NIST Internal Report
NIST IR 8356

Security and Trust Considerations for Digital Twin Technology

Jeffrey Voas
Peter Mell
Phillip Laplante
Vartan Piroumian

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8356

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Internal Report
NIST IR 8356

Security and Trust Considerations for Digital Twin Technology

Jeffrey Voas
Peter Mell
Phillip Laplante
Computer Security Division
Information Technology Laboratory

Vartan Piroumian
Independent Consultant

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8356

February 2025

U.S. Department of Commerce
Jeremy Pelter, Acting Secretary of Commerce

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Certain commercial equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on 2025-01-31

How to Cite this NIST Technical Series Publication:
Voas J, Mell P, Laplante P, Piroumian V (2025) Security and Trust Considerations for Digital Twin Technology. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Internal Report (IR) NIST IR 8356. https://doi.org/10.6028/NIST.IR.8356

Author ORCID iDs
Jeffrey Voas: 0000-0003-1139-3690
Peter Mell: 0000-0003-2938-897X
Phillip Laplante: 0000-0002-0415-271X
Vartan Piroumian: 0000-0000-0000-0000

Contact Information
nistir-8356-comments@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Additional Information
Additional information about this publication is available at https://csrc.nist.gov/pubs/ir/8356/final, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

Digital twin technology enables the creation of electronic representations of real-world entities and the ability to view the states and transitions between states of these entities. This report discusses the concept and purpose of digital twin technology and describes its characteristics, features, functions, and expected operational uses. This report also discusses both traditional and novel cybersecurity challenges presented by digital twin technology as well as trust considerations in the context of existing NIST guidance and documents.

Keywords

computer cybersecurity; control; digital twin technology; instrumentation; real-time command; real-time monitoring; simulation; standards; testing; trust; use case scenarios.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Audience

This publication is intended for anyone who wants to understand the underlying technology and envisioned capabilities of digital twin technology. It is particularly applicable to Standards Developing Organizations (SDOs) and implementers of digital twin technology.

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, no such patent claims have been identified to ITL.

No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Table of Contents

1. Introduction

A digital twin (DT) is an electronic representation of a real-world entity; providing the capability to evaluate this entity. A digital twin can emulate both physical things (e.g., buildings, electronics, living things), and non-physical things (e.g., processes, conceptual models). As with many new information technologies, digital twin technology employs existing foundational technologies and may reflect existing capabilities. It covers what currently exists in modeling and simulation but then casts a broader vision for future capabilities. The full benefits of digital twin technology will require interoperable definitions, tools, and standards as well as early consideration of digital twin cybersecurity and trust. This situation is especially true for nascent standards efforts that seek to define and structure the technology.

This report introduces the concept of a digital twin, describes the underlying technologies, and expands on its current capabilities by discussing key components, functions, and cybersecurity and trust considerations. It is not intended to define "digital twin" – this activity should be undertaken by SDOs. Any definition of "digital twin," however, should address a set of technical considerations, which are offered herein. These technical considerations can also be used to test any definition that is created by any SDO.

This report is organized as follows:

- Section 2 defines digital twin technology.
- Section 3 describes the motivations for using digital twin technology, including advantages and supportive technologies.
- Section 4 discusses typical operations performed on digital twins.
- Section 5 describes technical usage scenarios for digital twins.
- Section 6 provides example industry applications of digital twin technology.
- Section 7 explores cybersecurity considerations in the context of existing NIST guidance by identifying and exploring traditional cybersecurity needs, novel cybersecurity challenges, and approaches that apply to digital twin technology.
- Section 8 discusses trust issues that can prevent a digital twin from providing the desired operational functionality with an acceptable level of quality in the context of existing NIST guidance.
- Section 9 offers concluding thoughts.

2. Definitions of Digital Twins

There are several existing definitions for digital twins. Some have been created by researchers, standards committees, consortia, and industry. Others are implicitly suggested by commercial enterprises that make statements about how their software applications are "digital twin-compliant." Despite these definitions there is no agreed-upon definition for or consensus on the full potential of digital twins [1].

The Digital Twin Consortium (DTC) offers the following description for digital twins:

> A digital twin is a virtual representation of real-world entities and processes synchronized at a specified frequency and fidelity.
>
> - Digital twin systems transform business by accelerating holistic understanding, optimal decision-making, and effective action.
> - Digital twins use real-time and historical data to represent the past and present and simulate predicted futures.
> - Digital twins are motivated by outcomes, tailored to use cases, powered by integration, built on data, guided by domain knowledge, and implemented in IT/OT systems [1].

A simplified definition for digital twins[1] could be:

> *A digital twin is the virtual (i.e., digital) representation of a physical or perceived real-world entity, concept, or notion.*

The use of the word "virtual" is appropriate because a digital twin is something that has the effect but not the actual form of what is specified. DT software implementations present a human user with an object's visual graphic representation, either static or dynamic, via the object's DT.

A related important term is that of a *digital twin definition*:

> *A digital twin definition is a machine-readable specification that describes features that may be modeled for a particular type of real-world entity.*

Thus, a digital twin definition refers to a particular type of entity rather than the specific entity itself. It defines the features of an entity type that can be statically and dynamically modeled, how those features will be digitally encoded and represented, and how they will persist in a digital computer environment. Computer software applications will read digital twin definitions to create digital twin instances (or simply, *digital twin*), which are instantiations of real-world objects that model the state of represented objects.

While many of these real-world entities have physical forms (e.g., an aircraft engine, an oil derrick, a valve in an oil pipeline pumping station), digital twins can also represent something abstract. The DTC definition contains the word "processes," which is an abstract notion, and the simplified definition discusses entities that may be perceived or conceived without having a

[1] Section 3 discusses the differences between digital twin technology and traditional modeling and simulation, and Sec. 9 discusses the importance of synchronization between a digital twin and a real-world entity.

physical form. For example, a process in a computer operating system is real, even if it does not have a concrete or physical shape. It is a conglomeration of multiple intangible things, such as electrical signals, the states of registers that contain voltage and current levels, and the electrical state of memory. Whether it is viewed statically or dynamically, a computer program is real, and one can clearly observe the effects that it has on other objects.

