industries. NVIDIA Omniverse is a simulation and collaboration, supporting industrial digital twins, AI, and robotics, with Isaac Sim providing high-fidelity physics and robotics simulation (NVIDIA, 2021). Unity is a real-time 3D development platform originally used for video games but now widely adopted for simulations, AR/VR, and industrial applications (Unity, 2021). Unreal Engine, developed by Epic Games, is known for its high-fidelity rendering, real-time physics, and use in gaming, film, and architectural visualization (Epic Games, 2021).

Table 2.2 compares these platforms based on key factors affecting digital twin interoperability such as physics simulation, data format compatibility, scripting support, and plugin ecosystems. These are critical considerations when analyzing interoperability among digital twins developed in different environments, helping us to understand the constraints and challenges of integrating them.

Table 2.2

Comparison Table of Digital Twin Platforms

Category	NVIDIA Omniverse	Unity	Unreal Engine
Physics Simulation	PhysX, FleX, Omniverse Kit Physics	PhysX, Unity Physics, Havok Physics (DOTS)	PhysX, Chaos Physics, Niagara for VFX
Supported Data Formats	USD (native), FBX, OBJ, glTF, STEP, IGES	FBX, OBJ, glTF, limited CAD support	FBX, OBJ, glTF, USD, limited CAD support
Scripting & API Support	Python-first, C++, Omniverse Kit API	C#, DOTS (ECS), limited Python support	C++, Blueprints, Python API available
Plugin Ecosystem & Integrations	Deep ROS integration, AI & ML tools, cloud, real-time data	ROS#, Machine Learning Agents (ML-Agents), IoT integrations	ROS 2, Twinmotion, MetaHumans, AI tools
Rendering & Visualization	RTX-based ray tracing, USDView, path tracing	HDRP/URP, real-time rendering, optimized for games	High-fidelity graphics with Lumen, Nanite

Note. The digital twin platforms—NVIDIA Omniverse, Unity, and Unreal Engine— are compared based on physics simulation, data format support, scripting, plugin ecosystems, and rendering. Omniverse excels in AI and real-time data integration, Unity is flexible for games and IoT, and Unreal Engine is noted for high-quality graphics and ROS 2 support (NVIDIA, 2021; Unity, 2021; Epic Games, 2021).

From this comparison, we can observe that there is an apparent disparity in the levels of interoperability among these platforms. Omniverse seems to be the most advanced, integrating with external tools and frameworks through native USD support and industry-standard compatibility. Unity possesses an extensive asset store and APIs for integration, but requires additional tools for advanced simulations. Unreal Engine shows moderate interoperability, often requiring custom plugins and scripts for some advanced integrations with robotics and industrial digital twins. These differences highlight the challenges of achieving seamless interoperability across platforms.

2.3. The Spatial Web: Interoperability, Security, and Implementation

The Spatial Web standard, developed by the Spatial Web Foundation and IEEE, addresses several critical gaps in digital twins by enabling interoperability, security, and privacy. This research explores the standard's application to DTs, focusing on its feasibility and key components. The Spatial Web promotes interoperability through a common ontology, HSML, which facilitates communication across diverse platforms. It ensures trust through governance mechanisms that enforce authenticity, integrity, confidentiality, and privacy. While the standard shows promise, some challenges like integration with existing protocols or scalability for space missions remain. This study aims to evaluate these aspects and provide insight into how the Spatial Web can be implemented for future space-based DT applications.

2.3.1. The Value of the Spatial Web for Digital Twins

Digital twins struggle with standardization and interoperability across diverse platforms, and security concerns regarding data privacy. The Spatial Web bridges these gaps by offering a common framework that enables seamless communication and robust data protection. This section explores how the Spatial Web addresses these concerns, and why applying their standard to DTs is advantageous.

Property and Security through Decentralization

In the Spatial Web, decentralization ensures trust and security by providing verifiable and trusted identities. This is crucial for enabling secure interactions, transactions, and the transportation of data between people, places, and things (René and Mapes, 2019, p. 135). Identity is handled using Self-Sovereign Identity (SSI) ledgers, where decentralized identifiers (DIDs) ensure privacy and security through encryption and multi-factor authentication (René and Mapes, 2019, p. 183). Stored on blockchains with quantumresistant encryption, DIDs can integrate biometric markers and location-specific anchors to strengthen authentication, mitigating risks like Sybil attacks (René and Mapes, 2019, p. 142). This decentralization moves us closer to zero-trust architectures, which are essential for privacy and data security.

Furthermore, the Spatial Web's Distributed Ledger Technologies ensure that assets and transactions are secure (René and Mapes, 2019, p. 75), while spatial contracts (smart contracts) provide a layer of security by specifying the conditions under which actions and transactions can take place (René and Mapes, 2019, p. 183). By using secure channels and multi-layered encryption, the Spatial Web guarantees that data can be safely stored on public or private ledgers, ensuring that sensitive data remains protected (René and Mapes, 2019, p. 183).

Interoperability

Interoperability is at the core of the Spatial Web's value proposition, enabling different platforms and systems to communicate seamlessly. A significant part of this interoperability is the use of common ontologies, like the Hyperspatial Modeling Language (HSML). HSML offers a standardized language that allows diverse DTs to interact, regardless of their platform or origin. This common ontology facilitates the exchange of data across various domains, ensuring that both physical and virtual objects can be integrated and interact smoothly.

As the Spatial Web spans across heterogeneous systems with differing protocols, data types, and applications, it must ensure true interoperability—the ability of two or more systems to exchange information and mutually use the information that has been exchanged (IEEE Computer Society, 2024). By offering a shared modeling language, the Spatial Web promotes a unified approach to spatial data, overcoming the fragmentation typically seen across different DTs and platforms. This ensures that components within a digital twin ecosystem can collaborate efficiently, whether they are within the same domain or distributed across diverse systems.

Governance, Privacy, and Trustworthiness

Governance in the Spatial Web is essential for ensuring safety and trustworthiness. Governance is enabled through credentials, norms, and contracts. Credentials are used for authentication, ensuring that only authorized entities can access certain domains and interact with the system. Norms define the principles of conduct, ensuring that interactions and transactions occur in a way that respects both security and privacy. Contracts, often implemented as smart contracts, establish binding agreements that define the conditions under which actions can be performed, dictating the activities and relationships between different domains (IEEE Computer Society, 2024).

Safety and trustworthiness are key to the Spatial Web's ability to create reliable and secure interactions. To achieve this, the Spatial Web standard focuses on four key aspects of trust in digital interactions: Authenticity, supported in HSML by SWIDS to verify identity; Integrity, ensuring communications remain untampered and complete; Confidentiality, enforced through the transactional layer (HSTP) to protect sensitive exchanges; and Privacy, embedded as a core design principle (IEEE Computer Society, 2024). By implementing these governance principles, the Spatial Web not only facilitates interoperability but also ensures that data privacy and security are maintained throughout all transactions and interactions.

2.3.2. Implementation of the Spatial Web Architecture

To implement the Spatial Web standard and achieve digital twin interoperability, we need to understand its key architectural requirements. These are Domain & Address, which defines the address and ownership of a space; Program, which establishes governance, rules, and permissions; Protocol, which ensures seamless communication between entities; and State, which records interactions and maintains system integrity. Together, these elements form the foundation for enabling interoperable, context-aware, and scalable digital environments (René and Mapes, 2019, p. 113).

A. Domain, Address & Ownership of a Space (Where)

A Domain is an entity with identity (IEEE Computer Society, 2024). To register the identity of any user, space, or asset on the Spatial Web, we need a globally unique "decentralized identifier" (DID) based on W3C Standards (René and Mapes, 2019). The unique identifier receives the name of Spatial Web Identifier (SWID); anything in the Spatial Web that is addressable shall have a SWID defined using W3C did-core (IEEE Computer Society, 2024). Finally, a domain authority is an Entity, usually a person or organization, that is approved to define within a domain its norms and terms. The Universal Domain Graph (UDG) records the relationships between all these Domains, all with known SWIDs (IEEE Computer Society, 2024).

B. Program: Governance, Rules & Permissions (Who can do What, Where, When, How)

First, to register the identity of any user, space, or asset on the Spatial Web, an individual or organization must first create an account and request a SWID (René and Mapes, 2019). These SWIDs may be registered in a system of distibuted, decentralized registeries (IEEE Computer Society, 2024). Using these identifiers, we can establish norms, rules, and permissions within a domain. Similarly, credentials and claims can be handled through a decentralized registry management. Domain authorities are the ones in charge of issuing credentials and setting interaction norms for their domains, in our case, digital twins. Thus, verifying credentials, identities, and access rights requires a combination of cryptographic techniques, authentication protocols, and access control mechanisms. These must be established in order to successfully implement the Spatial Web standard.

C. Protocol: Communication Between Addresses (What to Where)

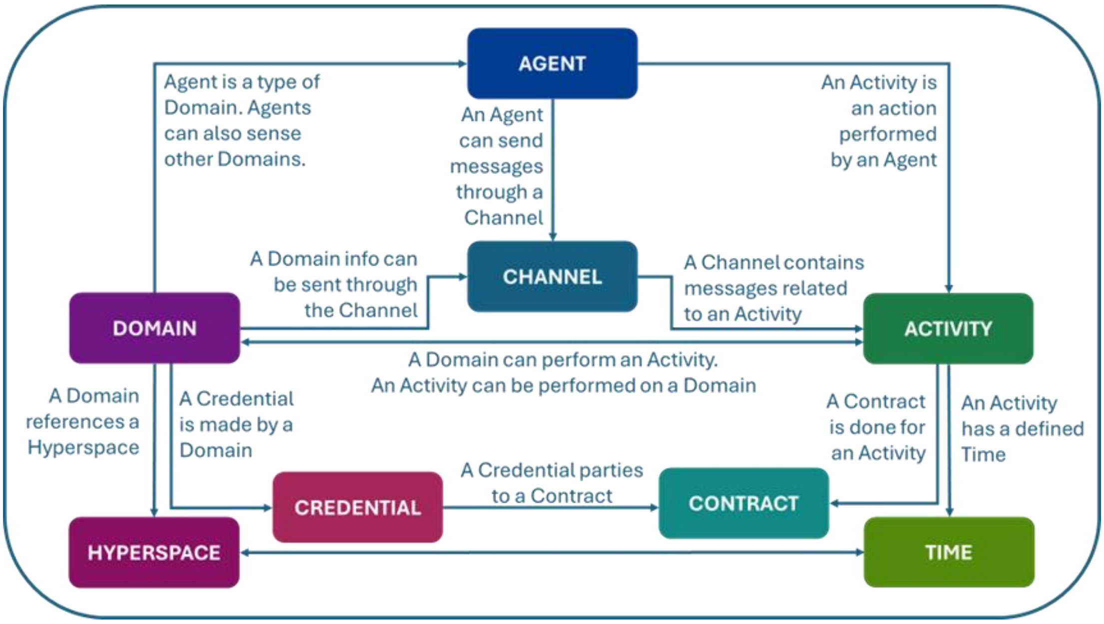
To implement HSTP, systems must support agent discovery, authentication, and message validation for secure communication across the Distributed Computing Continuum, from edge devices to cloud infrastructure. HSTP messages, encoded in HSML, enable standardized data exchange between Domains, Agents, and Devices (IEEE Computer Society, 2024). Implementation requires integrating HSTP with existing network layers, ensuring scalability and real-time responsiveness. Efficient routing, verification, and synchronization mechanisms must support transaction processing, from simple commands to complex high-dimensional data exchanges, enabling interoperability in Spatial Web applications like Digital Twins and autonomous systems.

D. State: Records of Interactions (Who did What, When, Where, and How)

Tracking changes and maintaining data integrity are essential for digital environments. HSML provides a structured way to model diverse data types and map them into a knowledge graph that captures their interrelationships and dependencies. At the core of this structure is the HSML ontology, which defines key elements or classes that categorize entities in the Spatial Web. The highest class in the hierarchy is Entity. Entities are the base items used across the Spatial Web in HSML, and all other classes are subtypes of the Entity base class. These include, but are not limited to, Activity, Agent, Contract, Channel, Credential, Domain, Hyperspace/Space, and Time (IEEE Computer Society, 2024). These classes allow us to identify, categorize, and relate both physical and virtual elements, forming a standardized representation of reality. Their meaning and relations are represented more clearly in Figure 2.2.

