Draft NISTIR 8356

Considerations for Digital Twin Technology and Emerging Standards

Jeffrey Voas
Peter Mell
Vartan Piroumian

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

National Institute of
Standards and Technology
U.S. Department of Commerce

Draft NISTIR 8356

Considerations for Digital Twin Technology and Emerging Standards

Jeffrey Voas
Peter Mell
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-draft

April 2021

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce for Standards and Technology & Director, National Institute of Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report 8356
34 pages (April 2021)

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

Public comment period: *April 16, 2021* through *June 16, 2021*

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: nistir-8356-comments@nist.gov

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

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.

Abstract

Digital twin technology enables the creation of electronic representations of real-world entities and the viewing of the state of those entities. Its full vision will require standards that have not yet been developed. It is relatively new although it uses many existing foundational technologies and, in many cases, appears similar to existing modeling and simulation capabilities. This report attempts to provide clarity in understanding the concept and purpose of digital twins. It offers a new definition for a digital twin, and describes characteristics, features, functions, and expected operational uses. The report then discusses novel cybersecurity challenges presented by digital twin architectures. Lastly, it discusses traditional cybersecurity challenges as well as trust considerations in the context of existing NIST guidance and documents.

Keywords

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

Audience

This publication is accessible for anyone desiring to understand the envisioned capabilities of digital twin technology as well as the associated cybersecurity and trust issues. It is particularly applicable to developers of digital twin standards as well as implementers of the technology.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

   i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
   ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: nistir-8356-comments@nist.gov

Table of Contents

List of Appendices

1 Introduction

Digital twin (DT) technology is an emerging field that needs additional definitional clarity as well as an analysis of cybersecurity and trust considerations. As with many new technologies, digital twins are in that initial period of flux characterized by a healthy dose of evolution, discussion, and confusion regarding what a digital twin really is—or should be—and what its full scope and applicability will be.

Digital twin technology enables the creation of electronic representations of real-world entities and the viewing of the state of those entities. These entities can be either physical or perceived (e.g., processes). The digital facsimiles can represent entities to be constructed or existing entities. With existing entities, the copies can be static (representing a point in time) or dynamic (linked to an entity and continuously updated). Dynamic twins can be used to control their linked objects. Twins can be instrumented to test their functionality and utility. Multiple digital twins can be composed, and the composition of twins can be tested.

As with many new information technologies, it uses many existing foundational technologies and it many cases appears similar to existing capabilities. It covers what exists today in modeling and simulation but then casts a broader vision for what could be. Standards will play a significant role as the full digital twin vision will not be possible without interoperable digital twin definitions and tools. Particularly important in this space are considerations for digital twin cybersecurity and trust. For any new digital technology, cybersecurity and trust should be addressed early. Digital twin technology is no exception; this is especially true for nascent standards efforts that seek to define and structure the technology.

This report provides an introduction to digital twin technology, to provide clarity on what it really is and how it expands on current capabilities. It explains the important components and functions, and then discuss cybersecurity and trust considerations. It begins with a section defining digital twins followed by a section providing the motivation for using digital twin technology. This is followed by a discussion on typical operations performed on digital twins and then an explanation of technical usage scenarios. It then provides example applications of digital twin technology in industry. Having given the reader a high-level understanding of digital twin technology, the report then explores cybersecurity consideration. It identifies and explores novel cybersecurity challenges related to digital twins. It also discusses traditional cybersecurity needs and approaches that apply to digital twin systems. Lastly, it discusses trust issues that can inhibit a digital twin implementation from providing the desired operational functionality with an acceptable level of quality. These last two discussions of traditional cybersecurity and trust considerations are conducted in the context of existing NIST guidance and documents.

2 Definition of Digital Twins

Currently, there are several unofficial "definitions" for digital twins—some created by researchers, some by standards committees and consortia, some by industry, and still others that are implicitly suggested by commercial enterprises that make statements that their software applications are "digital twin-compliant" in spite of the absence of any agreed-upon definition or consensus of vision for digital twins [1].

Despite today's nebulous understanding and lack of formal definition of what digital twins really are—or will eventually become—the definition of a digital twin used in this paper is:

> *A digital twin is the electronic representation—the digital representation—of a real-world entity, concept, or notion, either physical or perceived.*

In practice, a digital twin will consist of a definition that will be created and persist in a digital computer environment. Computer software applications will read digital twin definitions to present a human user with a virtual view of the real-world object represented by the digital twin.

The use of the term *virtual* is appropriate here. A common theme that is part of every conversation or unofficial definition of digital twins is that software applications will typically present for the benefit of a human user a visual graphic representation, either static or dynamic, of a real-world object via the object's digital twin. Many of these real-world entities will be things that are commonly recognized as having physical form, such as an aircraft engine, an oil derrick, a valve in an oil pipeline pumping station, a bicycle, or a human heart.

However, according to the definition of digital twins given above, an entity represented by a digital twin can also be something abstract. The following definition of *abstract* from the Merriam-Webster online dictionary is appropriate for a discussion of the various types of entities that a digital twin can represent:

> **abstract**: noun: *expressing a quality apart from an object* [2]

In this context, a digital twin could represent a real-world entity that is neither concrete nor physical in the traditional sense of these words. That is, the digital twin could represent something that is certainly real in the sense that it is perceived by a human being as something that truly exists—it just does not have an identifiable physical form in the traditional sense.