A business process is another example of an abstract concept that is real but has no material form. In fact, some software defines digital twins as representations of business processes. Digital twins could even describe the steps in a manufacturing process or simulate aspects of the dynamic execution of specific processes in a factory or chemical plant, such as oil refining or the production of nuclear fuel. In short, a digital twin can represent anything that a human can conceive or perceive, whether physical or not.

3. Motivation and Vision

Elements of digital twin technology have long existed in computers and software that represent entities and simulate dynamic behavior. Now, the maturation of numerous underlying technologies is making it possible to broadly apply simulation and modeling in the form of digital representations and make the technology accessible to a much wider user base. The Internet of Things (IoT) has led to the emergence of small, low-cost, battery-powered sensors that connect over a network and enable massive sensor deployment to a wide variety of objects (e.g., modern buildings may have thousands of sensors). These sensors then provide information that can feed and maintain complex models of those objects. The advances in powerful but low-cost processing and storage enable us to maintain, view, and manipulate these digital replicas without having to use special-purpose or expensive hardware. The recent advances in virtual reality (VR) and augmented reality (AR) have enabled inexpensive visualization of digital twins.

Digital twin technology is also an advancement over existing simulation and modeling because it allows for the real-time monitoring of entities while dynamically updating their digital twins. There is also a trend to remote control physical entities by manipulating dynamic models (i.e., digital twins) as opposed to directly manipulating the objects themselves. Such control is more indirect and abstracts away details that humans may not be able to manage.

Standards development will likely impact whether digital twin technology becomes widely used. Most IoT systems, simulation and modeling software, and VR and AR systems currently exist in stovepipe proprietary systems and integrating them requires significant work. Much of the work in emerging digital twin technology is in the creation of protocols and standards to enable plug and play integration. The goal is to be able to load any digital twin computer file into a digital twin system and have it function regardless of what is being modeled.

3.1. Advantages of Digital Twin Technology

A platform or mechanism that supports the creation of digital models of real-world objects is advantageous for several reasons. For example, one can study the object via its model prior to building the real-world version, study the object as it progresses through its life cycle, and conceivably control the object through the model to prevent undesirable outcomes for the object, thus reducing certain types of risk.

This advantage increases when modeling multiple objects that need to work together, even if the objects are maintained by different organizations. If cooperating entities can share digital twin definitions, then they can more easily model and digitally simulate object interactions prior to the realization of the output product. However, the internal definitions and representations of the objects being modeled by each software application tend to be highly proprietary. The digital artifacts created by today's applications are not easily shared, and the applications are therefore not interoperable.

3.2. Expectation of Standards

The adoption of and adherence to standards may ensure interoperability, compatibility, safety, and cybersecurity. Moreover, the assurance that software and hardware systems, tools, and applications adhere to and properly implement standards engenders credibility and trust [3]. Efforts are underway to develop digital twin specific standards that will be utilized in addition to the existing various information and communication technology standards.

Digital twin technology will be built upon existing computing system stacks, platform architectures, programming platforms, systems, libraries, application programmer interfaces (APIs), and infrastructure. For example, OpenGL is a possible 3D graphics standard used to render the visual representation component of digital twins. While existing cybersecurity encryption standards will be leveraged, multiple cooperating (or competing) standards that are specific to digital twin technology will be needed. A single standard may not adequately address all needs, and standards harmonization or blending may be appropriate [4]. Tool vendors, software and hardware application vendors, and users can comply with standards by ensuring that they only use vetted elements, which should also lead to the interoperability of tools and applications.

One approach to establishing standards for digital twin technology is to focus on standard mechanisms for exchanging information, such as the representation of real-world objects. Algorithms that simulate the dynamic behavior of objects could remain proprietary to protect intellectual property, but the description of the object whose behavior is being simulated could be standardized and open in order to be exchanged between applications, domains, industries, and vertical markets. In the simulation and modeling arena, there are many sophisticated software tools and applications that support 2D and 3D modeling and engineering analysis. Each of these applications uses proprietary internal models to represent the objects being modeled (i.e., those created by the user). They also use proprietary and often closely guarded algorithms that represent the functional capabilities for modeling and simulation.

Potential standards would need to cover each involved business domain. Everyone who employs digital twin technology in specific business domains would need their own unique standards and standards-based products that adhere to a common set of business processes and use cases where interoperability can be achieved. It is insufficient for standards to merely enable interoperability in purely technological domains.

3.3. Supportive Technologies

Two of the supportive technologies that support the recent interest in digital twin technology are VR/AR and IoT. One expectation for digital twin technology is to leverage VR and AR to create enhanced user interfaces and user experiences for human beings to comprehend the modeling, simulation, monitoring, and command and control of complex entities. Humans rely heavily on visual sensory input, and VR and AR promise to present models of real-world entities through a medium that is amenable to human consumption and comprehension.

IoT has been referenced in digital twin discussions and literature thanks to recent advances in sensors and their ongoing and dramatic proliferation in various operations. These sensors are

typically network-connected and drive the ability of digital twins to model real-world objects in ways that were not possible until recently. Additionally, IoT devices are often used to create an information fabric or "network" that consists of the observed entities, the sensors that observe and gather information, the connectivity elements, the processing components (i.e., backend compute servers), and the components that use the processed IoT data. With these new sensors being deployed on an IoT fabric, digital twins can represent and dynamically maintain the representation of an instance of an instrumented object [5].

Thus, the application of digital twins goes beyond simply modeling a class of real-world entities. It can also be used to represent and track a specific object, maintain the real-time status, and present a dynamically updated view to a user. With an accurate understanding of the state of an object, a digital twin may also be manipulated by a user to control the actual object, meaning that DT technology may advance beyond traditional modeling and simulation software to encompass command-and-control. For example, operators could remotely command surface rail or subway trains from an operations center.