Figure 2.2

Map of the Entity subtypes that conform the HSML ontology.



Note. This diagram shows the relationships between core HSML entity subclasses. Agents are specialized Domains that can perform Activities and send messages through Channels. Domains can issue Credentials and enter Contracts. Activities occur over Time and involve communication via Channels. Adapted from (IEEE Computer Society, 2024).

HSML encodes Entities in a machine-readable format such as JSON, ensuring interoperability across platforms. Like HSTP, the development of HSML requires meeting certain implementation requirements, including mechanisms for data structuring, querying, and updating (IEEE Computer Society, 2024). Additionally, a database system is needed to store, track, and manage these entities, their modifications, and their interactions. The Spatial Web standard discusses approaches for storing knowledge graphs, potentially including vector databases for efficient querying and retrieval.

2.3.3. Spatial Web Challenges for Digital Twins Interoperability

From analyzing the Spatial Web standard and the literature on digital twins, emerging technologies, and communication challenges for space exploration, several key challenges must be addressed to effectively apply the standard for digital twin interoperability. These challenges highlight areas where further development, validation, and adaptation are needed to determine how the Spatial Web can meet the unique demands of space-based systems.

1. **Early Development & Generalization.** The Spatial Web standard is still evolving, with broad guidelines to support various industries. While this flexibility is beneficial, its lack of specificity means additional layers or extensions may be needed for space-based digital twin applications.

2. **Integration Challenges.** Space systems rely on established protocols for communication and data exchange. Aligning the Spatial Web standard, including HSML and HSTP, with these existing frameworks may require significant adaptation.

3. **Scalability & Performance.** DTs for space exploration involve massive, real-time data streams. We need to evaluate whether the Spatial Web standard can efficiently manage this scale, especially in environments with limited computing resources, and determine what adaptations may be necessary.

4. **Practical Implementation Gaps.** The standard outlines key principles but lacks detailed implementation examples for digital twins. Bridging the gap between theory and real-world application is essential, particularly for validating HSML's ability to accommodate diverse DT data.

5. **Governance & Security Complexity.** The Spatial Web introduces governance mechanisms for data trust and access control. Applying these in space missions involves regulatory challenges, cross-organization collaboration, and ensuring robust cybersecurity measures.

3. METHODS

Chapter 3 outlines the methodology used to both develop and test a prototype framework for achieving digital twin interoperability and data trust in the context of space exploration. This chapter details the creation of the HSML schema and supporting tools, including the HSML API, registration mechanisms, and authentication/authorization system, which were essential for implementing the Spatial Web standard and enabling structured data exchange. The experimental phase evaluates this framework through a series of simulations conducted in Unity, Unreal Engine, and Omniverse Isaac Sim. These tests assess the exchange of structured HSML JSON messages, DID-based identity management, and the use of credentials for secure access control. A multi-agent lunar rover scenario serves as the main test case to validate cross-platform interoperability. Performance is measured through metrics such as latency, authentication robustness, and compliance with standardization principles.

3.1. Overview

Based on the literature review conducted, three critical challenges in digital twin technology for space exploration were identified: standardization, interoperability, and data trust. To address these gaps and demonstrate potential solutions, the following experimental framework was designed. This framework leverages simulations across multiple platforms (Unity, Unreal Engine, and Omniverse Isaac Sim) to evaluate interoperability, validate the Spatial Web standard, and assess mechanisms for building data trust in digital twin systems.

The experimental setup consists of the following key components:

- **HSML Schema:** An extended implementation of the Hyperspatial Modeling Language, used to define a common data structure for entities and interactions in JSON format, enabling semantic alignment and standardized data exchange.
- **HSML API:** A RESTful API built using FastAPI that serves as the main interface for entity registration, authentication, and authorization. It implements the Hyper-

space Transaction Protocol (HSTP) defined in the Spatial Web standard, ensuring secure and standardized message exchange. The API interacts with the Kafka messaging system and the SQL registry to enforce identity and access control across platforms.

- **Authentication and Authorization Mechanism:** Identity and access control are managed using the DID:key method. Users must prove ownership of their private keys before being allowed to publish or subscribe to messages, ensuring secure communication between entities.
- **SQL Database:** A MySQL-based registry that stores entity metadata, public identifiers (SWIDs), credentials, and topic associations. It acts as the persistent backend for identity verification and authorization tracking.
- **3D Models and Simulations:** Digital representations of lunar rovers and environments created in Unity, Unreal Engine, and Omniverse Isaac Sim, designed to test interoperability across simulation platforms.
- **Scripts & Business Logic Plugins:** Custom extensions that parse and transform HSML messages into platform-specific actions. These plugins serve as the middleware between the HSML API and the simulation platforms, translating structured data into simulation actions.

3.2. HSML Schema Development

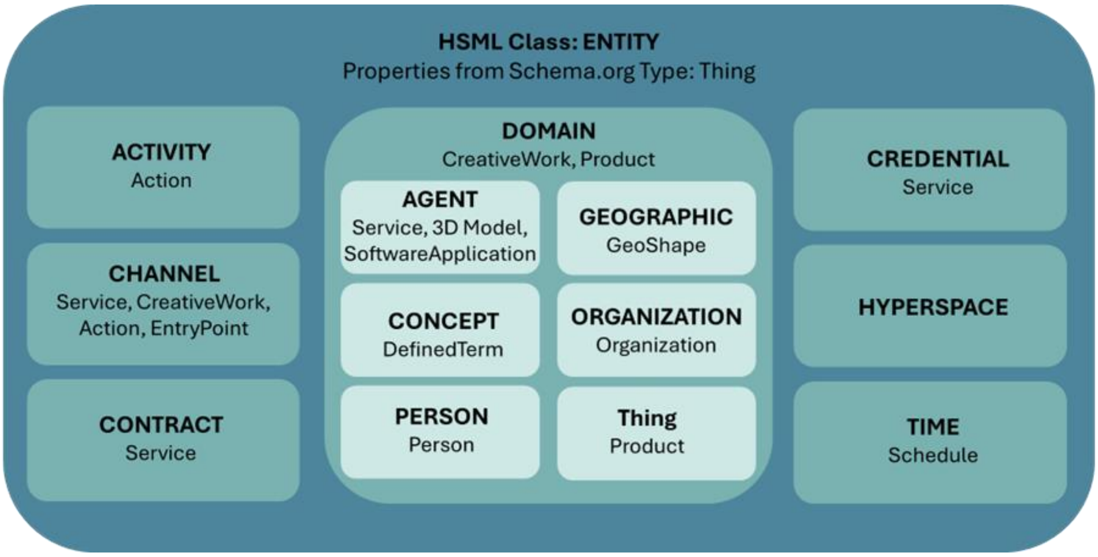
To enable interoperability and data trust in Digital Twin environments, the Hyperspatial Modeling Language (HSML) was extended and implemented based on the high-level description provided in the Spatial Web standard. This process involved constructing the HSML schema structure, utilizing Schema.org as a foundational vocabulary, and enabling its use across multiple simulation platforms.

The development of the HSML schema began by identifying the core ontology classes defined by the Spatial Web standard, such as Entity, Agent, Activity, Credential, Hyperspace, and Time. These were structured into a class hierarchy, with Entity as the base class from which the others inherit. Definitions and requirements were obtained directly from the standard to ensure conceptual consistency.

Once the core classes were defined, they were mapped to equivalent types in Schema.org where possible. Schema.org served as a foundational vocabulary to promote compatibility and ease adoption. For each HSML class, existing Schema.org properties were reused when relevant. Figure 3.1 captures the hierarchy of HSML classes defined in the schema, along with the corresponding Schema.org types from which properties were drawn. When no equivalent property existed, new custom properties were created to capture the necessary semantics. It is important to note that Schema.org was used as a starting point; future extensions may incorporate other domain-specific schemas, allowing the HSML model to evolve based on application needs.

Figure 3.1

HSML Ontology Classes Hierarchy and their correspondent Schema.org Types



Note. The HSML clasess are displayed according to their hierarchy, each box indicating of which Entity they are a subclass of. Below each HSML class, the correspondent Schema.org types can be found, from which it draws properties inside the HSML Schema.

A key component of the implementation was the creation of a hsml.jsonld context file, which formalizes the vocabulary and structure of the schema. This file constructs a semantic graph of the entities and their interrelationships. For testing purposes, it was hosted on GitHub Pages, although more permanent hosting is planned for future deployments.

To validate and apply the schema, HSML-compliant JSON files were used as the standard format for structured data exchange across simulation platforms. These files encode both static metadata (e.g., identity, type, description) and dynamic state information (e.g., position, rotation) in a consistent, machine-readable format. A typical HSML JSON file includes the following fields:

- @context: URL pointing to the hsml.jsonld file in GitHub Pages
- swid:a DID:key identifier that uniquely identifies the Entity
- @type: the HSML class type (e.g., Agent, Activity)
- name, description, and additional metadata such as creator, linkedTo, or domain-specific properties.

By encoding data in this way, the HSML schema enables consistent interpretation and exchange of information between systems. While the Spatial Web ontology remains at the core, the use of Schema.org enhances flexibility, allowing other ontologies to be added as required. Figure 3.2 illustrates an example HSML JSON file representing a digital twin of a lunar rover, structured according to the schema.

Figure 3.2

Example of an HSML JSON File for an Agent

```
{
  "@context": "https://digital-twin-interoperability.github.io/hsml-schema-context/hsml.jsonld",
  "@type": "Agent",
  "name": "Lunar Rover A",
  "swid": "did:key:6Mkq47kEVpasYSLxc2DM1951tSw2hPQyT7nhdVrnT6YawM5",
  "description": "A digital twin rover used for lunar exploration mission simulations.",
  "creator": {
    "@type": "Person",
    "name": "Alicia Sanjurjo",
    "swid": "did:key:6MktsGJxcNwZaC1vuimSYai7Zs9ykPwwrAfeVQtMLDU3nqQ"
  },
  "platform": "Unity",
  "dateCreated": "2025-03-01",
  "linkedTo": [
    {
      "@type": "Activity",
      "name": "Surface Sampling"
    }
  ]
}
```

Note. The JSON file uses the HSML schema context to ensure interoperability. Properties are drawn from Schema.org where applicable and extended with HSML-specific metadata as needed.

3.3. HSTP Communication Protocol Implementation

To enable secure and trusted interactions between digital twins, the Spatial Web standard introduces a communication framework based on verifiable identities and contextual messaging. At its core, this relies on Spatial Web Identifiers (SWIDs) for uniquely identifying entities, and Credentials to authorize access and define relationships. The Hyperspace Transaction Protocol (HSTP) governs how these messages are exchanged, ensuring interoperability and accountability across distributed systems. This section outlines how these foundational elements—identification, verification, and secure communication—were implemented in this prototype system.

3.3.1. The SWID & the DID:key Method

In the Spatial Web standard, each entity in the system must be uniquely identified to enable traceability, accountability, and secure communication. This is the role of the Spatial Web Identifier (SWID). However, the standard does not yet define an official method for generating SWIDs. To implement identity and authentication within the scope of this project, we adopted the DID:key method, a lightweight and self-contained approach from the W3C Decentralized Identifiers (DID) specification.

The DID:key method was chosen over alternatives such as DID:web or DID:ethr because it is fast, does not require domain registration or interaction with a blockchain, and involves no hosting or financial costs. These features make it particularly suitable for experimentation and simulation scenarios, as well as for edge computing environments like those found in space systems.

In practice, a Python-based cryptographic tool was used to implement the method. The tool generates a public/private key pair using the Ed25519 algorithm, which is known for its speed and strong security properties. The public key is then transformed into a DID of the form `did:key:<identifier>`, which serves as the SWID in the HSML JSON, as shown below. The private key is stored locally in a `private_key.pem` file and remains confidential.

```
"swid": "did:key:6Mkq47kEVpasYSLxc2DM1951tSw2hPQyT7nhdVrnT6YawM5"
```

The uniqueness of the DID:key method is mathematically guaranteed. It uses a cryptographically secure random number generator (CSPRNG) to generate private keys, resulting in approximately 2^{256} possible combinations; making the probability of collision effectively zero. Public keys are derived deterministically and encoded using multicodec prefixes and Base58, ensuring a consistent and interoperable format. This key-based identification system supports the authentication and authorization protocols implemented in the experiment.