A good example of such an entity that is real but is without traditional physical form or manifestation is a *process*. For instance, a *process* in a computer operating system is certainly real, but it is quite different from the typical entity that one would normally think of as physical. Our cognitive human perception tells us that the computer process is indeed real; one can clearly observe the effects it has on other objects, such as the disk drives in a computer. However, its nature as a concrete, tangible, physical entity is not so obvious.

The computer operating system process itself is really a conglomeration of other intangible things, such as electrical signals, the states of registers containing voltage and current levels, the electrical state of memory, and so forth. To further complicate matters, one can view these register states and electrical signals as representing the running computer program. The program itself is an abstract entity. Its physical form, residing as a magnetic state on a hard disk, is certainly not the notion one has of the program when looking at its graphical user interface. Likewise, the programmer's view of the source code or the breakpointed state of the program in a debugger are quite different manifestations from the magnetic state of sectors that store the program on a hard disk drive. Nonetheless, one can think of a computer program—either statically or dynamically—as something real.

A *business process* is another example of a process and a very abstract entity that is certainly real to a human, albeit not so material. Yet one could define a digital twin to represent a business process. In fact, software exists today to do just that.

A *manufacturing process* is yet another example. Chemists and chemical engineers define processes to produce compounds and chemicals, as well as facilitate material processing, such as oil refining or the production of nuclear fuel. A chemist could describe such a process using pencil and paper, or a digital twin could describe it, its steps, or a simulation of every aspect of the dynamic execution of such a process in a factory or chemical plant. Clearly, these processes are real.

The notion of a process described above is certainly not the only type of abstraction that digital twins can represent. As stated previously, a digital twin can represent anything that a human can conceive or perceive, whether real or not, material or not, or physical or not—basically, any abstraction.

For example, astrophysicists and cosmologists have postulated the existence of black holes, but none are known to exist. Nevertheless, scientists have created both static and dynamic models of black holes, and these models include the characteristics and dynamic behavior of black holes according to theory and the mathematical equations that attempt to describe their static and dynamic nature. Such a model could be built via a digital twin and used by computer programs to present humans with static views and dynamic simulations of the behavior of a black hole.

3 Motivation and Vision

Nothing in our definition of digital twin technology is new; it has all existed in one form or another. The use of computers and software to represent entities and simulate dynamic behavior has existed since the first computers. Engineers use software to design, simulate, and verify amusement park roller coasters, bridges in high winds, buildings in earthquakes, and the aerodynamics of aircraft.

However, this does not mean that digital twin technology is just a repackaging and renaming of existing technology. The focus on, and interest in, digital twins is due to the maturation of multiple underlying technologies that is making it possible to apply simulation and modeling, in the form of creating digital representations, much more broadly and make it accessible to a much wider user base. The emergence, through the Internet of Things (IoT) revolution, of small low-cost battery powered sensors that are network connected has enabled 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 expensive hardware. And finally, the recent advances in virtual reality (VR) and augmented reality (AR) have enabled inexpensive visual viewing of digital twin representations.

Whether or not these developments catalyze digital twin technology into widespread use may depend upon work in standards development. Currently most IoT systems, simulation and modeling software, and VR and AR systems exist in stovepipe proprietary systems. It is possible to combine them, but it takes significant work to integrate them. Much of the work in the emerging digital twin area is in the creation of protocols and standards to enable plug and play integration. The idea is to mix and match and be able to use any viewer with any digital twin simulator and modeler along with any sensing device. The idea 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. These are lofty goals for the emerging digital twin community; their success in standards may largely determine the extent to which the technology is used.

3.1 Advantages of Digital Twin Technology

Creating a platform or mechanism that supports the creation of digital models of real-world objects is advantageous for several reasons. One major motivation is that one can study the object via its model prior to building the real-world version. This practice can reduce certain types of risk. This advantage increases when multiple entities are modeling different objects that need to work together. If cooperating entities can share digital twin definitions, then they can more easily model and simulate object interactions digitally prior to the realization of the output product.

In fact, this is the case today in select industries; it is simply not widely deployed and done in proprietary systems. All manner of real-world objects is conceived, described, modeled, and designed on the computer, from electrical circuits, integrated circuits, and amusement park rides to bridges, oil derricks, aircraft, power plants, and nuclear reactors. All of the engineering analysis is also performed on a computer prior to building anything physical. The software applications employed to do this work contain dynamic 2D and 3D graphics to visually represent everything from the interaction of atoms and molecules in chemical and biological reactions to fatigue and flutter in aerodynamic structures.

3.2 Expectation of Standards

The adoption of standards and the adherence to standards by the systems built around them 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]. However, no digital twin specific standard currently exists (although certainly digital twin implementations can leverage many generic information technology standards). When such standards do exist, there may be multiple cooperating (or competing) ones. Rarely does a single standard adequately address everyone's needs. In this case, there must be some synergy such as that realized by either harmonization or standards blending [4]. In the standards blending approach, each standard—in fact, each functional area of each standard—will still need to be vetted. Tool vendors, software and hardware application vendors, and users can be compliant with all standards by ensuring that they use only vetted elements. Using this approach should lead to the interoperability of tools and applications.