Depending on the object, a system that monitors the object's state may need to understand the state's rates of change. For example, a system that monitors the rate of velocity change may need to detect when the boundaries of safe operation are surpassed. For this, artificial intelligence (AI) and machine learning (ML) could potentially outperform traditional computing methods as well as humans who monitor conditions and make predictions. Scientists and engineers could create models of real-world conditions and employ them to train AI systems to recognize those conditions. Such applications for simulation, modeling, and monitoring are major motivation for advancing digital twin technology.

4. Operations on Digital Twins

This section discusses several lower-level operations that are performed on digital twins:

- Digital twin definitions (i.e., descriptions of object types) and the creation of digital twin instances
- Manipulation and modification of digital twin definitions and instances
- Exchange via electronic communications of digital twin definitions and instances

It is envisioned that there will be many digital twin definitions that describe many kinds of entities. This section also discusses operations on digital twins,[2] which are specific instances created from digital twin definitions.

4.1. Digital Twins Definitions and the Creation of Digital Twin Instances

A digital twin *definition* is a *formal description* of the real-world entity that the digital twin represents. For the purposes of this report, think of a formal description as a technical definition of a particular category or class of real-world objects.

The starting point for all activity involving digital twin technology is to create or find a digital twin definition that represents the type of real-world object that is to be represented virtually. Computer software applications are then used to create an instance for that definition to hold and/or maintain the state of the represented object. These digital twin electronic instances can represent both static and dynamic models of the real-world entities that correspond to their respective digital twin definitions.

The specific digital twin definition created for some object types will dictate the precise makeup of the artifacts that are instantiated from the definition. For example, a digital twin definition is not required to include a dynamic view of its real-world counterpart. Rather, it could comprise only a static view of the object. Thus, not all digital twin definitions will necessarily contain all possible declarations or definition constructs defined in some future standard. Similarly, not all hypertext markup language (HTML) files utilize all tags defined by the HTML standard [6]. If a particular digital twin definition only supports representing a static model of an entity, the related instances would contain no dynamic information, such as how to render animation, video, or dynamic graphics. Consider, for example, a VR presentation of a naval vessel. A static view could represent the internal elements of the ship seen through VR as if a person were literally walking through the vessel. VR technology would be more amenable to this application than a 3D PDF view. The latter would comprise detailed engineering drawings tantamount to an architect's blueprint drawings. However, it would be difficult to present the equivalent of what a person would see walking through the interior of the ship.

A digital twin definition should create a model of the object it represents, not just a particular view. The model could then be used to present the desired viewpoint of the real-world entity. A definition can contain as much or as little information about its real-world counterpart as its

[2] A reference to a "digital twin" without other qualifiers refers to the digital twin instance that represents the entity.

author desires, which may limit the types of views that can be created. Digital twin definition authors decide the breadth, scope, degree of granularity, and detail.

A practical consideration is the process of digital twin definition authoring and encoding. While it may be possible to author definitions using a text editor, this practice could become supplanted by more advanced tools. This is similar to how not many people hand-code HTML or XML anymore [7]. Moreover, the complexity of digital twin definitions could entirely preclude the ability to craft definitions by hand. Many industries use sophisticated software applications to create digital artifacts that represent what they plan to build. Some of these software applications support the "export" of their artifacts in standard file formats and encodings, such as the 3D PDF standard [8]. However, the majority of these applications use their own proprietary file formats and encodings to define, capture, and persist the models, drawings, and various artifacts that they create.

Like existing commercial applications, any future digital twin standard should include a *language* for describing and defining a digital twin, including *formal grammar*, *syntax*, and *semantics*. It would have to be comprehensive enough to support the definition of artifacts to represent any arbitrary real-world entity that a digital twin can represent [5]. Most likely, a standard would accommodate the creation of static and dynamic 2D, 3D, VR, and AR models for visual presentation to human users. It may also accommodate the creation, manipulation, and persistence of presentation forms that are intended for machine rather than human consumption.

4.2. Manipulation and Modification of Digital Twin Definitions and Instances

Digital twin definitions will likely be available from libraries to enable reuse by software applications. Software can execute a digital twin definition to create a specific instantiation linked to a real-world object. Software can both read and modify digital twin definitions and their instantiations, which would allow for modeling, simulation, monitoring, and other applications.

Industries may develop digital twins editing tools and integrated development environments (IDE) with the capabilities to test static and dynamic operations. Such tools would be able to read the digital twin definition language and its file formats, encodings, grammar, syntax, and semantics in order to support its review or modification by a human user.

4.3. Exchange of Digital Twin Definitions and Instances

Digital twin definitions and object instances are simply computer files or collections of files that are available for reading, writing, execution, and general manipulation. They can be sent to recipients for instantiation, similar to how 3D printer files are shared to enable multiple people to create the same object. The power in sharing these files is that they follow a standard. Such standards will need to be developed for digital twin technology to harness this advantage as current systems use proprietary formats.

5. Usage Scenarios for Digital Twins

This section describes scenarios that are likely to represent the main general usage categories of digital twin technology in practice. The broad categories listed below represent the major functions of digital twins (i.e., the ways in which digital twins interact with the real-world objects they represent):

- Viewing static models
- Executing and viewing dynamic simulation models
- Real-time monitoring of real-world entities
- Real-time command and control of real-world entities

There are effectively a limitless number of applications within any one of the above categories. For example, a software application that monitors an operating automobile engine could use digital twins to represent one or more subsystems, which would fall into the "real-time monitoring of real-world entities" category above. However, the purpose of the application would be to monitor, detect, and report any faults detected in the engine operation.

5.1. Viewing Static Models of Digital Twins

A static digital twin does not describe its corresponding real-world entity's behavior [11]. This type of view presents a non-changing model of a real-world twin, regardless of the nature of the real-world entity or how that entity may change over time. The real-world entity may not even exist yet as it would occur during the initial design of an object. Such a model would only be suitable for examining the nature of an object at a point in time. For example, a computer numerical control (CNC) milling machine uses a static 3D model to describe the object to be milled. In the aerospace industry, designers or modelers first create what they call a *solid model* of the component or entity that they are designing. These comprise static 2D or 3D views of a component, such as an aircraft wing or empennage (i.e., tail assembly). A building architect typically creates drawings of a house to be built with various 2D views, such as a site plan, floor plan, and elevation plans. While architects could adopt the practice of creating 3D views, they would be more difficult to read and less useful to building contractors.