3.3.2. MySQL Database for Storage

To manage identity verification and access control within the Spatial Web implementation, a structured and persistent data storage solution is required. In this project, a MySQL relational database was chosen to serve as a registry for HSML entities, storing both their SWID (in the form of a did:key public identifier) and associated metadata (encoded in HSML JSON format). This database enables both authentication—verifying an entity's identity—and authorization—granting access to domains based on registered credentials.

While the Spatial Web standard ultimately anticipates the use of more sophisticated graph databases for navigating the Universal Domain Graph (UDG) (IEEE Computer Society, 2024), a MySQL database was selected for this prototype due to its simplicity, widespread adoption, and native support for JSON storage. MySQL's ability to handle complex queries, support multi-user access, and offer structured, long-term data management makes it a practical choice for this proof-of-concept phase.

The database used in this implementation is called did_registry, and contains a single table, did_keys, structured as shown in Table 3.1. This structure supports the core verification functions needed for the implementation: associating agents with credentials, registering domains, managing authorized access, and mapping relationships between entities. It also enables rich metadata queries that are key to enabling trust-based interoperability across platforms.

Table 3.1

Structure of the did_keys Table in the MySQL Registry

Field	Type	Description
id	Int	Row number automatically generated.
did	Varchar(255)	The SWID of the Entity being registered (did:key:xyz123).
registered_by	Varchar(255)	The SWID of the Person/Organization that registered the Entity.
public_key	Text	The public key corresponding to the DID.
metadata	JSON	HSML-encoded JSON metadata associated with the Entity.
created_at	Timestamp	Timestamp of when the record was created.
kafka_topic	Varchar(255)	Kafka topic associated with an Agent for pub/sub messaging.
allowed_did	Text	List of SWIDs allowed to interact with this Entity (based on Credentials).

Note. This table defines the configuration used for the *did_keys* table in the MySQL registry database. It serves as the verification layer for SWIDs (DIDs) and their metadata, supporting the authentication and authorization components of the Spatial Web prototype.

According to the Spatial Web standard, future queries on the Universal Domain Graph are expected to be expressed in HSML and potentially translated from SQL or GraphQL (IEEE Computer Society, 2024). As such, MySQL remains a compatible backend option in the short term, while providing a pathway toward more context-aware and situated queries in future deployments.

3.3.3. The Verification Logic

In the context of enabling secure and interoperable interactions between digital twins using the Spatial Web standard, the verification system implemented in this study ensures that every entity can be uniquely identified, authenticated, and authorized to interact with others. This section describes the logic behind the verification workflow, the use of cryptographic identifiers, and the authentication/authorization process for publishing and subscribing to Kafka-based communication topics.

The verification logic is built upon three core components: the registration of entities, the use of Decentralized Identifiers (DIDs) for identity management, and the validation of access rights through a MySQL registry. The central principle is that each Entity in the system has a unique Spatial Web Identifier (SWID) based on the DID:key method, and a corresponding private key that serves as a password.

Entity Registration Process

The registration of new HSML entities is managed through a Python-based tool. The process involves:

- JSON validation: The system verifies that the uploaded file is a valid HSML-compliant JSON, and that it contains all required fields based on its @type (e.g., Person, Agent, Credential).
- DID generation: Using the cryptographic tool, a cryptographic key pair (Ed25519) is generated. The public key is formatted as a DID:key and attached to the swid field in the entity´s JSON. The private key is then provided to the user for secure storage, as it is required for future authentication or registration actions.
- Metadata storage: The updated JSON and public key are stored in a MySQL database (did_registry), which serves as a persistent registry.
- Login verification: Only registered users (Persons or Organizations) are allowed to register new entities. Their identity is verified using the private key.

To test the verification framework, a minimum set of three HSML entity types was used: Person, Agent, and Credential. Figure 3.3 shows these minimum required entities. These entities cover the core components of the authentication and authorization logic and are sufficient to validate the interaction pipeline. Each entity has a distinct registration process:

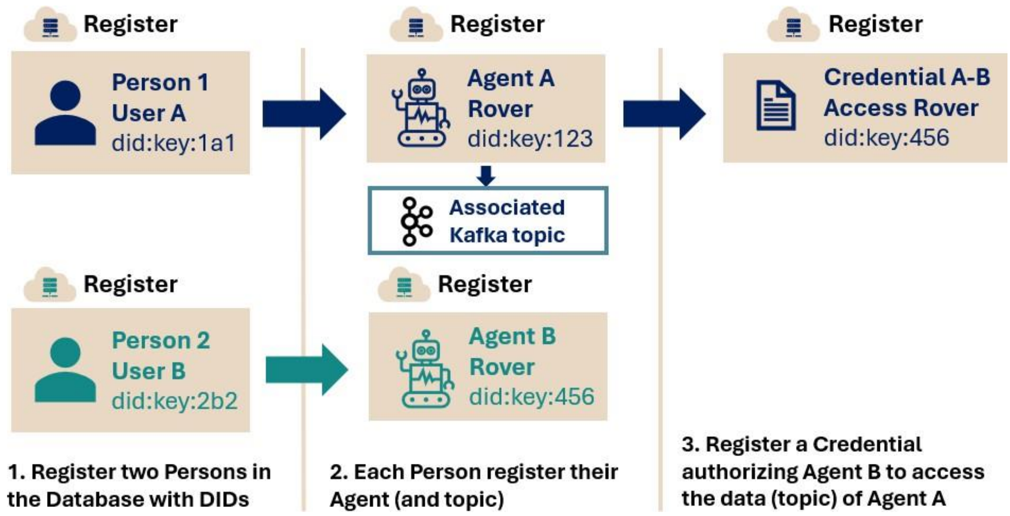
- **Person:** A new user begins by registering themselves as a Person entity. This together with Organization are the only types allowed to be created without logging in. The user uploads a valid Person-type HSML JSON, and the system generates a DID key pair, stores the JSON in the registry, and returns the private key to the user.

- **Agent:** Once logged in, a user can register an Agent. The JSON must include required metadata such as name, creator, dateCreated, and description. Upon registration, a Kafka topic is automatically created for the Agent, enabling future message exchange.

- **Credential:** A logged-in user can also register a Credential entity, which defines authorization rights between entities. The system checks that the issuer (the user) is the legitimate owner of the referenced Agent, and it updates the allowed_did list of that Agent accordingly.

Figure 3.3

Minimum Entities Needed to Register to Test Verification Logic



Note. This figure illustrates the minimum set of entities required to test the verification logic within the HSML API. Two persons (User A and User B) are first registered with their decentralized identifiers (DIDs). Each person then registers a corresponding agent (rover) along with an associated Kafka topic. Finally, a credential is registered that authorizes Agent B to access the topic associated with Agent A, completing the setup needed to test identity registration, authentication, and authorization.

Authentication and Authorization

Before entities can communicate, the system ensures that they are authenticated (who they are) and authorized (what they can do):

- **Producer Authentication:** Before an agent is allowed to publish data to a Kafka topic, the authentication procedure ensures it is the legitimate owner of that topic. The authentication flow is as follows:

  1. The user must provide the `private_key.pem` of the producer Agent.
  2. Using a cryptographic tool, the public key (DID:key) is derived from the private key.
  3. This derived DID is then checked against the MySQL registry database to confirm that it matches the SWID of the Agent registered for that Kafka topic.
  4. If the identity is verified successfully, the agent is authenticated and allowed to publish messages to the topic during that session.

- **Consumer Authorization:** To ensure that only authorized entities can access data from a given Kafka topic, the following authorization procedure is performed:

  1. The user must provide the `private_key.pem` of the consumer entity.
  2. The public key (DID:key) is derived from the private key using the cryptographic tool.
  3. This derived DID is verified against the `allowed_did` list in the MySQL database for the producer Agent associated with that Kafka topic. This authorization is granted via the issuance of a Credential beforehand.
  4. If the DID is present in the authorization list, the consumer is granted access to subscribe to the topic.

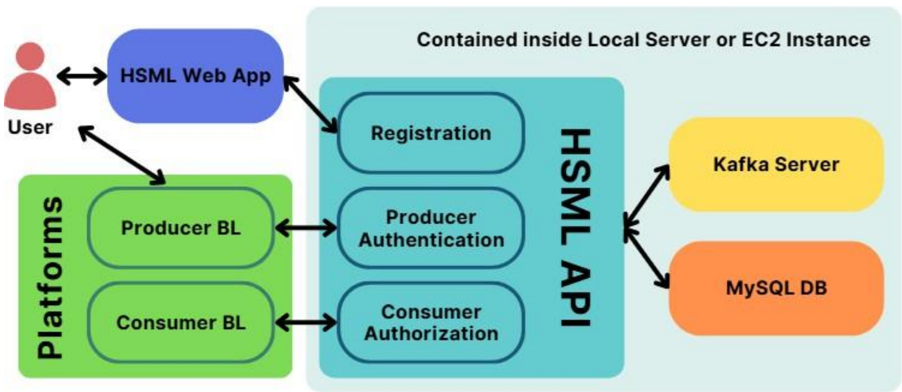
3.3.4. The HSML API

To implement the HSTP protocol and provide a structured interface for exchanging messages in HSML format, the HSML API was developed. This API acts as an intermediary layer between the simulation platforms and the backend infrastructure, including the Kafka server and MySQL registry database. Its primary role is to enable secure and standardized data exchange between digital twin instances by handling authentication, authorization, and registration processes.

The HSML API was developed using FastAPI and follows a RESTful architectural style. A RESTful API, or Representational State Transfer API, is a type of web service that adheres to principles such as stateless communication, resource-based endpoints, and standard HTTP methods like GET, POST, and DELETE. These characteristics make RESTful APIs simple to use, scalable, and platform-agnostic; making them an ideal fit for distributed environments like digital twin systems. In the case of the HSML API, each interaction (e.g., registering a new entity, authenticating a producer, or authorizing a consumer) is handled through independent HTTP requests that send and receive structured JSON data. This stateless and modular structure ensures that the API can be deployed flexibly across different systems or locations, such as a local lab server or a cloud-hosted instance.

Figure 3.4

HSML API Architecture Block Diagram



Note. Block diagram showing the internal architecture of the HSML API, including registration, authentication, and authorization components, and their interaction with external platforms, the HSML Web App, MySQL database, and Kafka server.

The HSML API allows simulation platforms to connect and verify their identities via private key authentication before they are allowed to exchange data through Kafka topics. It also connects with the MySQL registry database to validate credentials and authorization rights. In other words, simulation platforms are not permitted to publish or subscribe to Kafka topics unless they have authenticated themselves at the beginning of the session and passed all necessary verification checks.

The architecture of the HSML API is composed of the main application and three specialized routers:

- **Registration Router:** Used to register new entities. This router is typically called by the front-end Web App, developed by project member Gerald Parish, which provides a graphical interface for interacting with the registration system.
- **Authentication Router:** Used by producers. It verifies the identity of the sender through their private key and allows them to publish HSML messages to a Kafka topic if successfully authenticated.
- **Authorization Router:** Used by consumers. It ensures that a subscriber holds a valid Credential authorizing access to the data stream before allowing them to receive messages from a Kafka topic.

Supporting these routers is a shared cryptographic tool used for operations like extracting the public DID from a private key and generating secure key pairs during registration. The API and its cryptographic tool were bundled into a Python package, allowing for easy deployment across multiple nodes within the test environment.

By consolidating the logic for registration, authentication, and authorization into a single cohesive service, the HSML API effectively enables the secure and scalable communication layer envisioned by the Spatial Web standard.

3.4. Experimental Setup

The system architecture developed in this thesis was realized within the context of the lab's broader use case: collaborative lunar rover assistance, specifically a rescue scenario involving multi-agent coordination. While this scenario was not fully simulated, it provided a guiding framework for developing the interoperability and trust infrastructure. The experimental setup consists of the following phases:

1. **Preliminary Connectivity Tests:**

- Establishing communication between 3D models in Unity, Unreal Engine, and Omniverse Isaac Sim using a Kafka server.
- Implementing the HSML schema for structured data exchange between platforms.
- Validating the verification system using DID:key and SQL database to ensure secure identity registration and storage.