Additionally, for standards to work they would need to cover each involved business domain. Everyone operating in the digital twin ecosystem in specific business domains would need standards and standards-based products that adhere to a common set of business processes and use cases such that interoperability can be achieved in the business domain. It is not enough to have standards just to enable interoperability in the purely technology domain.

3.3 A More Detailed Look at Supportive Technologies

This section discusses in more detail two of the supportive technologies undergirding the recent interest in digital twin technology: VR/AR and IoT.

One will frequently encounter VR and AR in digital twin discussions online and in the opus of digital twin-related publications and articles. One of the visions for digital twins is to leverage VR and AR to create the enhanced user interfaces and user experiences for human beings to comprehend complex entities. Humans rely heavily on visual sensory input, and VR and AR promise to help describe real-world entities through the models created for them.

IoT is also often referenced in digital twin discussions and literature because of the recent advances in sensors and sensing. There is both a significant advance the creation of sensors of all kinds while there is also an ongoing, dramatic proliferation of such sensors being put into operation. These sensors are typically network connected and drive the ability of twins to model real world objects in ways that were not possible until recently.

IoT is often used to create what is referred to as an information fabric or "network." This fabric encompasses more than just the sensors although the sensors are obviously the central component. An IoT fabric consists of the observed entities, the sensors that observe and gather information, the connectivity elements, the processing components (i.e., what one might think of as “the back-end compute servers”), and the components that process the 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. The digital twin model may also be manipulated by a user to control the actual object. This is where digital twin technology may advance beyond traditional modeling and simulation software.

4 Operations on Digital Twins

In an effort to provide a full overview of digital twin technology, this section discusses lower level operations that are performed on digital twins:

- Definition of digital twins and the creation of digital twin electronic artifacts
- Manipulation and modification of electronic digital twin definitions
- Exchange via electronic communications of digital twin artifacts

Note that these are operations typically performed on electronic assets; we discuss here how they apply to digital twins and any special considerations.

4.1 Creation and Definition of Digital Twins

A digital twin *definition* is really a *formal description* of the real-world twin—the real-world entity—that the digital twin represents. The starting point for all activity involving digital twin technology is the creation and persistence of digital twin definitions as digital artifacts by computer software applications. These artifacts can represent both static and dynamic models of the real-world entities that correspond to their respective digital twin.

Since there is currently no digital twin standards definition, there is no formalization of a digital twin definitions at this time. The type of digital twin defined will dictate the precise makeup of the definition files. For example, there is no requirement that a given digital twin definition include a dynamic view of its real-world counterpart; it could comprise only a static view of the object, if so desired. 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 the digital twin represents only a static model of an entity, there would be 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 internals of the ship seen through VR as if a person was conducting a literal walk-through of 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 creates a model of the object it represents, not just a particular view. It can present whatever view and viewpoint of the real-world entity that the author desires.

A digital twin definition can contain as much or as little information about its real-world counterpart as its authors desire. The breadth, scope, degree of granularity, and detail are completely the decision of the creators. As with other models, the *nature* of the real-world object and its digital twin's presentation of it are independent of the scope, extent, granularity, and level of detail contained in the digital twin definition. The main consideration is that all aspects of the model created by the digital twin—nature, scope, granularity, and level of detail—should be suitable for the intended application of the digital twin definition.

A practical consideration is the digital twin definition authoring itself. While it might very well be possible to author definitions using a text editor, this practice could become supplanted by tools. This is similar to how not many people hand-code HTML or XML anymore [7]. The complexity of digital twin definitions could even preclude the ability to craft definitions by hand.

In industries such as aerospace, civil engineering, mechanical engineering, and heavy construction, designers, modelers, and engineers use sophisticated software applications to create digital artifacts that represent what they plan to build, including aircrafts, bridges, buildings, amusement park rides, dams, tunnels, cranes, oil rigs, and a plethora of other things. Some of these software applications support the writing, persistence, and "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 completely proprietary file formats and encodings to define, capture, and persist the models, drawings, and various artifacts that they can create.

Standards will be important here. Like existing commercial applications, any future digital twin standard may include a *language* for describing and defining digital twins. Such a language, like any other language, would consist of a *formal grammar*, *syntax*, and *semantics*. It would have to be comprehensive enough—both general and specific—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 2D, 3D, VR, and AR models, both static and dynamic, for visual presentation to human users. It might also accommodate the creation, manipulation, and persistence of presentation forms that might not be human-comprehensible (i.e., presentations intended for machine consumption).

4.2 Manipulation and Modification of Digital Twin Definitions

Once a digital twin definition exists, it will be available for viewing (reading), modification (editing), resaving modifications, and duplication. This refers to the review or viewing of the definition itself, not the viewing of the *presentation* of the digital twin's model of the real-world entity that it describes.

If standards are developed, industry may develop "what-you-see-is-what-you-get" (WYSIWYG) editors and tools to complement the standard text editors that exist, which include stand-alone editors as well as interactive developer environment (IDE) tools. Such a tool would comprehend the digital twin definition language and its file formats, encodings, grammar, syntax, and semantics. This would enable the tool to correctly read a digital twin definition in order to support its review or modification by a human user. This is no different than the need for a WYSIWYG HTML editor to comprehend how to properly produce formatted HTML source.