The creator of the corresponding digital twin definition will define a model that supports certain static presentations, and the instantiation will provide a subset of those available from the model. For example, the digital twin definition may allow for 3D modeling information, but the instantiation may only be provided with 2D information, limiting its presentations to 2D.

5.2. Executing and Viewing Dynamic Models of Digital Twins

A human user may execute a digital twin instance to model an object's changes over time and view the dynamic changes to the object. The object may or may not yet exist. A dynamic model presents a *simulation* of the *operation* or *dynamic behavior* of a real-world entity or object, which describes how an object changes as measured via one or more metrics that represent one or more aspects or characteristics of the object [12]. For example, visual updates to

graphics can show how the track of a roller coaster bends as a function of applied force from wind loading, the dynamic response of a building during an earthquake, or how an aircraft's wing flutters or bends under changing loads in flight. An engineer who wants to understand how a milled block changes in malleability, ductility, or tensile strength over time as it is heated at some rate would need a dynamic model that includes a knowledge of thermodynamics, mechanical engineering, and materials engineering. That is, a dynamic model shows more than just the static dimensions, shape, material, or density of an object.

A visual presentation can use many methods to produce information in a format that is comprehensible to humans. The choice is at the discretion of the model's author. For example, a visual presentation of a wing in flight could include the use of various colors to show the variability of stress along the wing's surface area with the application of force. However, visual graphical user interfaces may not be required for applications that perform simulations. The results of a simulation could be a table of numbers displayed on the user's console or written to a file. The numbers could represent the change in some aspect of the object according to a suitable metric, which may not be user-friendly but is a presentation of the model, nonetheless.

The various types of presentation of dynamic simulations that digital twin technology may support can be categorized as:

- Real-time or near real-time presentation of a simulation during the simulation run (i.e., execution of the dynamic simulation model)
- Local playback of a previously recorded simulation run
- Streaming of a dynamic simulation run
- Download and local playback of a previously recorded simulation run

The *imperative* and *declarative* programming paradigms are both important for the kinds of software applications that will use digital twin technology. Think of the MIT X Window System,[3] which represents the *imperative programming paradigm* to display graphics [15]. Applications make calls to X library routines and those of the graphics toolkits that are built on top of the venerable `Xlib` and `Xt` X Window System libraries. Those calls draw the graphics, and the X display server renders the visual graphics on the graphics display [16].

In the *declarative programming paradigm*, the information encoded indicates *what* is to be displayed rather than *how* to do it [17]. There are no imperative calls to execute the steps to display the graphics. HTML is an example of a declarative programming paradigm. An HTML file represents directives of *what* to display, not *how* to display it. Therefore, there are no imperative commands to display the content like the programmatic calls to routines in the X Window System libraries.

The streamed or downloaded *content* that represents digital twin-based dynamic simulations could consist of pre-captured video, such as an MPEG-encoded video. In that case, the digital twin application software probably creates the standard video from the simulation run. Alternatively, digital twin application software could create declarative-style content to be

[3] See https://en.wikipedia.org/wiki/X_Window_System.

parsed, comprehended, and manipulated for display by the client receiving the content. This might look something like HTML from an architecture viewpoint. The content would consist of a combination of declarative constructs, including some that point to other content such as pre-recorded video or even executable code that is in the imperative style. Web pages today contain directives to download and run code, such as JavaScript programs.

5.3. Real-Time Monitoring of Real-World Entities

Monitoring the state or condition of real objects is fundamentally different from simulation. Monitoring collects the information of actual real-world entities, typically in real-time or near real-time, to create a dynamically updated digital replica and to enable the interoperability of different tools to view and manipulate the digital replica through standards [19].

With monitoring, sensors on an object send data to a digital twin instance that maintains a model of that object. The system can be local or remote, use wired or wireless connections, and employ any number of transmission media technologies and protocols (e.g., satellites have sensors that perform ground imaging). The data gathered by the sensors can be transmitted to applications that present 3D graphics, VR or AR experiences to users, or data intended for consumption by another computer program. Users can then view and inspect the model and even run virtual tests on the represented object. For example, a VR capability could enable an airline mechanic to view an engine while it is in operation during flight.

5.4. Real-Time Command and Control of Real-World Entities

Real-time links to real-world objects allow for command and control via a digital twin instance. Command and control systems require bidirectional links, and the transmission of information that must be encoded in the digital twin definition upon which the instance is based. The instance can provide a model of an object that is continuously updated with information from sensors and present that model to users (i.e., humans and other computer systems). The users can then provide high-level modifications to the object model, which the instance transforms into specific detailed commands to achieve the desired final state. AI and ML may be needed to achieve this, such as a self-driving car that receives a destination from a human but handles the actual navigation and steering commands.

Standards for digital twins may enable interoperability between tools and formats to enable command and control. For example, applications may not need to use proprietary schemes for defining and controlling objects and representing models, views, and other aspects, such as semantics, syntax, file formats, and tools.

6. Highlighted Use Cases

The following example applications are taken from industries that are already exploring and using digital twin technology. They will provide context for the potential cybersecurity vulnerabilities in digital twin technology that are discussed in Sec. 7.

- **Unmanned aerial vehicles (drones):** Unmanned aerial vehicles (UAVs) or drones are used in environmental monitoring. UAVs come in all sizes, shapes, configurations, and levels of sophistication. More advanced drones can operate autonomously or under the control of a human in a command-and-control center that is far from the UAV's physical location. Remote operators of UAVs have user interfaces that provide the real-time information about the UAV's state and condition.
- **Ocean-going vessels:** Digital twins can be used to construct 2D and 3D static views or VR/AR views of ocean vessels. VR/AR views would enable architects, designers, engineers, and maintenance crews to see the vessel as if they were physically walking through it. During operations, digital twin technology would enable operators and crew members to monitor every aspect of the ship, possibly precluding the need for certain physical monitoring and inspections.
- **Oil derricks and ocean-drilling platforms:** Oil derricks may drill for oil in inhospitable and potentially treacherous environments at great depths. Systems built around digital twins enable designers, engineers, and operators to form models and visual representations of oil rigs, drill rigs, and drill bit heads deep in the ocean.
- **Robotic surgery:** There are surgical robots that perform various kinds of surgery. Some require an actual human surgeon to control the robot, while others only require a human surgeon to monitor the robot's automatic execution of the surgery. Digital twin technology could be used in pre-surgery planning and to foster surgical tool interoperability. For example, one company's surgical robot could interoperate with another company's VR system that specializes in the representation of human organs.