2. **Core Experiment: Interoperability and Verification**

   - Demonstrate synchronized data exchange between simulation platforms using the HSML schema.
   - Evaluate identity-based message authentication and topic-level authorization using the HSML API.
   - Assess performance of real-time communication, including message flow latency and system response under secured conditions.

3. **Assessment of Standardization & Data Trust**

   - Analyze data integrity, security, and schema consistency during platform communication.
   - Evaluate the effectiveness of HSML and HSTP as a standardization layer for secure digital twin interoperability.
   - Measure key system metrics, such as latency, data consistency, and cross-platform fidelity.

3.4.1. Simulation Environment & Tools

Hardware Setup

Three computers were used to run the simulations, each playing a different role in the experimental architecture. Their specifications can be seen in Table 3.2.

Table 3.2

Hardware Configuration Used in Simulation Experiments

Specification	Computer 1 (Linux)	Computer 2 (Windows)	Computer 3 (Windows)
CPU	Intel Core i9-13900K (24 cores, 32 threads, 5.8 GHz max)	Intel Core i9-13900K (32 logical processors, ~3.0 GHz)	[To be added]
GPU	NVIDIA RTX 4090	NVIDIA RTX 6000 Ada Generation	[To be added]
RAM	128 GB	128 GB	[To be added]
Storage	2 TB Samsung SSD 970 EVO Plus (NVMe)	2 × 2 TB Samsung SSD 970 EVO Plus (NVMe)	[To be added]
Role	Kafka, HSML API, MySQL DB, Simulation client (Omniverse)	Simulation client (Unity)	Simulation client (Unreal Engine)
OS	Ubuntu Linux	Windows 10 Home	[To be added]
Network Address	192.168.1.55	192.168.1.55	192.168.1.55

Note. This table summarizes the hardware configuration of the simulation environment used to evaluate interoperability and protocol efficiency. All machines are networked locally and used in real-time experiments.

Software Stack

- **Simulation Platforms:** Unity (2021.3.1f1), Unreal Engine 5.4.4., and Omniverse Isaac Sim (2023.1.1) are used to simulate rover behaviors and environments, each acting as a Digital Twin platform capable of producing or consuming data streams.

- **Programming Languages:** Python (main language for backend and APIs), C# and C++ (used in simulation platforms and plugin integrations).

- **Data Exchange Middleware:** Apache Kafka is used for real-time, topic-based message passing between simulation platforms. Topics are created per Agent to manage pub/sub data streams.

- **Data Management:** MySQL is a structured SQL database that is used to store SWIDs (DIDs), entity metadata, and authorization information.

- **Communication Standard:**

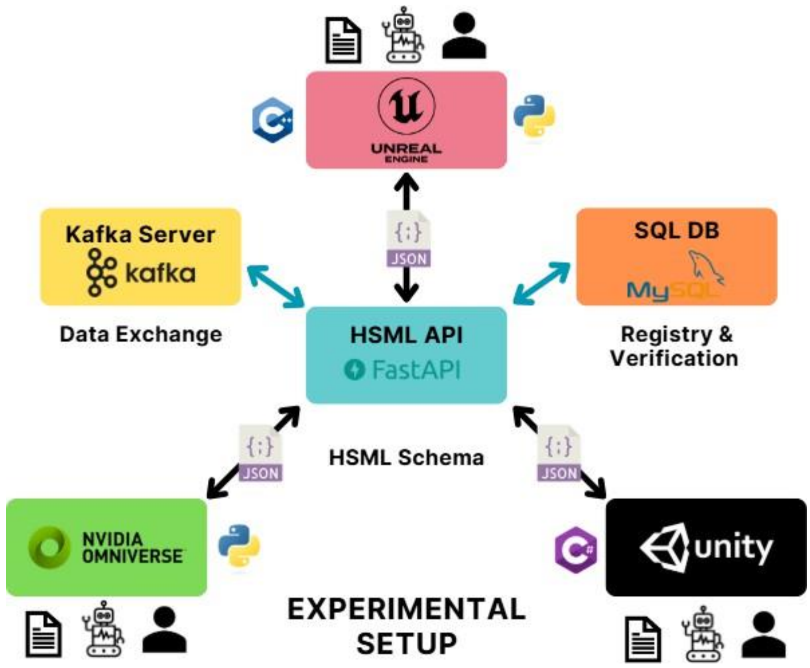
- **The HSML (Hyperspace Modeling Language) schema** enables structured and interoperable data exchange through JSON files.
- **The HSTP (Hyperspace Transaction Protocol)** serves as the underlying communication protocol architecture guiding interaction and message flow.

- **Verification & Registry Mechanism:** The system uses the DID:key method to generate for each HSML Entity a unique SWID, used for identity verification and access control.

- **Applications & Interfaces:** HSML API (FastApi) and Web App (Flask)
  - **The HSML API** (FastAPI) acts as a bridge between simulation platforms, Kafka messaging, and the MySQL registry.
  - **The HSML Web App** (Flask) is a web-based graphical interface for registering new entities through the HSML API.

Figure 3.5

Experimental Setup Graph



Note. This figure presents the full experimental setup used in the second simulation. It displays the platforms involved (Unity, Omniverse, Unreal Engine), along with the programming languages used to implement their respective Business Logic (BL) plugins. Each agent was registered with associated user and credential information via the HSML API, which handles registration, authentication, and authorization using a MySQL database. HSML JSON messages were exchanged between platforms using Kafka through the HSML API.

3.4.2. Simulation Scenarios

This subsection outlines the technical experiments and unit tests that validate the infrastructure developed to support the Lunar Rover Collaboration use case.

Use Case Context: Lunar Rover Collaboration

The system architecture was realized within the context of the lab's broader use case: collaborative rover assistance during lunar exploration. The envisioned scenario involves three rovers:

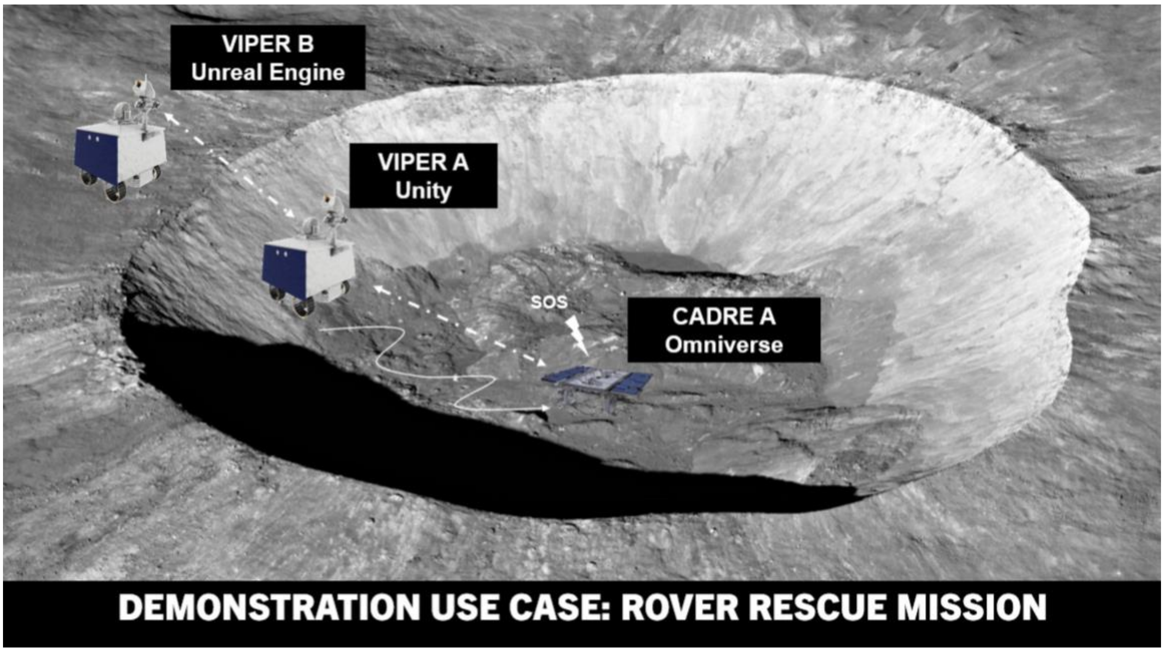
- A stuck rover requiring extraction (Cadre A).

- An assisting rover that provides mechanical support (Viper A).
- A relay rover that establishes communication with mission control (Viper B).

It should be noted that this thesis does not simulate rover interaction across physics engines. Instead, it focuses on developing the infrastructure that would enable this scenario: real-time, cross-platform data exchange, secure identity verification, and trusted communication between agents operating in different simulation environments. The use case is illustrated in Figure 3.6.

Figure 3.6

Lunar rover collaboration use case with three simulated agents.



Note. This figure illustrates a conceptual scenario involving three NASA rover models: one CADRE rover (developed by JPL) hosted in Unity, and two VIPER rovers (developed by NASA Johnson Space Center), placed in Omniverse and Unreal Engine respectively. These models represent a potential lunar assistance mission involving extraction and communication relaying.

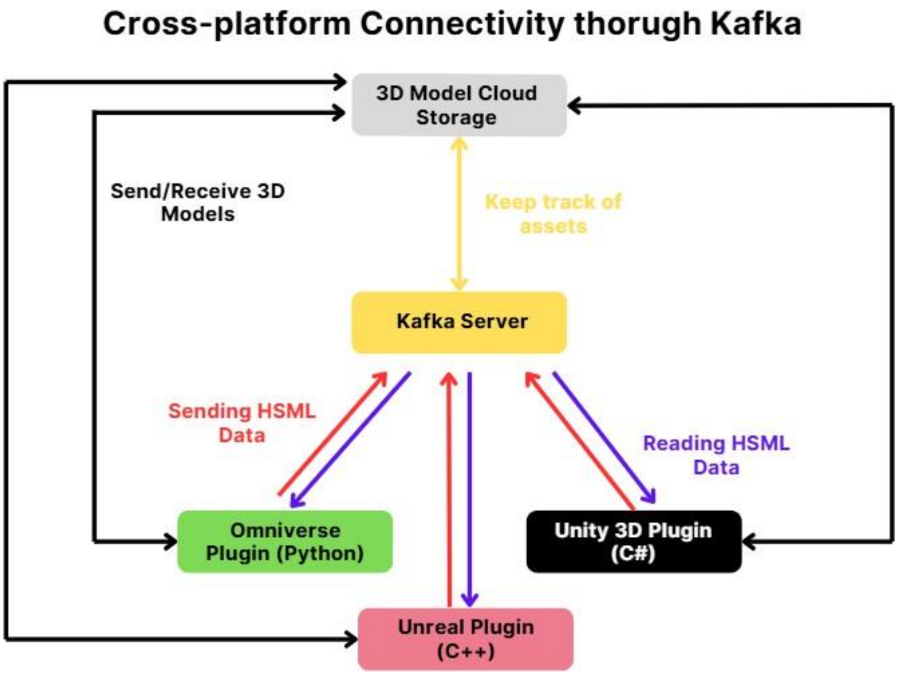
The experiments below validate the key technical components needed to support such a scenario.

1. **Experiment 1: Platform Connectivity and HSML Messaging**

   - Objective: Demonstrate interoperability and real-time synchronization between Unity, Unreal Engine, and Omniverse Isaac Sim using the HSML schema.
   - Method: Identical 3D models are deployed across platforms, with state synchronization achieved through structured messages defined by the HSML schema. Kafka is used as the messaging backend. This validates basic connectivity and message format alignment.