4.3 Exchange of Digital Twin Definitions

Digital twin definitions are simply computer files that are available for reading, writing, execution, and general manipulation. They can be sent to recipients for use, similar to how 3D printer files are shared to enable many 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 categories of usage of digital twins in practice. The broad categories of application for digital twins are:

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

5.1 Viewing Static Models of Digital Twins

This section describes the viewing of static digital twins. This type of view presents a non-changing model of a real-world twin, regardless of the nature of the real-world entity and regardless of how that entity may change over time. The real-world entity may not even exist yet as 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. Static descriptions do not model an object's behavior [11].

Examples include a computer numerical control (CNC) milling machine using a static 3D model to describe the object to be milled. In the aerospace industry, designers or modelers first create what they call a *solids 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 (tail assembly). A building architect creates drawings of a house to be built. Typically, the architect creates various 2D views, such as a site plan, floor plan, and elevation plans. Architects could adopt the practice of creating 3D views, but these would be less useful to building contractors, and they are more difficult to read. The creator of the model can define it as they see fit for the intended use, and the data in the model can be used to create a suitable view.

5.2 Executing and Viewing Dynamic Models of Digital Twins

This section describes scenarios in which a human user executes a digital twin definition to model an object's changes over time and views the dynamic changes to the object. As with the static viewing, 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; it describes how an object changes as measured via one or more metrics that represent one or more aspects or characteristics of the object [12].

Some examples are the visual updates to graphics that show how the scaffolding or track of a roller coaster bends as a function of applied force from wind loading or from the traveling toboggan, the dynamic response of a building—how the building moves—in an earthquake, or how an aircraft's wing bends under changing loads in flight.

Dynamic modeling, simulation, and presentation (display) are quite different from the static modeling and visual presentation of objects discussed in the previous section. An engineer who wants to understand how the milled block mentioned above 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 or information about the shape, material, or density of an object.

For the aerospace engineer, a static model is adequate to show the material or dimensions of a component, such as a wing assembly. However, one needs a dynamic model to analyze how the wing performs dynamically in a wind tunnel or in flight and to understand materials stress, fluid dynamics, and dynamics (e.g., natural frequency, normal modes, and flutter).

In dynamic modeling, simulation software applications update their model and view at some rapid frequency that represents real-time or near real-time behavior. A dynamic display should be updated at a frame rate that ensures smooth motion, such as 24 frames per second or higher.

Although most engineers would envision a graphical user interface when one mentions dynamic simulation and modeling, the presentation—the MVC *presentation or view*—need not be graphical. It 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.

A visual presentation can use many methods to create a more comprehensible presentation. 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. The choice is up to the author of the model and its views, all of which represent the dynamic nature of the object being simulated.

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 (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, which represents the *imperative programming paradigm* to display graphics [15]. Applications make calls to X library routines (and those of the graphics toolkits built atop 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 present (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; 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 dynamic simulations could consist of pre-captured video, such as an MPEG-encoded video. In this case, the digital twin application software probably creates the standard video from the simulation run.

Alternatively, digital twin software applications could create declarative-style content to be 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 to point to other content such as pre-captured or pre-recorded video or even executable code that is in the imperative style. Web pages today contain directives to download executable code, such as JavaScript, that are executable programs.

5.3 Real-time Monitoring of Real-world Entities

This section describes monitoring the state or condition of real objects. Monitoring is fundamentally different from simulation. Monitoring collects the information—typically in real-time or near real-time—of actual real-world entities. There is no simulation or emulation at all.

A tangible example would be the real-time monitoring of the state of an aircraft's wing in flight (e.g., in a wind tunnel, on a test bench, or in real flight on a real aircraft). The monitoring could capture information about anything, such as measuring materials stress or fatigue, aerodynamic drag, or laminar flow.

Typically, monitoring will involve the use of sensors to gather the data to be used to construct a view of the state of the real-world entity being monitored. Today, the term "sensor" may immediately bring to mind the "Internet of Things" (IoT) [18], but the terminology is an unnecessary distraction. Conceptually, a sensor can exist anywhere. It can be on the object being observed and monitored, or it can be physically remote. For example, satellites now have sensors that perform ground imaging. Upon gathering data that represents some aspect of the object's state or condition, sensors send the collected data to software systems. The system could be local or remote, and remote connections could be wired or wireless. They could also use any of a number of transmission mediums, technologies, and protocols.

These systems are not new. Airline operations centers have been monitoring their aircraft in-flight for years. The vision for digital twins is to use this monitoring data to create a dynamically updated digital replica and to, through standards [19], enable the interoperability of different tools to view and manipulate the digital replica. This could enable, for example, an operator to take a current digital replica of an aircraft and manipulate it to apply artificial stressors to calculate how that particular airplane in its lifecycle and with its specific operating parameters can handle unexpected events.

Monitoring can be real-time or in near real-time. A typical jet engine has numerous sensors that monitor every imaginable aspect of the engine's parts and operations. The data gathered by the sensors can be transmitted to applications that present 3D graphics or VR or AR experiences to users. An airline mechanic can use a VR system to view an operating engine in flight as if they were standing in the engine to examine a particular part.