7. Cybersecurity Considerations

The integration of known components combined with a certain maturation in the industry has created novel characteristics and features, many of which come with unique cybersecurity challenges that did not necessarily exist for each of the component pieces. While traditional cybersecurity is still necessary for each individual component in the aggregated technology, more unique challenges may require novel cybersecurity approaches or a new application of traditional cybersecurity techniques. Similarly, digital twin technology may enable a new and powerful paradigm of familiar components.

This section will explore new features in digital twin technology from a cybersecurity perspective, what challenges these new features present, how they might be secured, and how traditional cybersecurity approaches still apply to the individual components and mechanisms that make up digital twin technology.

7.1. Novel Cybersecurity Challenges

Digital twin technology has at least five novel features that require special cybersecurity considerations:

1. Massive instrumentation of objects (usually using IoT technology)
2. Centralization of object measurements
3. Visualization/representation of object operation
4. Remote control of objects
5. Standards for digital twin definitions that allow for universal access and control

This list is not exhaustive, and additional novel cybersecurity challenges will indubitably arise as digital twin technology matures.

7.1.1. Massive Instrumentation of Objects

Advances in IoT technology have led to the development of a variety of inexpensive and network-connected sensors that can be used to instrument objects. This instrumentation can then feed digital twin instantiations, enabling the modeling of real-world objects and real-time monitoring (and possibly remote control) of many objects to a fine level of granularity. This monitoring will likely be done with inexpensive, network-connected IoT sensors that may produce untrustworthy data or have vulnerabilities, availability issues, or limited computing capacity, network throughput, power, and upgrade potential. Such cybersecurity issues could lead to the inner workings of real-world objects being revealed and possibly controlled via the digital sphere.

7.1.2. Centralization of Object Measurements

Each sensor or controller is a separate IoT device. The sheer number and distribution of them could inhibit a malicious entity from completely taking control of the instrumented physical object or gaining a sufficiently broad view. However, digital twin technology involves centralizing data and control feeds from the massive instrumentation of an object. This creates great efficiency in simulation, modeling, and control, but it also centralizes sensitive data and control interfaces. If the digital twin is compromised, the attacker has total access to all data about the instrumented object.

7.1.3. Visualization and Representation of Object Operation

An attacker with control of a digital twin instance and instrumented object data could manipulate how the object is presented to users. For example, an attacker could manipulate the reality presented to a human operator using VR/AR, change the status of a monitored object and cause an operator to damage that object or the people around it, or remotely control digital twin while the visualization to the user hides any changes. Digital twin instances could even be designed to present object representations to other consuming digital systems, including other digital twin instances. Thus, a vulnerability in one instance could allow a hacker to affect or corrupt other linked instances that compromise a larger system.

Similarly, digital twin definitions may be built on top of one another following an object-oriented programming (OOP) model. They may also use an object representation from another digital twin definition to model objects that have some linkage, be it physical or virtual. The manipulation of a digital twin definition representation can then deceive or corrupt related digital twin definitions and other digital systems that consume the digital twin definition's object representation.

7.1.4. Remote Control of Objects

A hacked digital twin instance could provide an attacker with access to the raw remote-control mechanisms as well as a real-time, updated, digital facsimile with a possibly higher level of abstraction control mechanisms. These higher-level controls would be easier to understand and use. The attacker could manipulate these controls at the model level or at the level of the raw remote-control signals while deceiving any human operator by presenting a false digital facsimile.

7.1.5. Standards for Digital Twin Definitions That Allow for Universal Access and Control

In order to promote rapid integration of system, device and component models into digital twins, SDOs must develop widely applicable standards. Since these models may be distributed across hosting sites, SDOs must also develop standards for secure and reliable access and control.

7.2. Traditional Cybersecurity Challenges and Tools

The components that make up digital twin technology have traditional cybersecurity challenges as well, particularly in the areas of confidentiality, integrity, availability, maintainability, reliability, and safety. This section reviews some of these needs and the cybersecurity approaches that are commonly used to address them. Many of these techniques also apply to addressing the novel security challenges discussed in the previous sections.

Any serious effort to secure a digital twin system should follow more exhaustive risk management guidance, such as the NIST Risk Management Framework (RMF) [20], the NIST Cybersecurity Framework [21], and the NIST Privacy Framework [22]. Additionally, both a digital twin and its instrumentation should have cybersecurity controls implemented and tested to protect against attack using a comprehensive cybersecurity control catalog, such as the previously referenced NIST Cybersecurity Framework [21] or NIST Special Publication (SP) 800-53, Rev. 5, *Security and Privacy Controls for Info Systems and Organizations* [23]. Digital twin technology relies upon IoT cybersecurity for physical objects and instrumentation, and guidance is available through the NIST Cybersecurity for IoT Program [24]. Public and standardized encryption algorithms should be used to ensure the cybersecurity of data in transit from IoT devices to the central digital twin definition repository since proprietary encryption schemes can be weak and lack thorough vetting. Other mechanisms, such as hashes and error detection, should be used to verify the authenticity and integrity of the communications and the data regardless of strong encryption. Such additional protections will enable the system to protect against certain attacks, such as adversary-in-the-middle attacks and support non-repudiation and other services required for certain applications.

A digital twin instance, its current state, and the collected data should be encrypted when not being actively used in order to achieve data at rest cybersecurity. Data governance policies and mechanisms must be in place to ensure that only the correct staff have access to the necessary data within a digital twin instance. Strong authentication mechanisms must then support this governance to ensure that the access policies are not subverted. This can include two-factor or multi-factor authentication as well as the use of hardware keys. The physical security of a digital twin instance and related supportive IT system need to be maintained since physical access is often sufficient to circumvent many digital security mechanisms. This includes both the IoT instrumentation of the monitored object and the hardware maintaining the digital twin instance[4].