Figure 3.7

Architecture of Experiment 2 — Cross-Platform Interoperability Using HSML Schema



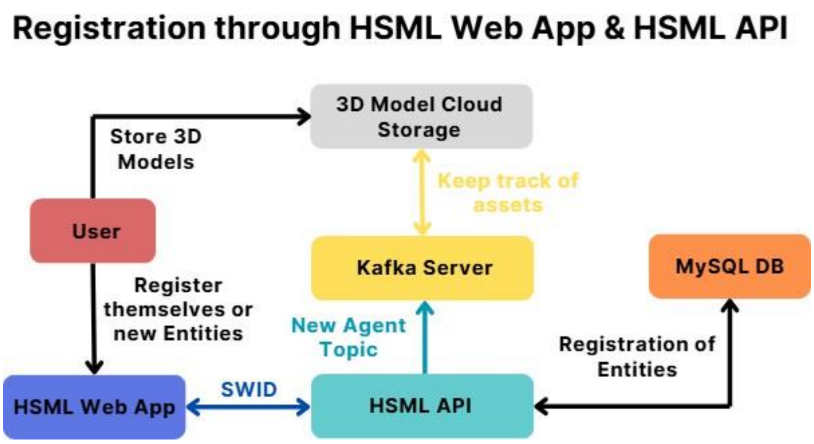
Note. This diagram, created by teammate Jared Carrillo, illustrates the architecture for Experiment 1, in which each simulation platform (Unity, Omniverse, Unreal) directly connects to a shared Kafka server to exchange HSML-formatted messages. Plugins developed in C#, Python, and C++ allow the platforms to send and receive position, rotation, and other runtime data through dedicated Kafka topics.

2. Unit Tests: Verification Logic

- Objective: Demonstrate the use of HSTP-based mechanisms for identity registration, authentication, and authorization.
- Method: Register a basic schema (2 Persons, 2 Agents, 1 Credential) using the HSML Web App and the HSML API, authenticate a producer using DID:key, and authorize a consumer for a specific Kafka topic. Measure storage time and verify correct entries in the SQL database.

Figure 3.8

Architecture of the Preliminary Registration through the HSML Web App & HSML API



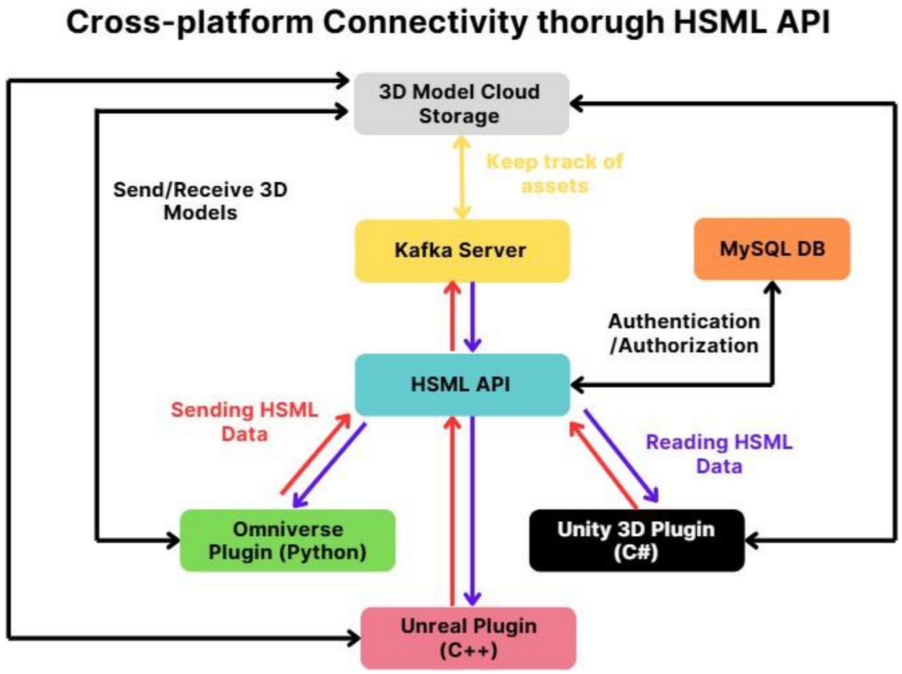
Note. The diagram illustrates the registration phase conducted prior to the simulation. Users register themselves and their entities through the HSML Web Interface, which generates a SWID and stores entity metadata in a MySQL database. The HSML API also creates a dedicated Kafka topic for each new agent and keeps track of assets via the 3D model cloud storage.

3. Experiment 2: Secure Cross-platform Messaging thorugh the HSML API

- Objective: Enable trusted cross-platform communication using authenticated and authorized message flows.
- Method: Platforms communicate through the HSML API, which enforces verification checks before allowing data exchange via Kafka. This experiment tests end-to-end integration of all components under real-time conditions.

Figure 3.9

Architecture of Experiment 2 — HSTP-Based Cross-Platform Interoperability with HSML API



Note. During the simulation, the Omniverse, Unity, and Unreal platforms connect to the HSML API to send and receive HSML messages. The API enforces authentication and authorization by querying the MySQL registry before granting access to Kafka topics, effectively serving as a secure gate between platforms and the messaging system.

3.5. Data Collection & Analysis

3.5.1. Logging Mechanisms & Analysis Tools

To monitor and validate the functionality of the framework, various logging and analysis methods were used throughout the experiment:

- **Kafka server logs** were used to confirm message exchange between producers and consumers, including message timestamps and delivery status.
- **MySQL database entries** were checked to ensure entities were correctly registered, including their metadata, credentials, and access permissions.
- **Platform logs** from Unity, Unreal Engine, and Omniverse were reviewed to verify

successful ingestion of JSON files and simulation responses.

- **Python scripts** were used to inspect and validate the structure of the HSML JSON files, making sure they followed the schema and contained the required fields.
- **Screen recordings** were captured during test runs to provide a visual record of interactions and to cross-validate with backend logs.

Together, these tools allowed for step-by-step debugging, tracking the message flow, and confirming that entities interacted securely and correctly.

3.5.2. Evaluation Metrics

To assess the effectiveness of the proposed framework, the following key metrics will be analyzed:

- **Interoperability Success Rate:** Ability to exchange and interpret HSML data correctly across platforms.
- **Authentication Robustness:** Verification that only authenticated entities (using DID:key) were able to publish or subscribe to Kafka topics.
- **Latency & Data Consistency:** Measurement of the time delay between message transmission and reception, and checking that the data received matches what was sent.
- **Standardization Compliance:** Degree to which HSML and HSTP together enable structured, standardized, and interoperable data exchange across platforms.

3.6. Validation & Limitations

To ensure that the system performed as expected, a combination of unit and integration tests were conducted across different components. Each step of the verification framework (registration, authentication, and authorization) was validated individually using test HSML JSON files and manually checked for correctness. For example, the correct association of DID:key public identifiers with entities was confirmed by inspecting entries in the MySQL registry and ensuring the private key was required for access when streaming data. Kafka message flow was also monitored using both console output and topic logs to verify message delivery and ordering. Cross-platform functionality was confirmed by successfully parsing HSML JSON files across Unity, Unreal Engine, and Omniverse Isaac Sim, demonstrating that the schema structure remained consistent and interpretable across systems.

Despite successful validation, the prototype has several limitations. Only three simulation platforms were used (Unity, Unreal Engine, and Omniverse) all of which already support basic JSON and external data handling. This means results may not generalize to closed or proprietary systems. The data exchanged was limited to essential digital twin properties (e.g., name, type, position, rotation), rather than large-scale telemetry or sensor streams. Additionally, while the implementation of HSML and HSTP followed the Spatial Web standard, not all elements of a fully decentralized ecosystem were realized. For example, identity and access verification relied on a centralized SQL registry, and key management was performed manually. These choices were made to simplify development and testing but represent areas for future work.

4. RESULTS AND ANALYSIS

This chapter presents the results obtained from the development and testing of the proposed framework for secure, interoperable digital twin communication using the Spatial Web standard. It begins with the implementation and evaluation of the HSML schema, highlighting its extensibility and alignment with standardization requirements. This is followed by unit tests and partial evaluations of the HSML API and verification logic prior to platform integration. Two main simulation experiments are then analyzed: the first demonstrates real-time interoperability across Unity, Omniverse, and Unreal Engine using HSML-formatted messages exchanged via Kafka; the second builds on this by incorporating the full verification pipeline through the HSML API. Each experiment is examined in terms of data exchange effectiveness, latency, and conformance to interoperability and security goals. The chapter concludes with insights drawn from both experiments and testing phases.

Note: Waiting on other teammates to also be able to perform the Simulation Experiments with Unreal Engine, but results should not vary.

4.1. HSML Schema Results and Discussion

The HSML schema developed for this project was hosted using GitHub Pages in the following hsml.jsonld context file. The schema is based on the HSML ontology, with its classes and subclasses structured according to the hierarchical model defined in the Spatial Web standard. For each class, properties were reused from Schema.org wherever relevant to promote compatibility and accelerate development. In cases where existing vocabularies were insufficient, custom properties were introduced following the definitions and roles outlined in the Spatial Web ontology. During testing, the HSML schema proved to be both flexible and extensible; as it allowed the addition of new properties as needed to better describe entities and support the structured exchange of data between digital twins across platforms.

Table 4.1

HSML Schema Statistics Overview

HSML Classes	# Schema.org Types Used	Schema.org Types Used	# Schema.org Properties Used	Custom Properties
Entity	1	Thing	12	2
Activity	1	Action	11	11
Channel	4	Service, CreativeWork, Action, EntryPoint	12	7
Contract	1	Service	9	5
Credential	1	Permit	7	7
Domain	2	CreativeWork, Product	64	9
Agent	3	3DModel, MediaObject, SoftwareApplication	30	4
Geographic	1	GeoShape	8	0
Concept	1	DefinedTerm	2	0
Organization	1	Organization	64	0
Person	1	Person	61	0
Thing	1	Product	19	2
Hyperspace	0	none	0	9
Time	1	Duration	13	7
Total = 14	Total = 15		Total = 302	Total = 58

Note. This table shows the number of Schema.org types and properties used for each HSML class, along with the number of custom properties developed. Some properties are reused across classes, so the totals represent unique values.

Table 4.1 presents an overview of the HSML schema developed in this project. Out of the 14 HSML classes, 13 were able to incorporate existing properties from Schema.org types. The only exception was the Hyperspace class, which could not be mapped to any Schema.org type. This is expected, as Hyperspace introduces a novel concept specific to the Spatial Web that is not currently represented in traditional schemas. As a result, this class was entirely defined using custom properties.

In contrast, classes such as Person and Organization are already well supported by Schema.org. Therefore, no additional custom properties were required for these during this experiment, which simplified development. Geographic and Concept also did not include custom properties, but this was primarily due to their limited use in the prototype rather than completeness.

It is also important to note that some Schema.org and custom properties were reused across multiple HSML classes. As such, while the table provides number of properties by class, the actual total number of unique properties, both custom and from Schema.org, is lower than this sum. Overall, reusing Schema.org facilitated faster schema development. However, as the Spatial Web evolves and applications become more specialized, incorporating additional vocabularies may be beneficial. Alternative ontologies such as the OGC SensorThings API and SSN (Semantic Sensor Network) for sensor data, GeoSPARQL for geospatial reasoning, and the W3C Web of Things (WoT) for IoT device interoperability offer promising extensions. Custom properties will continue to play a key role in adapting the HSML schema to meet future domain-specific requirements.

HSML Schema Standardization Compliance

To assess standardization compliance, the HSML schema developed in this study was evaluated against the implementation requirements outlined in Section 6.4.3.4 of the Spatial Web standard, titled *"Requirements and Recommendations — Agent Interactions"* (IEEE Computer Society, 2024). The schema was designed to represent a variety of environments (physical, virtual, and hybrid) using a structured and consistent data format. While the data exchanged between agents is encoded in JSON, semantic interoperability is achieved through the use of JSON-LD for the `@context`, which links each entity to the HSML ontology. This setup fulfills the requirement of providing a common language to ensure that messages are interpreted consistently across diverse agent architectures. Although higher-level features such as automatic conflict resolution or contract negotiation were not part of this implementation, foundational elements like Credential and Contract classes were included in the schema to support such functions in future work. In addition, the extensibility of the schema, through the addition of custom properties, supports the ability of domain authorities to define interaction preferences or rules as needed. Overall, the HSML schema demonstrates alignment with the Spatial Web's standardization goals while remaining flexible and expandable for future interoperability needs.

4.2. Unit Tests & HSML API Evaluation

This section presents the partial evaluations and unit tests carried out during the development of the HSML API and its associated components. These tests were performed before the full simulation integration, to ensure that individual parts of the system were functioning correctly and that the verification mechanisms required by the Spatial Web standard were being met.

The following components were tested individually:

1. **JSON Validation:** HSML JSON files were tested using built-in Python validation scripts to ensure correct format, schema context, and required fields depending on the @type (e.g., Person, Agent, Credential). Files missing mandatory fields were correctly flagged and rejected.