5.4 Real-time Command and Control of Real-world Entities

Real-time links to a real-world object create an opportunity for yet another usage scenario, namely command and control via a digital twin platform. Command and control systems require bidirectional links and the transmission of information. The digital twin platform is amenable to presenting a view of the object or objects associated with their digital twins. However, the software systems built around the digital twin cornerstone can command the observed objects, such as an oil well drill, deep sea exploration vessel, unmanned aerial vehicle (UAV), satellite, or spacecraft.

Sensors and communications mechanisms collect and send status information to the main control. The main system will most likely present the object's status to a human user, although this is not required. In response to some evaluation, assessment, or processing of the status information, commands are sent back to the object to modify its state or command it to do something.

This kind of bidirectional command, control, and communications is not new. The difference with digital twin technology is the detailed digital representation of the controlled object that is itself a model of the object (not just information feeds about the object). Also, digital twin standards may enable interoperability between tools and formats to enable this 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

This section presents a few sample applications of digital twins to prepare the reader for the subsequent discussion of the potential cybersecurity vulnerabilities of digital twin environments and ecosystems. The examples in this section are taken from industries that are already exploring and using digital twin technology.

Unmanned aerial vehicles (drones): The unmanned aerial vehicle (UAV) or drone is used in environmental monitoring. UAVs come in all sizes, shapes, configurations, and sophistication. The more advanced drones can operate either autonomously or via the control of a human sitting in a command and control center miles away from the UAV's physical location. In the case of UAVs that are operated and controlled remotely, the operator has a user interface that gives real-time status on many aspects of the UAV's state and condition.

Ocean-going supertankers: Digital twin systems could be used at every life stage of various ocean-going vessels, from naval vessels to commercial supertankers. During construction, digital twins could be used to construct 2D or 3D static views or VR/AR views of the vessel. Such VR or 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 crewmembers to monitor every aspect of the ship, from its propeller screws and drive shafts to the engine room, providing views to someone on the ship's bridge—possibly precluding the need for certain physical monitoring and inspection.

Oil derricks and ocean drilling platforms: Oil derricks drill for oil in some of the most inhospitable and potentially treacherous environments. Sometimes, they drill down to great depths below the ocean's surface. Systems built around digital twins could enable designers, engineers, and operators to form a model and visual representation of the oil rig, drill rig, and drill bit head deep down in the ocean.

Telemedicine and remote patient monitoring: There are currently systems that remotely monitor certain parameters of a patient's health. One system places a wireless communications and sensor apparatus under the patient's bed, monitoring the signals sent by the patient's pacemaker and sending the data to a doctor or hospital.

Robotic surgery: There are surgical robots that perform various kinds of surgery. Some require an actual human surgeon to control the robot, but others require only a human surgeon to monitor the robot's automatic execution of the surgery. Future systems built around standards based digital twin technology could enable interoperability. For instance, one company's surgical robot could be able to interoperate with another company's VR system that specializes in the representation of human organs.

7 Cybersecurity Considerations

New trends in computing may appear to be little more than a simple rebranding of existing technology, such as cloud computing and big data. However, a closer inspection can reveal that the integration of known components combined with a certain maturation in the industry has created novel characteristics and features, and these new capabilities often come with unique cybersecurity challenges that did not necessarily exist for each of the component pieces. These challenges may then require novel cybersecurity approaches or a new application of traditional cybersecurity techniques. That said, the traditional cybersecurity necessary for each individual component is almost always still necessary in the aggregated technology.

Digital twin technology is no different. The components of digital twin technology (e.g., instrumented devices, aggregated metrics, visualization, and remote control) are not new. However, in the aggregate, it may enable a new and powerful paradigm. This section will explore what is new 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 into digital twin definitions
3. Visualization/representation of object operation through digital twin definitions
4. Remote control of objects through manipulation of digital twin definitions
5. Standards for digital twin definitions that allow for universal access and control

This list is not necessarily 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 provided a wide variety of cheap and network-connected sensors that can be used to instrument objects. This instrumentation can then feed digital twin definitions, 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. These sensors may have vulnerabilities as well as limited computing capacity, network throughput, power, and upgrade potential. This has cybersecurity significance because the detailed innerworkings of real-world objects would be revealed (and possibly controlled) via the digital sphere. Previously, such objects were protected from digital interference because they were not digitally instrumented. All of that would change with digital twin instrumented objects (e.g., 'smart' homes, 'smart' buildings, and 'smart' cities).

7.1.2 Centralization of Object Measurements

This massive instrumentation of objects could provide a malicious entity visibility into (and perhaps control over) the detailed inner workings of an object through the compromise of the sensors. However, 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. However, digital twin technology involves centralizing the 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 definition is hacked, the attacker has total access to all data about the instrumented object.

7.1.3 Visualization/Representation of Object Operation

An attacker with control of the digital twin definition and instrumented object data could manipulate how the object is presented to the users of the digital twin definition. There is heavy emphasis in digital twin technology on VR and AR. Control of the digital twin definition enables manipulation of the reality presented to the human operator. The design of an object to be built could appear to be correct when in reality the attacker has redesigned it with flaws. The status of a monitored object could be changed to cause an operator to take action that would then damage the object or the people and things around the object. A remotely controlled digital twin object could be manipulated by an attacker while the visualization to the user hides any changes.