The software and hardware used for digital twin definition maintenance and simulation should be designed and tested to be robust and fault-tolerant since failure could result in significant physical world consequences. This is especially true since standards may enable digital twin instances to work with other instances based on other digital twin definitions, all of which will have differing sensitivities to faults and failures.

[4] An appropriate architecture for securing aggregated data from disparate databases with different access policies is discussed in DeFranco, J. F., Ferraiolo, D. F., Kuhn, R., & Roberts, J. (2021). A trusted federated system to share granular data among disparate database resources. *Computer*, 54(3), 55-62.

The entire digital twin system — including the instrumentation, control/data channels, digital twin definition, and visualization/representation mechanisms — needs to be properly authorized by appropriate organizational officials as having sufficient cybersecurity given the risk tolerance of the system. In addition, a privacy analysis should be conducted and privacy controls implemented based on a comprehensive privacy control catalog if the system contains any privacy-sensitive data (e.g., using the NIST Privacy Framework) [22].

Even the most secure networks have some connections to the outside world, even if they are not persistent (e.g., program updates through USB key transfers or the introduction of new hardware). It is best to plan cybersecurity based on a zero-trust model [25] where everything does its best to protect itself against everything else.

8. Trust Considerations[5]

This section lists a set of 14 trust considerations to help determine whether digital twin technology can provide desired operational functionality with an acceptable level of quality. Trust is the probability that the intended behavior and the actual behavior are equivalent given a fixed context, environment, and point in time. Here, trust is viewed as a level of confidence that a digital twin is functionally equivalent to a physical object, that a specific digital twin can be composed with another digital twin, that enough information is available about the environment and context of the physical object, and that digital twin technology can be standardized to the point where certification of a digital twin is possible.

1. **Digital twin creation ordering:** The point in time at which a digital twin is created will affect the correctness of the digital twin, such as whether it is created before the physical object is created or whether it is reverse-engineered from the physical entity that it is intended to mirror. While both approaches are valid, the fidelity of the digital twin may be reduced if it is created after the physical entity exists because there may be internal unknowns about the existing physical entity that cannot be discovered. For example, the source code for commercial off-the-shelf (COTS) software is unavailable to customers or integrators and, thus, hides internal syntax. This is a trust consideration for digital twins.
2. **Temporal:** Digital twin technology has an implied temporal component to it, particularly since it deals with physical objects that are bound by time. Hardware reliability theory and modeling states that physical objects suffer from levels of decay over time, even when idle. For example, if a car has not been turned on for years, the battery is likely dead, and the car will not start. However, a digital twin will not degrade or fatigue over time. Therefore, at some point, the real-world entity and the digital twin will be in conflict on some level, and synchronization of the two should occur. For example, a metal part could develop hairline fractures after usage that are not represented in the digital twin. This might suggest that the digital twin needs to be reworked or maintained to account for this. For example, a physical object at time t+1 will likely be different than at time t. However, the digital twin should be the same at times t and t+1 unless it updates dynamically with feeds from the physical object. Having access to an accurate timestamp [26] for the physical object and digital twin is a trust consideration.
3. **Environment:** Digital twin technology has an implied or explicit environmental component that cannot be overlooked. For physical objects, a description of the environmental tolerances or expected usage profiles is needed for many of the “ilities” [27], particularly interoperability. For example, bricks used to construct buildings are made from a variety of materials; some bricks will break easier under stress than others, and some bricks are better suited to certain temperatures and climates. This additional expected operational usage information should be stored with a digital twin. Without it, it will be difficult to determine whether the physical object is “fit for purpose” since purpose implies environment and context. Unknown environmental influences have

[5] An earlier version of these concerns, based on the draft of this NIST IR, appeared in P. Laplante, "Trusting Digital Twins." Computer 55.7 (2022): 73-77.

plagued safety-critical systems and software. Consider PowerPoint running during a presentation. Usually, the presenter does little more than touch the page-up or page-down keys. One could argue that the operational profile for executing PowerPoint during a presentation is two-fold: 1) the loaded presentation and 2) the button inputs from the presenter. However, whether the presentation goes smoothly (e.g., reliably and in a timely manner) is also a function of all of the inputs that PowerPoint is receiving from the disk, memory, and the OS in real time. If, for example, the presentation gets stuck going from slide x to slide x+1, then something related to "unknown" (i.e., phantom-like) environmental influences is probably involved (e.g., another process running on the machine at the same time and stealing resources and computing cycles). Accurately defining as many environmental factors as possible is a trust consideration.

4. **Manufacturing defects:** A digital twin may be used to guide a manufacturing process. For example, a factory that produces light bulbs will have a certain defect rate per thousand bulbs, and the packaging will offer an approximation for how long a bulb will operate before burnout. This highlights that a digital twin could not only describe the underlying components of an average bulb but also suggest how it should be manufactured if the representation details a metric, such as time-to-burnout. Ensuring that a manufacturing process produces a product with the correct life expectancy based on the information in a digital twin is a trust consideration.
5. **Functional equivalence:** Digital twin technology needs a means to determine functional equivalence between the digital twin and the physical object. If the digital twin is an executable specification, then it should produce the same outputs that the physical object produces for the same input data. Otherwise, functional equivalence has not been achieved. This could occur for a variety of factors, such as decay, fatigue, manufacturing variances, or other environmental influences that the physical object experiences during operation but that the digital twin does not. Without some assessment of the level of functional equivalence, it is difficult to argue for trustworthiness.[6]
6. **Composability and complexity:** A digital twin that is too complicated can create a composability problem in terms of predicting the trustworthiness of a final composed system from more than one digital twin. Assume that a system has five physical components (i.e., real-world entities), and each component has a corresponding digital twin definition. Physically connecting the five components may be straightforward, but composing the five digital twins may not be, particularly if the digital twins contain information such as tolerances and expected operational usages. Standards can help prune extraneous information contained in a digital twin by defining required interconnects between components of a domain and enabling the composition to be modelled and tested. One approach might be separating classes of information into categories, such as "need to know" and "extraneous."