2. **DID:key Generation and Uniqueness:** Key pairs were generated using the cryptographic CLI tool, and public keys were checked for uniqueness in the registry.

3. **Registration & Login Logic:** Tested that users were able to register Person and Organization entities, log in with their private key, and register additional entities on behalf of their account.

4. **Credential Assignment and Access Rights:** Credentials were created through the registration endpoint and manually verified by inspecting the resulting JSON and confirming that the appropriate allowed_did fields were updated in the database. Access relationships were validated through test queries.

5. **Authentication and Authorization (No Platforms):** Python scripts tested the verification logic before directly connecting to Kafka. Private keys were used to extract DID:key identities, and access was manually validated against stored credentials. Successful message exchange confirmed that authentication and authorization worked as expected.

6. **HSML API Operation (Standalone):** The API was launched independently of the platforms, and its endpoints were tested using custom Python and C# scripts developed to simulate platform interactions. Each route (registration, authentica-

tion, authorization) responded correctly to valid and invalid requests, confirming expected behavior.

A summary of test outcomes and timing data is presented in Table 4.X. These results confirmed that all core features worked as expected before being deployed for testing with the platforms. Some minor debugging was required in credential validation and login routing. These insights helped shape the final experimental setup by ensuring that only verified and authorized entities could participate in data exchange.

Placeholder for Unit Tests Results TABLE: Show pass/fail and/or timing per unit.

HSML API Standardization Compliance to HSTP

To evaluate standardization compliance, the HSML API developed in this project was assessed against the implementation requirements defined in Sections 6.4.3.4, 7.1.3.1, and Appendix A.3 of the Spatial Web standard (IEEE Computer Society, 2024). The API supports agent and domain-level interactions through identity registration, authentication, and authorization workflows, all built on top of structured HSML messages. It provides a mechanism for verifying capabilities and credentials, therefore fulfilling the requirement that "HSTP shall provide mechanisms for discovering an Agent's or Domain's capabilities, permissions, and credentials to facilitate seamless interactions" (IEEE Computer Society, 2024). The HSML API handles these checks before allowing access to data channels via Kafka topics.

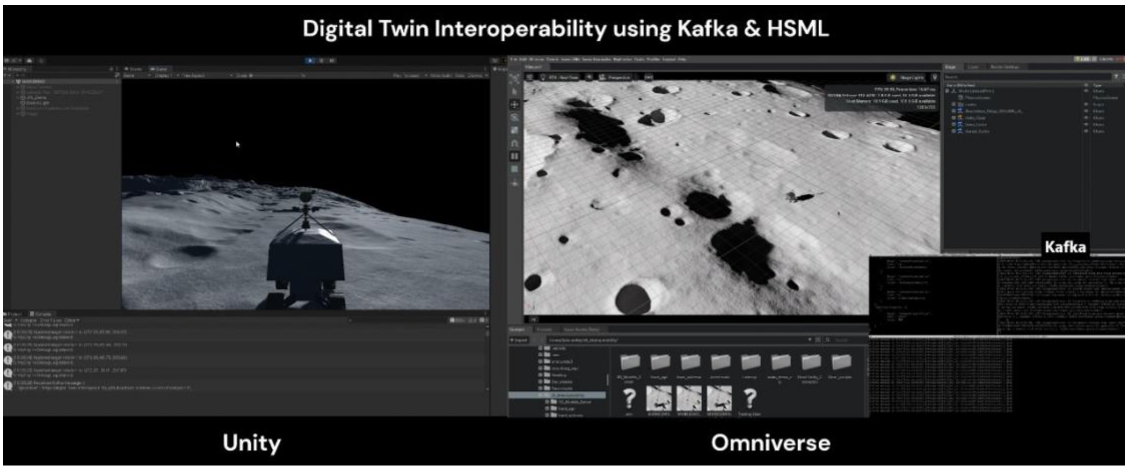
Additionally, the use of DID:key cryptographic identities and a centralized MySQL registry ensures that communication remains both secure and efficient, aligning with the requirement that "HSTP shall help ensure communication protocols are reliable and efficient, facilitating information transmission between Agents regardless of their underlying implementation or the network over which they communicate" (IEEE Computer Society, 2024). While this prototype does not yet support semantically rich, high-dimensional vector representations (a requirement that may be revisited later), it does enable structured JSON-based messaging for positional and state data. Lastly, although full scalability testing was outside the scope of this study, the stateless and modular design of the API was developed with expansion in mind, in line with HSTP's requirement to support large-scale Agent interactions without compromising performance.

4.3. Experiment 1: Cross-Platform Interoperability Using HSML Schema

The first experiment focused on testing interoperability across platforms by evaluating whether digital twins built in Unity, Unreal Engine, and Omniverse Isaac Sim could effectively exchange data using a shared communication format. At this stage, no verification mechanisms were implemented—Kafka was used as the direct messaging middleware, and HSML JSON files were exchanged without authentication or authorization. As described in Chapter 3, each platform relied on its own business logic plugins to parse the incoming HSML messages and render the simulation accordingly. This setup aimed to validate the ability to synchronize simulation behavior across different engines using the Spatial Web´s common schema.

Figure 4.1

Experiment 1 Cross-platform Simulation Screenshot



Note. Unity (left) shows Viper A under manual control, while Omniverse (right) mirrors its movement in real time. Kafka is running in the terminal window, handling the direct communication between platforms through dedicated agent topics. Cadre B, native to Omniverse, is actively being mirrored off screen.

The results, illustrated in Figure 4.1 and supported by video recordings (not embedded here), demonstrate successful two-way synchronization between digital twins across platforms within a simulated lunar environment. In one direction, the Viper A replica inside Omniverse Isaac Sim smoothly mirrors the movements of the original Viper A rover navigating the Moon surface in Unity by subscribing to its Kafka topic, where position and rotation data are published in HSML JSON format. Similarly, the Cadre A rover in

Isaac Sim publishes its own data to a separate Kafka topic, which is subscribed to by its replica in Unity, allowing the Unity-based twin to replicate Cadre A's trajectory across the Moon terrain in real time. These results confirm that both platforms can act as producers and consumers, using HSML-structured data to exchange spatial information reliably. This experiment thus demonstrates effective, real-time interoperability of digital twins simulating rover activity on the lunar surface using the HSML schema.

Table 4.2

Usage of HSML Fields in the Cross-Platform Simulation

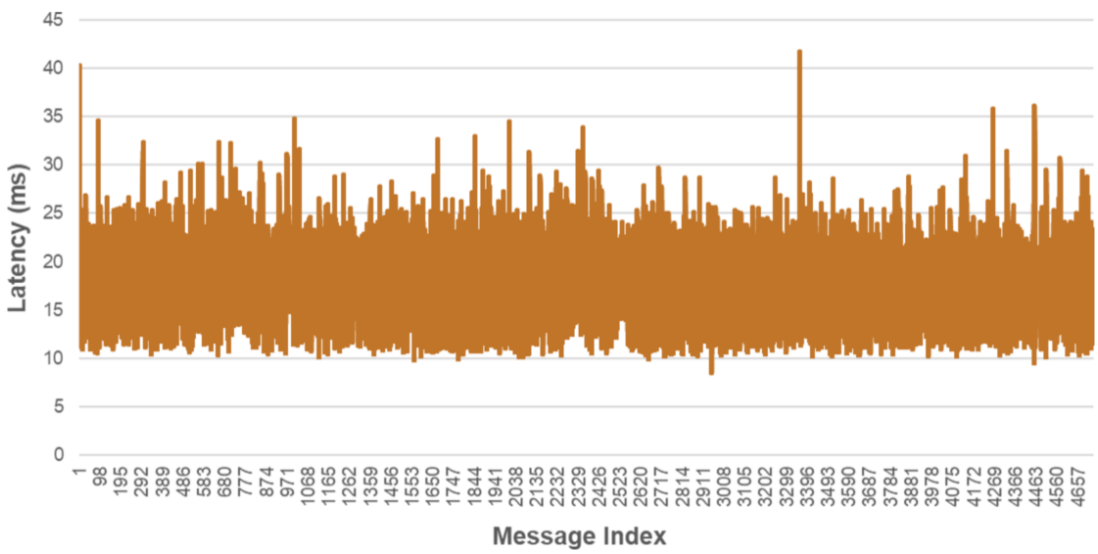
HSML Field	Usage in Simulation	Source
@context	Parsed only	HSML
@type	Parsed only	HSML
name	Parsed only	Schema.org
url	Parsed only	Schema.org
creator	Parsed only	Mixed
dateCreated	Parsed only	Schema.org
dateModified	Parsed only	Schema.org
platform	Parsed only	HSML
spaceLocation	Parsed only	HSML
description	Parsed only	Schema.org
contentUrl	Parsed only	Schema.org
additionalProperty:scale	Parsed only	Mixed
position (Nested Field)		
xCoordinate	Applied	HSML
yCoordinate	Applied	HSML
zCoordinate	Applied	HSML
rotation (Nested Field)		
rx	Applied	HSML
ry	Applied	HSML
rz	Applied	HSML
w	Applied	HSML

Note. This table distinguishes between the HSML JSON properties and whether they were merely parsed or also applied during the simulation. Fields such as position and rotation were used to drive real-time simulation behavior, while metadata like description or creator were parsed for context but did not influence runtime states. The "Source" column indicates whether the property is inherited from Schema.org, defined as a custom HSML extension, or a combination of both.

To better understand how HSML data was processed within each platform, Table 4.2 outlines which fields in the HSML JSON were successfully parsed and which were actively applied in the simulation. While spatial fields like position and rotation directly influenced rover behavior, other fields such as description and creator served as metadata context. In addition to field usage, we also tracked the latency of HSML message exchange across Kafka to evaluate the real-time performance of the system. The results, displayed in Figure 4.2, indicate that during steady-state operation, message transmission latency consistently remained below 50 milliseconds, with an average latency of 18.20 milliseconds. This latency remains well below the threshold of human reaction time, confirming that the system leveraging the HSML schema and Kafka achieves the responsiveness required for real-time digital twin synchronization.

Figure 4.2

Latency Between HSML Messages Sent Directly Thorugh Kafka



Note. Line plot showing latency measurements (in milliseconds) between HSML messages exchanged directly through the Kafka server, using the local lab Wi-Fi network. Average latency = 18.20 ms, max = 41.67 ms, min = 8.53 ms, median = 18.02 ms, std. dev. = 4.42 ms.

Overall, the interoperability success rate for this first experiment was high, with consistent and correct interpretation of HSML-encoded data across all platforms. Both Unity and Omniverse were able to publish and consume messages in the same standardized format, ensuring reliable synchronization of rover behavior. The HSML schema proved effective in providing a shared vocabulary that allowed different engines to interpret data uniformly, demonstrating strong standardization compliance. This confirms that, without additional verification mechanisms, the HSML schema alone enables structured and interoperable communication between heterogeneous digital twin environments.

4.4. Experiment 2: HSTP-Based Cross-Platform Interoperability with HSML API

This second experiment focused on testing end-to-end data verification and secure communication using the full set of verification mechanisms defined in the Spatial Web architecture. Unlike Experiment 1, where messages were sent directly through Kafka, this setup employed the HSML API as an intermediary layer to handle registration, authentication, and credential-based authorization. The goal was to validate whether simulation platforms could still exchange data effectively under these constraints, while enforcing secure access control.

The experimental setup for this simulation built upon the previous experiment by incorporating the full verification pipeline. It included the HSML API, which served as the interface between the simulation platforms and the backend infrastructure. The MySQL database was used to store entity metadata, public keys, and access credentials. Identities were generated using the DID:key method, and access control was enforced through pre-issued Credentials, which authorized each rover to subscribe to the Kafka topic of the other.