Since a digital twin definition can present its related object's state through more than just visualization to a human, digital twin definitions could be designed to present object representations to other consuming digital systems, including other digital twin definitions. 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 representation can then deceive or corrupt related digital twin definitions and other digital systems consuming the digital twin definition's object representation.

7.1.4 Remote Control of Objects

A goal of digital twin technology is to have a detailed simulation of an object through extensive instrumentation and use that simulation to remotely control the object. Remote control technology has long existed for many types of objects; what is different here is the goal of building a digital facsimile that is constantly updated and to add controls to the digital representation while having the effects transmitted to the controlled object.

A hacked digital twin definition would then not only provide an attacker access to the raw remote-control mechanisms but also to a real-time, updated, digital facsimile with 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 digital twin definition level or at the level of the raw remote-control signals while deceiving any human operator by presenting a false digital facsimile. This could deceive the user who is relying on the provided digital facsimile rather than the individual raw metrics from the instrumented object.

7.1.5 Standards for Digital Twin Definitions

A significant push in the digital twin technology community is to create standards for digital twins that will be adopted by tools for digital twin definition creation, monitoring, linkage to monitored objects, and remote manipulation. This standardization push, if successful, would enable any standards-based tool to work with any digital twin definition. This is a significant factor in the excitement around digital twin technology because it could eliminate what are now proprietary silos of remote-controlled instrumentation.

Standardization, while beneficial in general, could aid attackers in subverting digital twins. Standardization in the IoT devices (and their communication protocols) used to instrument objects could make it easier for attackers to decipher measurements and send commands to the instrumentation. Standardization of the digital twin definition representation could enable an attacker to more easily subvert an existing definition or replace one altogether (e.g., with a malicious one created by the attacker). It could enable an attacker to more easily provide false visualizations to human operators. In summary, standardization could vastly reduce the learning curve for attackers to manipulate digital twin definitions by removing the unique proprietary measurement and control mechanisms that exist in many of today's remotely monitored and controlled objects.

Another possible scenario (that is similar to the problem today of malicious apps in app stores) is that of attackers creating useful looking but malicious digital twin definitions and providing them to the public. Digital twin standards would make it easy for anyone to understand how to write a digital twin definition and thus how to create a false one that might appear to be authentic. This is similar to phishers who create emails that look legitimate but lead users to malicious web pages. Once published, anyone with a standardized digital twin toolkit could then execute the malicious definitions.

7.2 Traditional Cybersecurity Challenges and Tools

While the novel cybersecurity challenges of digital twin architectures have been the focus of attention so far, the components that make up digital twin technology have traditional cybersecurity challenges that must also be addressed. These challenges include the areas of confidentiality, integrity, availability, maintainability, reliability, and safety. This section reviews some of these needs as well as cybersecurity approaches commonly used to address them. This is not intended to be an exhaustive list but rather a sampling of important and obvious traditional cybersecurity that will need to be implemented. Any serious effort to secure a digital twin system should follow more exhaustive risk management guidance. For U.S. Government systems (also applicable to any system), this includes the NIST Risk Management Framework (RMF) [20]. Critical to the cybersecurity of all systems is the NIST Cybersecurity Framework [21]; privacy controls are covered by the NIST Privacy Framework [22].

Digital twin technology focuses on the instrumentation and control of an object. Both the digital twin implementation and its instrumentation should have cybersecurity controls implemented and tested to protect against attack using a comprehensive cybersecurity control catalog (e.g., using the previously referenced NIST Cybersecurity Framework [21] or NIST Special Publication 800-53, Rev. 5, *Security and Privacy Controls for Info Systems and Organizations*) [23]. For physical objects, IoT cybersecurity will be important as digital twin technology relies upon thorough instrumentation. IoT cybersecurity guidance can be found at the NIST Cybersecurity for IoT Program [24]. It will be important to ensure the cybersecurity of data in transit from IoT devices to the central digital twin definition repository. Public and standardized encryption algorithms should be used since proprietary encryption schemes can be weak and lack thorough vetting. The digital twin definition, 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 definition. 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 the digital twin system needs 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 as well as the hardware maintaining the digital twin facsimile.

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 will enable digital twin programs to work with any number of digital twin definitions, all of which will have differing sensitivities to faults and failures.

Lastly, the digital twin system (which covers the instrumentation, control/data channels, digital twin definition, and visualization/representation mechanisms) needs to be properly authorized by the appropriate organization 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, Privacy Framework) [22].

One could argue that, in a specialized, secure environment, it is not necessary to have this level of cybersecurity in place. However, even the most secure networks usually have some connections to the outside world, even if 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

This section considers a set of 14 trust considerations that may need to be addressed to enhance the usefulness of digital twin technology. It is not directly focused on risk assessment and risk mitigation but rather on trust. That is, will digital twin technology provide the desired operational functionality with an acceptable level of quality? Answering this question begins with an understanding of trust. Here, trust is the probability that the intended behavior and the actual behavior are equivalent given a fixed context, fixed environment, and fixed point in time. Trust is viewed as a level of confidence. In this section, trust is considered at several levels: 1) Is the digital twin functionally equivalent to the physical object? 2) Can a specific digital twin be composed with another digital twin? 3) Is enough information available about the environment and context of the physical object? 4) Can digital twin technology be standardized to the point where certification of digital twins is possible?