[6] Verification and validation can be used to provide evidence of functional equivalence.

7. **Instrumentation and monitoring:** Instrumentation of a digital twin is a beneficial and unique advantage that digital twin technology offers. While one might not be able to instrument the physical object, one may be able to instrument the digital twin. However, instrumentation and probes are not as simple or easy to correctly inject into a digital twin as might be expected. Much can be learned here from the safety-critical software community. First, a determination of where to inject the probes is necessary [28], though this is not easy and can be more art than science. Second, the number of probes to inject is also a consideration. As shown in real-time systems, probes can slow down performance and timing, which may cause a problem for synchronization between the digital twin and physical object. That said, there are ways to reduce this impact by having the probes only collect raw data rather than compute internal test results, such as built-in self-tests. Collecting the "right" information from the internal state of an executing digital twin is an expensive and error-prone effort.
8. **Heterogeneity of standards:** Heterogeneity of different formats for digital twins may cause composability problems [29]. Composing digital twin definitions from different component vendors may not be achievable if vendors misuse standardized formats [4]. This is a consideration for trusting composed digital twins.
9. **Non-functional requirements:** Functional requirements state what a system shall do, negative requirements state what a system shall not do, and non-functional requirements (i.e., the "ilities"[7]) typically state what level of quality the system shall exhibit for both the functional and negative requirements. The "ilities" apply to both "things" and the systems into which they are built. The issue for digital twin technology concerns how many of the non-functional requirements can be written for the functional and negative requirements, thus defining the level of quality for what the system should and should not do. The ability to write these non-functional requirements will affect the ability to claim the trustworthiness of a composite object.
10. **Digital twin accuracy:** The degree to which the digital twin is correct is a trust consideration that may benefit from having more than one independently created digital twin for a specific physical object. In n-version programming [30], more than one independent software implementation is created for highly critical systems that the software impacts because no single implementation can be assumed to be adequately trustworthy. Each independent implementation is run in parallel, and the outputs from each implementation are sent to a voter that then decides on the final output that the system receives.
11. **Testing:** The testability of a digital twin refers to measuring how likely an error or defect will be detected during testing. Systems that are less likely to reveal the presence of defects are deemed less testable. Physical objects are testable to different degrees using this definition, though the methods for testing digital twins that are most likely to

[7] Examples of "ilities" include availability, composability, compatibility, dependability, discoverability, durability, fault tolerance, flexibility, interoperability, insurability, liability, maintainability, observability, privacy, performance, portability, predictability, probability of failure, readability, reliability, resilience, reachability, safety, scalability, cybersecurity, sustainability, testability, traceability, usability, visibility, and vulnerability [27].

demonstrate that the digital representation is correct are unclear. One option is to ignore this trust consideration and decide that a digital twin is untestable and, therefore, stands alone as the "oracle" or "gold standard." Moreover, although testing usually involves expected use cases, cases of misuse should also be considered.

12. **Certification:** Certification usually occurs by certifying either the process used to develop or the final artifact that comes from that process [31]. These two types of certification are distinct [3][32][33][34][36][37]. For digital twin technology, this means that one could attempt to certify how the digital twin was created or certify the accuracy of the digital twin itself. Certification of a twin will be complicated by information overload. For example, most prescription drugs come with warnings about who can take them, disclaimers about negative side effects, and when to discontinue use. However, the vast amount of information known about a drug and the vaster amount of unknown information about a drug at time *t* will not be known until time *t*+1, and much of this vital information is only understandable by medical experts. The trust consideration for digital twin technology is how much information can be provided in a specific digital twin without overloading a user with extraneous information that leads to confusion about how to use the twin or what the twin even represents.
13. **Propagation:** One of the greatest trust concerns with any system of systems is how errors and corrupt data propagate during execution [38]. Digital twin technology experiences this trust consideration as well, particularly when different twins representing different physical objects are composed. This may suggest that a digital twin should be wrapped with pre-conditions and post-conditions to determine whether the output from one digital twin will be acceptable as input to another digital twin.
14. **Counterfeiting:** A digital twin could be tampered with or counterfeited, and there are schemes that could protect against this. For example, digital twin definitions could be hashed and the hash posted to a public web page, or users of a digital twin definition could hash their copy and compare it to the hash on the public web page. This said, web pages and other similar publicly accessible repositories can be hacked. To enhance trust, one could use a blockchain to publicly post a digital twin definition hash in an immutable data structure that could never be changed, even by malicious attackers. Alternatively, identical copies of a digital twin definitions and related instances could be stored in separate locations (e.g., in offline backups).

9. Conclusions

Digital twin technology is an emerging area of research and standardization, though its core elements of modeling and simulation are already mature and widely used. Other significant components, such as VR, are also frequently deployed (even as low-cost gaming units in homes), and IoT sensors are becoming commonplace. Because of this, there may be a lack of clarity as to what is new with digital twin technology and the promise it holds.

This work discusses the characteristics and underlying technology for digital twin, the motivation and vision for its use, common low-level operations, usage scenarios, and example use cases. It also discussed technical considerations for the cybersecurity and trust of digital twin technology by analyzing both traditional and novel cybersecurity challenges. Furthermore, this work evaluated 14 trust considerations for digital twin functionality and quality and mapped those evaluations to other NIST cybersecurity guidance.

The authors hope that SDOs and digital twin implementers will use this document to ensure the secure and trustworthy development of standards and architectures for digital twin technology as it progresses.