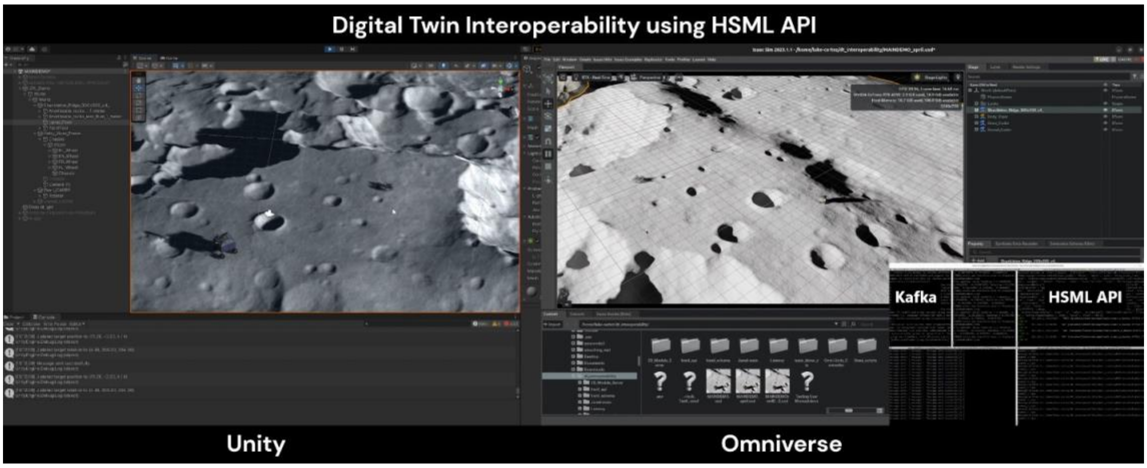
Before running the simulation, two rovers—Viper A and Cadre A—were registered under separate users (Jared Carrillo and Alicia Sanjurjo, respectively) to reflect a realistic multi-party scenario. Registration was handled via the HSML Web App, which connects to the HSML API that ensures all fields, including SWIDs and associated metadata, were stored properly in the registry. Kafka topics were automatically created for each agent and verified via the Kafka server logs.

To enable secure data exchange, credentials were issued through the Web App. These defined that both rovers were authorized to access the other's topic. The MySQL database was checked to confirm that the `allowed_did` fields were updated correctly, verifying that the registration mechanisms functioned as intended.

During the simulation, each producer rover authenticated itself using its private key before being allowed to publish data. The consumer counterpart on the other platform was given authorization by also using its own private key, which was validated against the access control list in the registry. Videos of the simulation were recorded (though not embedded here) to document the behavior of the digital twins under secure communication conditions. As shown in Figure 4.3, messages were successfully transmitted through the HSML API, and mirrored behavior between the digital twins was achieved, confirming functional interoperability under secure conditions.

Figure 4.3

Experiment 2 Cross-platform Simulation with HSML API Screenshot



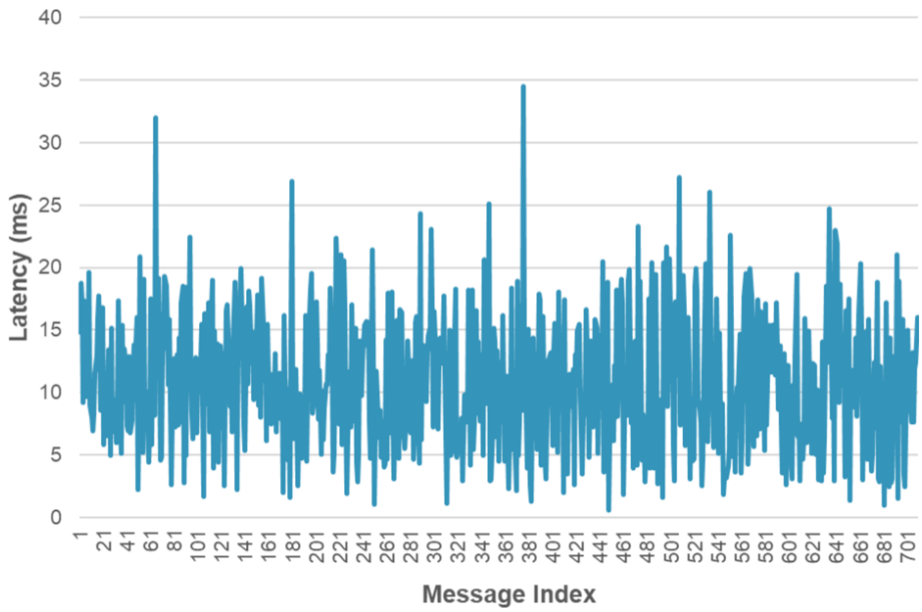
Note. Viper A (in Unity) and Cadre A (in Omniverse) are mirrored across platforms. The terminal window shows the HSML API facilitating secure communication with Kafka and connecting to the MySQL database for authentication and authorization.

The authentication and authorization processes were carried out successfully, and as in Experiment 1, the mirrored rovers moved smoothly, replicating the same displacements across platforms in what appeared to be real time. To verify this impression, the latency between the moment an HSML message was sent by the producer and the time it was received by the consumer, through the HSML API and Kafka, was measured. Figure 4.4 presents the results of this measurement. Latency remained consistently below 50 milliseconds, with an average of 10.83 ms, well below the threshold of human reaction time. This confirms that the synchronization achieved is indeed real-time.

The observed latency was even lower than in Experiment 1, likely due to the use of a fixed 10 ms interval between messages, which reduces the risk of overwhelming the Kafka server. It is also possible that the integration of the HSML API contributed to more efficient handling of messages and authentication. While the HSML messages used in this setup were relatively small, future space-based digital twin applications (such as communication between DT satellites) would involve significantly larger and more complex data exchanges. Ensuring minimal latency at this stage is therefore essential to support the scalability and responsiveness required in such contexts. These results indicate that the HSML API implementation of the Hyperspatial Transaction Protocol performed effectively under secure conditions, and supports its use in real-time, distributed digital twin systems.

Figure 4.4

Latency Between HSML Messages Sent Thorugh HSML API To Kafka



Note. Line plot showing latency measurements (in milliseconds) between HSML messages exchanged via authenticated producer and authorized consumer endpoints through the HSML API. The average latency was approximately 10.83 ms, with a maximum of 34.54 ms, a minimum of 0.60 ms, a median of 10.57 ms, and a standard deviation of 5.39 ms.

Table 4.3 summarizes the HSML fields that were parsed and applied in the verified cross-platform simulation. Similar to the previous experiment, spatial properties such as position and rotation were used to drive the behavior of the rovers, while metadata fields (e.g., creator, platform, description) were parsed for contextual purposes but not acted upon. In this case, an additional property (rotationWheels) was also parsed and used to animate the wheels of the rovers, demonstrating an increased complexity in the HSML data being applied in the simulation.

Table 4.3

Usage of HSML Fields in the Verified Cross-Platform Simulation

HSML Field	Usage in Simulation	Source
@context	Parsed only	HSML
@type	Parsed only	HSML
name	Parsed only	Schema.org
swid	Parsed only	HSML (DID:key)
url	Parsed only	Schema.org
creator	Parsed only	Mixed
dateCreated	Parsed only	Schema.org
dateModified	Parsed only	Schema.org
platform	Parsed only	HSML
spaceLocation	Parsed only	HSML
description	Parsed only	Schema.org
contentUrl	Parsed only	Schema.org
additionalProperty:scale	Parsed only	Mixed
position (Nested Field)		
xCoordinate	Applied	HSML
yCoordinate	Applied	HSML
zCoordinate	Applied	HSML
rotation (Nested Field)		
rx	Applied	HSML
ry	Applied	HSML
rz	Applied	HSML
w	Applied	HSML
rotationWheels (Nested Field)		
LF (Left Front)	Applied	HSML
LR (Left Rear)	Applied	HSML
RF (Right Front)	Applied	HSML
RR (Right Rear)	Applied	HSML

Note. Fields like position, rotation, and rotation of wheels were applied directly in the simulation to control digital twin behavior, while metadata such as SWID, creator, or description were parsed for context but did not affect runtime.

The verification system proved robust throughout the simulation, with all entities securely verified using their private keys and validated against the registry's access control list. No authentication failures or unauthorized access attempts were observed. HSTP enabled structured and secure interactions by enforcing identity verification and message authorization before data exchange. These mechanisms ensured that only trusted agents could participate in communication, supporting the integrity and confidentiality of the digital twin environment.

4.5. Summary & Insights

Throughout these experiments, key evaluation metrics such as latency, authentication success, and message consistency were monitored to assess the overall system performance.

Placeholder: Evaluation Metrics Summary TABLE mapping evaluation metrics to test scenarios

Together, the two experiments confirmed the feasibility and responsiveness of cross-platform interoperability using the Spatial Web's HSML schema and HSTP protocol. In the first experiment, structured HSML messages were exchanged directly through Kafka between Unity and Omniverse, enabling real-time synchronization of digital twin behavior with an average latency of 18.20 ms and no verification overhead. The second experiment introduced the full HSML API stack, incorporating registration, authentication, and authorization using DID:key credentials and HSTP-based communication. Despite these additional layers, the system still achieved a low average latency of 10.83 ms, demonstrating that secure, verified exchanges can occur without compromising performance. The smooth mirrored behavior observed in both simulations confirms that the HSML and HSTP protocols can support structured, trusted, and real-time communication between distributed digital twins across heterogeneous platforms.

One of the biggest implementation challenges was the need to adapt to each simulation platform by developing custom business logic (BL) plugins to connect to the HSML API, interface with Kafka, and correctly parse HSML messages. While this was manageable for the commercial platforms used in these experiments, which are well-documented and designed with API support in mind; it may be significantly more difficult for closed or proprietary digital twin systems. This highlights the need for future automation in the development of these platform-specific plugins. Leveraging Large Language Models (LLMs) or AI-assisted tools could help generate BL adapters more efficiently, reducing the required effort and improving scalability across heterogeneous environments.

While these results are promising, it is important to note that the tests were conducted using simplified HSML data and a prototype implementation. More complex payloads and real mission conditions could introduce additional challenges. Therefore, findings should be interpreted as an initial validation of the approach rather than a final benchmark. Scalability, larger data volumes, and more complex digital twin interactions should be tested in future work to confirm long-term viability. These insights inform the broader discussion and recommendations in the next chapter.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

This thesis explored how digital twin interoperability can be achieved for space exploration missions through the implementation of the Spatial Web standard. Motivated by the rising complexity and collaborative nature of lunar missions, the study aimed to address three main challenges identified in the literature: the lack of a standardized interoperability framework for digital twins, the fragmentation of simulation environments across platforms, and the lack for secure, trusted data exchange between stakeholders.

The problem this research addressed is central to the future of space operations: how to enable digital twins developed in different platforms to interact seamlessly and securely. Traditional digital twin architectures operate in isolation, using incompatible data formats, and communication protocols. This fragmentation limits collaboration between space agencies and private entities, especially as more missions start operating near the same valuable regions, like the lunar South Pole. To tackle this, the study proposed using the Spatial Web architecture as a unifying framework, specifically the Hyperspatial Modeling Language (HSML) and Hyperspatial Transaction Protocol (HSTP).

The work began by extending and implementing the HSML ontology to serve as a shared data language for digital twins. A structured JSON-based schema was created to describe entities and their relationships, leveraging Schema.org vocabulary where appropriate. This schema was used to generate compatible messages between simulation agents. In parallel, the HSML API was developed as a RESTful backend that connects simulation agents to Kafka topics through a secure authentication and authorization workflow. This system incorporated decentralized identity principles using the did:key method and supported topic-level access control using a credential-based verification logic. A MySQL database was used to store registered identities (SWIDs), credentials, and associated metadata.

To test the system, two main sets of experiments were conducted. The first experiment focused on cross-platform synchronization using direct Kafka messaging between

Unity, Unreal Engine, and Omniverse Isaac Sim. Agents in each platform published and subscribed to messages structured using the HSML schema. These tests validated the feasibility of interoperability under a shared ontology, showing that real-time data exchange was possible with average latencies under 50 milliseconds.

The second experiment introduced the full verification framework via the HSML API, simulating a more realistic, secure multi-stakeholder scenario. Agents were required to authenticate using private key signatures and were granted access to topics only if their credentials matched predefined authorization rules. These experiments confirmed that the authentication and authorization mechanisms based on did:key and HSTP did not compromise real-time performance and operated consistently under simulation loads.