1. **Digital Twin Creation Ordering:** The point in time at which a digital twin is created will have an impact on the correctness of the digital twin. For example, is it created before the physical object is created, or is it reverse-engineered from the physical entity (that it is intended to mirror)? Both approaches are valid. However, 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. A good analogy here is commercial off-the-shelf (COTS) software. Such products are black boxes—the source code is unavailable to the customer or integrator and, thus, hides internal syntax. For digital twin, this is a trust consideration.
2. **Temporal:** The digital twin paradigm has an implied temporal component to it, particularly since it deals with physical objects, and physical objects are bound by time. Hardware reliability theory and modeling states that physical objects, even when idle, suffer from levels of decay over time. For example, if a car has not been turned on for years, it is likely that the battery will be dead, and the car will not start. Physical objects will degrade and fatigue over time after usage. However, a digital twin will not degrade or fatigue over time. Therefore, at some point the physical and digital twins will be in conflict on some level. 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:** The digital twin paradigm 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 this, it will be difficult to determine if the physical object is “fit for purpose” since purpose

implies environment and context. Unknown environmental influences have 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" (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:** The digital twin paradigm has an interesting relationship to mass production. 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. Not all bulbs produced will be usable, and for those that are usable there will still be small (possibly microscopic) distinctions between individual bulbs. These small distinctions may impact the lifetime of a specific bulb. The packaging on a set of light bulbs 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 also 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:** The digital twin paradigm needs a means to determine functional equivalence between the digital twin and the physical object. Without this, trust is suspect. If the digital twin is an executable specification, then for the inputs that it is fed, it should produce the same outputs that the physical object produces for the same input data. If this does not occur, then functional equivalence has not been achieved. This could occur for a variety of factors, such as decay and 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. **Composability and Complexity:** There is a trust consideration regarding the size and complexity of the digital twin for its physical object. 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, and each component has a digital twin. 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 should be useful to prune extraneous information contained in a digital twin since standards can define required interconnects between components of a domain enabling the composition to be modelled and tested. One approach might be separating classes of information into categories, such

as "need to know" or "extraneous."

7. **Instrumentation and Monitoring:** Instrumentation of a digital twin is a beneficial and unique advantage that digital twins offer. 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]; this is not often easy and can be more art than science. Second, how many probes to inject is also a consideration. As shown in real-time systems, probes can slow down performance and timing. This 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 collecting raw data and not 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]. If vendors misuse standardized formats for the digital twins of their components, composing digital twins from different component vendors may not be achievable [4]. This is a consideration for trusting composed digital twins.
9. **Non-functional Requirements:** A trust consideration for systems composed of many components deals with quality attributes often referred to as "ilities." This also applies to digital twin technology. 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") typically state what level of quality the system shall exhibit both for the functional and negative requirements. "ilities" apply to both "things" and the systems they are built into. It is unclear how many "ilities" there are, though examples 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]. 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:** If the accuracy of a digital twin is questionable, or even found to be faulty, then trust is an issue. For software, faulty specifications lead to faulty designs that lead to faulty implementations. In digital twin technology, the degree to which the digital twin is correct is a trust consideration. It begs the question as to whether it might be prudent to have 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. To address this, 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 digital twins 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 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, consideration should also be given for cases of misuse.

12. **Certification:** Certification usually occurs in two different ways [31]. One type certifies the process used to develop, while the other certifies the final artifact that comes from that process. These two types of certification are distinct [3][32][33][34][35][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. For example, the pharmaceutical industry has illuminated the problem of information overload. Most prescription drugs come with warnings concerning who can take them based on sex, age, underlying conditions, negative drug interactions, and other factors. Most drugs come with disclaimers about negative side effects and when to discontinue use. This information is made available to patients, doctors, pharmacists, and other medical providers. The problem stems from the vast amount of information known about a drug and the vaster amount of unknown information about a drug at time t that will not be known until time $t+1$. Further, much of the information is only understandable by medical experts but is vital to determine a drug's fitness for purpose. The trust consideration here for digital twin technology is how much of this information can be provided in a digital twin description without overloading a twin 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 (cascade) during execution [38][39]. The digital twin paradigm experiences this trust consideration, particularly when different twins representing different physical objects are composed. This may, perhaps, suggest that twins should be wrapped with pre-conditions and post-conditions to determine if the output from one twin will be acceptable as input to another twin.

14. **Counterfeiting:** It is possible that a digital twin could be tampered with or counterfeited. There are schemes that could be used to protect against this. Digital twins could be hashed, and the hash posted to a public web page; users of a digital twin could hash their copy and compare it against the hash on the public web page. This said, web pages and other similar publicly accessible repositories can be hacked. To enhance trust here one could use a blockchain and post a digital twin hash publicly in an immutable data structure (it could never be changed even by malicious attackers). In these ways modifications to digital twin files could be discovered. Alternatively, identical copies of a digital twin could be stored in separate locations (e.g., in offline backups).

9 Conclusions

Digital twin is an emerging area of research and standardization. At the same time its core elements of modeling and simulation are already very mature and widely used. Other significant components, such as virtual reality, 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 twins and what promise this technology holds. In this work we attempted to provide this clarity. We provided a detailed definition of digital twins, the motivation and vision for their use, common low-level operations, usage scenarios, and example use cases.