References

[1] Digital Twin Consortium (2020) *The Definition of a Digital Twin*. Available at https://www.digitaltwinconsortium.org/hot-topics/the-definition-of-a-digital-twin.htm

[2] Merriam-Webster (2021) *Merriam-Webster.com Dictionary*. Available at http://www.m-w.com

[3] Voas JM, Hurlburt G (2015) Third Party Software's Trust Quagmire. *Computer* 48(12):80-87. https://doi.org/10.1109/MC.2015.372

[4] Voas JM, Laplante P (2007) Standards Confusion and Harmonization. *Computer* 40(7): 94-96

[5] Piroumian V (2021) Digital Twins: Universal Interoperability for the Digital Age. *Computer* 54(1):61-69

[6] W3C (2021) HTML 5.2. W3C Recommendation, 14 December 2017, superseded 28 January 2021. Available at https://www.w3.org/TR/html52/

[7] W3schools.com (2021) HTML Element Reference, Available at https://www.w3schools.com/tags/default.asp

[8] 3D PDF Consortium (2020) PDF in Manufacturing, pp 22-26

[9] Goldberg A (1983) *Smalltalk-80: The Interactive Programming Environment* (Addison-Wesley, Reading, MA)

[10] Yourdon E, Constantine L (1979) Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design (Prentice-Hall, Upper Saddle River, NJ)

[11] Booch G (1990) *Object-Oriented Analysis and Design with Applications*. (Benjamin/Cummings, Redwood City, CA)

[12] Woods RL, Lawrence KL (1997) *Modeling and Simulation of Dynamic Systems* (Prentice Hall, Englewood Cliffs, NJ)

[13] Costello S (2021) *Internet Streaming: What It Is and How It Works* (Lifewire), July 9, 2021. Available at https://www.lifewire.com/internet-streaming-how-it-works-1999513

[14] Costello S (2021) *What Is Streaming? How Video Streaming Works*. Available at https://www.cloudflare.com/learning/video/what-is-streaming/

[15] Reynolds JC (1998) *Theories of Programming Languages* (Cambridge University Press, Cambridge UK)

[16] Scheifler R, Gettys J, Flowers J, Rosenthal D (1992) *X Window System, The Complete Reference to Xlib, X Protocol, ICCCM, XLFD*. (Digital Press)

[17] Palamidessi C, Glaser H, Meinke K (1998) Principles of Declarative Programming. *Proceedings of the 10th Annual Symposium PLILP'98.* Springer. Available at https://doi.org/10.1007/BFb0056603

[18] Minerva R, Lee GM, Crespi N (2020) Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models. *Proceedings of the IEEE* Volume 108, Number 10, pp. 1786, 1790, 1792, 1775-1796, 1800, 1804-1805

[19] Digital Twin Consortium (2020) Available at https://www.digitaltwinconsortium.org

[20] National Institute of Standards and Technology (2021) *NIST Risk Management Framework*. Available at https://csrc.nist.gov/projects/risk-management

[21] National Institute of Standards and Technology (2021) *NIST Cybersecurity Framework*. Available at https://www.nist.gov/cyberframework

[22] National Institute of Standards and Technology (2021) *NIST Privacy Framework*. Available at https://www.nist.gov/privacy-framework

[23] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-53, Rev. 5. Includes updates as of December 10, 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[24] NIST Cybersecurity for IoT Program. Available at https://www.nist.gov/programs-projects/nist-cybersecurity-iot-program

[25] Kerman A, Borchert O, Rose S, Division E, Tan A (2020) Implementing a Zero Trust Architecture. (National Institute of Standards and Technology, Gaithersburg, MD), Project Description. Available at https://www.nccoe.nist.gov/library/implementing-zero-trust-architecture

[26] Stavrou A, Voas J (2017) Verified time. *Computer*, 50(3):78-82

[27] Voas J (2004) Software's secret sauce: The `ilities'. *Software*, 21(6):2-3

[28] Voas JM, Miller KW (1994) Putting Assertions in Their Place. *Proceedings of the International Symposium on Software Reliability Engineering* (IEEE, Monterey, CA), pp 152-157. Available at https://doi.org/10.1109/ISSRE.1994.341367

[29] Voas JM (2016) Networks of 'Things'. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-183. Available at https://doi.org/10.6028/NIST.SP.800-183

[30] Chen L, Avizienis, A (1978) N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation. *Proceedings of the Eighth International Symposium on Fault-Tolerant Computing*, pp. 3-9

[31] Voas J (1998) The Software Quality Certification Triangle. *Crosstalk* 11(11):12-14

[32] Voas J (1998) Certifying off-the-shelf software components. *Computer* 31(6): 53-59

[33] Voas J (1999) Certifying software for high assurance environments. *Software* 16(4):48-54

[34] Voas J, Payne J (2000) Dependability certification of software components. *Journal of Systems and Software* 52(2):165-172

[35] Voas J (2000) Toward a Usage-Based Software Certification Process. *Computer* 33(8):32-37

[36] Voas J, Laplante P (2017) The IoT Blame Game. *Computer* 50(6):69-73

[37] Voas J, Laplante P (2018) IoT's certification quagmire. *Computer* 51(4):86-89

[38] Voas J (1997) Error propagation analysis for COTS systems. *IEEE Computing and Control Engineering Journal*,8(6):269-272

Appendix A. List of Symbols, Abbreviations, and Acronyms

Selected acronyms and abbreviations used in this paper are defined below.

2D
Two-dimensional

3D
Three-dimensional

AI
Artificial Intelligence

AR
Augmented Reality

CNC
Computer Numerical Control

COTS
Commercial Off-the-Shelf

DT
Digital Twin

HTML
HyperText Markup Language

IoT
Internet of Things

IT
Information Technology

NIST
National Institute of Standards and Technology

OSI
Open Systems Interconnection

TCP/IP
Transmission Control Protocol/Internet Protocol

UAV
Unmanned Aerial Vehicle

VR
Virtual Reality

WYSIWYG
What You See Is What You Get

Appendix B. Glossary

digital twin
The virtual (i.e., digital) representation of a physical or perceived real-world entity, concept, or notion.

digital twin definition
A machine-readable specification that describes features that may be modeled for a particular type of real-world entity.

digital twin instance
A digital data structure, object, or entity in a computer software environment that represents a specific physical instance of a real-world object whose type or class is given by an associated digital twin definition.

digital twin application software
A software application that comprehends, manipulates, reads, writes, or modifies digital twin definitions and instances according to the digital twin standard.