Overall, the results show that the Spatial Web framework enables real-time, cross-platform interoperability while ensuring secure communication between digital twins. The system maintained high performance and low latency despite added security layers, proving that decentralized architectures are reliable and efficient. This has important implications for future lunar operations, where agents from different actors must coordinate and share mission-critical data. More broadly, the work serves as a proof of concept for the Spatial Web as a foundational architecture for interoperable, secure digital twin systems in space, with HSML providing the semantic layer for multi-agent coordination and HSTP enabling trusted and secure interactions.

However, this study also recognizes its limitations. The experimental framework was limited to three simulation platforms that already support interoperability features and focused on a narrow set of data types. The key management system used a centralized SQL database rather than a fully decentralized ledger, and the communication topology was limited to a local network setup. As a result, the findings may not fully generalize to more constrained or proprietary environments, such as those operating on deep space networks or under strict military governance.

Nonetheless, the prototype and its accompanying results offer a valuable starting point for future work. The HSML schema, HSML API, and verification logic form a working stack that demonstrates the Spatial Web's potential to unify digital twin development across domains and ultimately to support the next generation of collaborative, secure, and intelligent space missions.

5.2. Recommendations

Based on the findings and limitations of this work, the following five recommendations are proposed to guide future research and development:

1. **Generalize Across More Platforms and Frameworks.** To broaden applicability, the system should be tested with additional platforms beyond Unity, Unreal, and Omniverse; including space-specific tools like MATLAB, ROS-native systems, and proprietary mission software. This will help ensure the HSML schema and API are adaptable to diverse operational contexts.

2. **Expand the HSML Schema with Domain-Specific Extensions.** While the current HSML implementation covers core concepts, future work should incorporate aerospace and robotics-specific vocabularies (e.g., NASA Open MCT, ECSS standards) to improve semantic coverage and facilitate integration with established mission architectures.

3. **Use AI to Automate Plugin Development.** LLMs could be used to auto-generate adapters and plugins for simulation platforms. This would significantly reduce development time and simplify integration of new agents, enabling faster and more scalable deployment.

4. **Improve Security, Access Control, and Governance.** Enhance the verification layer with session expiration, access revocation, and logging. Introduce role-based access control for topic-level permissions and embed copyright tracking to ensure proper attribution of shared models and data.

5. **Move Toward Decentralization and Edge Readiness.** Replace the centralized key registry with decentralized systems like blockchain or verifiable credentials. Future testing should also target low-bandwidth, distributed environments such as those found in lunar or deep space operations.

These steps will help evolve the current prototype into a robust, secure, and scalable solution for real-world digital twin interoperability in space exploration.

BIBLIOGRAPHY

Acharya, S., Khan, A. A. and Päivärinta, T. (2024), 'Interoperability levels and challenges of digital twins in cyber–physical systems', *Journal of Industrial Information Integration* **42**, 100714.
URL: *https://www.sciencedirect.com/science/article/pii/S2452414X24001572*

Barnett, A. (2019), 'Moon's South Pole in NASA's Landing Sites', *NASA* . April 16, 2019.
URL: *https://solarsystem.nasa.gov/news/907/moons-south-pole-in-nasas-landing-sites/*

Bilka, M., Wright, R. and Wysack, J. (2024), Guiding Lunar Growth: Architectural Solutions for Space Traffic Management, *in* S. Ryan, ed., 'Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference', p. 72.

Botín-Sanabria, D. M., Mihaita, A.-S., Peimbert-García, R. E., Ramírez-Moreno, M. A., Ramírez-Mendoza, R. A. and Lozoya-Santos, J. d. J. (2022), 'Digital Twin Technology Challenges and Applications: A Comprehensive Review, journal = Remote Sensing', **14**(6).
URL: *https://www.mdpi.com/2072-4292/14/6/1335*

Brennan, P. (2021), 'Twin of NASA's Perseverance Mars Rover Begins Terrain Tests', *NASA Jet Propulsion Laboratory* . November 17, 2021.
URL: *https://www.nasa.gov/centers-and-facilities/jpl/twin-of-nasas-perseverance-mars-rover-begins-terrain-tests/*

Budiardjo, A. and Migliori, D. (2021), 'Digital Twin System Interoperability Framework', *Digital Twin Consortium* .
URL: *https://www.digitaltwinconsortium.org/pdf/Digital-Twin-System-Interoperability-Framework-12072021.pdf*

Böning, S. (2021), 'The Current State of Research and Technology of Digitalization in the Space Industry', *DLR* . DLR-IB-DW-JE-2021-55.
URL: *https://elib.dlr.de/141454/1/B%C3%B6ning_TheCurrent.pdf*

Carlos, J. (2021), ‘10 Ways Digital Twins Are Propelling Satellite Networks to the Future’, *Harvard. Digital Innovation and Transformation* . March 23, 2021.
URL: *https://d3.harvard.edu/platform-digit/submission/spacex-enabling-space-exploration-through-data-and-analytics/*

Carrillo, J., Chakraborty, S., Mridha, R., Janvisloo, N. E., Sadler, E., Li, B., Ho, N., de Fourtou, G. B., Lu, T., Chow, E., Petersen, C., Eslami, A., Muzzi, A., Percivall, G. and Hynes, J. (2025), ‘Enabling Interoperable Digital Twins for Collaborative Lunar Exploration’, *Presented at the IEEE Aerospace Conference* .

David, I., Shao, G., Gomes, C., Tilbury, D. and Zarkout, B. (2025), Interoperability of Digital Twins: Challenges, Success Factors, and Future Research Directions, *in* T. Margaria and B. Steffen, eds, ‘Leveraging Applications of Formal Methods, Verification and Validation. Application Areas’, Springer Nature Switzerland, Cham, pp. 27–46.

Dinner, J. (2025), ‘SpaceX will launch 2 private lunar landers this week, kicking off busy year for moon missions’, *SPACE.com* . January 12, 2025.
URL: *https://www.space.com/space-exploration/launches-spacecraft/spacex-will-launch-2-private-lunar-landers-this-week-kicking-off-busy-year-for-moon-missions*

Ehrenfreund, P. and Christensen, C. (2024), ‘The path forward for sustainable space exploration’, *World Economic Forum* . July 8, 2024.
URL: *https://www.weforum.org/stories/2024/07/sustainable-space-exploration-path-forward/*

Elewaily, D. I., Ali, H. A., Saleh, A. I. and Abdelsalam, M. M. (2024), ‘Delay/Disruption-Tolerant Networking-based the Integrated Deep-Space Relay Network: State-of-the-Art’, *Ad Hoc Networks* **152**, 103307.
URL: *https://www.sciencedirect.com/science/article/pii/S1570870523002275*

Epic Games (2021), ‘Unreal Engine: Real-Time 3D Creation Platform’, *Unreal Engine* . Epic Games Documentatio. Accessed on: March 23, 2025.
URL: *https://www.unrealengine.com/*

ESA (2024*a*), ‘ESA’s Moonlight programme: Pioneering the path for lunar exploration’. October 15, 2024.

URL: *https://www.esa.int/Applications/Connectivity_and_Secure_Communications/ESA_s_Moonlight_programme_Pioneering_the_path_for_lunar_exploration*

ESA (2024*b*), 'Lunar Economy Applications', *European Space Agency (ESA)* . Accessed on: December 28, 2024.
URL: *https://business.esa.int/funding/open-competition/lunar-economy-applicationsreturntomoon*

Freedman, A. (2024), '10 Ways Digital Twins Are Propelling Satellite Networks to the Future', *Keysight* . May 21, 2024.
URL: *https://www.keysight.com/blogs/en/inds/ai/10-ways-digital-twins-are-propelling-satellite-networks-to-the-future*

Guo, J. and Lv, Z. (2022), 'Application of Digital Twins in multiple fields', *Multimedia Tools and Applications* **81**(19), 26941–26967.
URL: *https://doi.org/10.1007/s11042-022-12536-5*

IEEE Computer Society (2024), 'Draft Standard for Spatial Web Protocol, Architecture, and Governance (P2874/D3.1)'.

Jones, A. (2024), 'India to target moon's south pole with sample return mission', *SpaceNews* . October 23, 2024.
URL: *https://spacenews.com/india-to-target-moons-south-pole-with-sample-return-mission/*

Kawahara, S. and Hanafusa, R. (2024), 'U.S. and Japan spearhead new era of private moon landings', *Nikkei Asia* . Accessed on: December 28, 2024.
URL: *https://asia.nikkei.com/Business/Aerospace-Defense-Industries/U.S.-and-Japan-spearhead-new-era-of-private-moon-landings*

Klar, R., Arvidsson, N. and Angelakis, V. (2024), 'Digital Twins' Maturity: The Need for Interoperability', *IEEE Systems Journal* **18**(1), 713–724.

Kuruvatti, N. P., Habibi, M. A., Partani, S., Han, B., Fellan, A. and Schotten, H. D. (2022), 'Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey', *IEEE Access* **10**, 112158–112186.

Lima Baima, R., Chovet, L., Hartwich, E., Bera, A., Sedlmeir, J., Fridgen, G. and Olivares-Mendez, M. A. (2024), Trustful Coopetitive Infrastructures for the New Space Exploration Era, *in* 'Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing', SAC '24, ACM, p. 612–621.
URL: *http://dx.doi.org/10.1145/3605098.3635887*

Liu, W., Wu, M., Wan, G. and Xu, M. (2024), 'Digital Twin of Space Environment: Development, Challenges, Applications, and Future Outlook', *Remote Sensing* **16**(16).
URL: *https://www.mdpi.com/2072-4292/16/16/3023*

Lloyd, V., Johnson, A. and Hambleton, K. (2022), 'NASA Identifies Candidate Regions for Landing Next Americans on Moon', *NASA* . August 19, 2022.
URL: *https://www.nasa.gov/news-release/nasa-identifies-candidate-regions-for-landing-next-americans-on-moon/*

Manning, C. G. (2023), 'LunaNet Interoperability Specification'. February 8, 2023.
URL: *https://www.nasa.gov/directorates/somd/space-communications-navigation-program/lunanet-interoperability-specification/*

NASA (2018), 'Global Exploration Roadmap'. Accessed on: March 3, 2025.
URL: *https://www.nasa.gov/wp-content/uploads/2015/01/ger2018smallmobile.pd f*

NASA (2020), NASA's Plan for Sustained Lunar Exploration and Development, Technical report, NSPC.
URL: *https://www.nasa.gov/wp-content/uploads/2020/08/a_sustained_lunar _presence_nspc_report4220final.pdf*

NASA (2023), 'Using Space-Based Resources for Deep Space Exploration'. Accessed on: March 3, 2025.
URL: *https://www.nasa.gov/overview-in-situ-resource-utilization/*

NASA (2025), 'The Multi-Mission Algorithm and Analysis Platform (MAAP)'. Accessed on: March 8, 2025.
URL: *https://www.earthdata.nasa.gov/about/maap*

NASA JPL (2025), 'Mars 2020 Perseverance Rover', *NASA Jet Propulsion Laboratory* .

Accessed on: February 17, 2025.
URL: *https://www.jpl.nasa.gov/missions/mars-2020-perseverance-rover/*

NSR (2023), *Lunar Markets 3rd edition*, Northern Sky Research.

NVIDIA (2021), 'NVIDIA Omniverse: A Platform for Real-Time Simulation and Collaboration', *NVIDIA* . Accessed on: March 23, 2025.
URL: *https://developer.nvidia.com/omniverse*

NVIDIA (2025), 'Building Interoperable 3D Workflows With OpenUSD', *NVIDIA Omniverse* . Accessed on: March 2, 2025.
URL: *https://www.nvidia.com/en-us/omniverse/usd/*

René, G. and Mapes, D. (2019), *The Spatial Web: How Web 3.0 Will Connect Humans, Machines and AI to Transform the World*, Gabriel René.

Unity (2021), 'Unity: Real-Time 3D Development Platform', *Unity* . Unity Technologies Documentation. Accessed on: March 23, 2025.
URL: *https://unity.com*

U.S. Department of State (2025), 'Artemis Accords'. Accessed on: February 17, 2025.
URL: *https://www.state.gov/bureau-of-oceans-and-international-environmental-and-scientific-affairs/artemis-accords*

Wörner, J. (2016), 'Moon Village: A vision for global cooperation and Space 4.0', *ESA* . Ministerial Council 2016. ESA Director General.
URL: *https://www.esa.int/About_Us/Ministerial_Council_2016/Moon_Village*