We also focused on technical considerations with the cybersecurity and trust of digital twins. We analyzed novel cybersecurity challenges arising from the use of digital twin architectures and then looked at the traditional cybersecurity challenges that apply. We evaluated trust issues and what a lack of trust and standards can do to digital twin functionality and quality. And we mapped our evaluations where appropriate to other NIST cybersecurity guidance.

It is our hope that our documentation of a definition and vision for digital twins along with an evaluation of cybersecurity and trust considerations will be useful in progressing this technology. In particular, we hope that standards developers and digital twin implementers will use this document to ensure the secure and trustworthy development of digital twin standards and architectures.

References

[1] The Definition of a Digital Twin. https://www.digitaltwinconsortium.org/hot-topics/the-definition-of-a-digital-twin.htm. (accessed 2021).

[2] Merriam-Webster online dictionary of American English. http://www.m-w.com (accessed 2021).

[3] Jeffrey M. Voas, George Hurlburt. “Third Party Software’s Trust Quagmire.” IEEE Computer, Volume 48 Issue 12:80-87.

[4] Jeffrey M. Voas, Phillip A. Laplante. July 2007. “Standards Confusion and Harmonization.” IEEE Computer Volume 40 Issue 7: 94-96.

[5] Piroumian, Vartan. January 2021. “Digital Twins: Universal Interoperability for the Digital Age.” IEEE Computer Volume 54, Number 1: 61-69.

[6] HTML 5.2, W3C Recommendation, 17 December 2017, superseded 28 January 2021. https://www.w3.org/TR/html52/ (accessed 2021).

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

[8] 3D PDF Consortium. “PDF in Manufacturing.” 2020. pp 22-26.

[9] Adele Goldberg. Smalltalk-80: The Interactive Programming Environment. Reading, MA. Addison-Wesley. 1983.

[10] Edward Yourdon, Larry Constantine. Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Upper Saddle River, NJ. Prentice-Hall. 1979.

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

[12] Robert L. Woods, Kent L. Lawrence. Modeling and Simulation of Dynamic Systems. Englewood Cliffs, NJ. Prentice Hall. 1997.

[13] “Internet Streaming: What It Is and How It Works.” 2021. https://www.lifewire.com/internet-streaming-how-it-works-1999513 (accessed 2021).

[14] “What Is Streaming? How Video Streaming Works.” Sam Costello. (accessed 2021). https://www.cloudflare.com/learning/video/what-is-streaming/

[15] John C. Reynolds. Theories of Programming Languages. Cambridge. Cambridge University Press. 1998.

[16] Robert Scheifler, James Gettys, Jim Flowers, David Rosenthal. X Window System, The

Complete Reference to Xlib, X Protocol, ICCCM, XLFD. Digital Press. 1992.

[17] Catuscia Palamidessi, Hugh Glaser, Karl Meinke. Principles of Declarative Programming: 10th Annual Symposium PLILP'98. Springer. 1998.

[18] Roberto Minerva, Gyu Myoung Lee, Noël Crespi. Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models. Proceedings of the IEEE Volume 108, Number 10. October 2020.

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

[20] NIST Risk Management Framework. https://csrc.nist.gov/projects/risk-management (accessed 2021).

[21] NIST Cybersecurity Framework. https://www.nist.gov/cyberframework (accessed 2021).

[22] NIST Privacy Framework. https://www.nist.gov/privacy-framework (accessed 2021).

[23] NIST, "Security and Privacy Controls for Information Systems and Organizations". 2020. https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final (accessed 2021).

[24] NIST Cybersecurity for IoT Program. https://www.nist.gov/programs-projects/nist-cybersecurity-iot-program (accessed 2021).

[25] Alper Kerman, Oliver Borchert, Scott Rose, Eileen Division, Allen Tan, "Implementing a Zero Trust Architecture" https://www.nccoe.nist.gov/library/implementing-zero-trust-architecture (accessed 2021).

[26] A. Stavrou & J. Voas, "Verified time," IEEE Computer, Vol. 50, 2017.

[27] J. Voas, "Software's secret sauce: The `ilities'," Quality Time Column, IEEE Software, 21(6), pp. 2-3, November 2004.

[28] J. Voas & K. Miller. "Putting Assertions in Their Place," Proceedings of the International Symposium on Software Reliability Engineering, November 1994, Monterey, CA

[29] J. Voas, Networks of Things, NIST Special Publication (SP) 800-183, National Institute of Standards and Technology, Gaithersburg, Maryland, 2016.

[30] N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation, Liming Chen; Avizienis, A., Fault-Tolerant Computing, 1995, ' Highlights from Twenty-Five Years'., Twenty-Fifth International Symposium on, Vol., Iss., 27-30 Jun 1995.

[31] J. Voas, "The Software Quality Certification Triangle," Crosstalk, 11(11), pp. 12-14, November 1998.

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

June 1998.

[33] J. Voas, “Certifying software for high assurance environments,” IEEE Software, 16(4), pp. 48-54, July 1999.

[34] J. Voas & J. Payne, “Dependability certification of software components,” Journal of Systems and Software, Vol. 52, p. 165-172, 2000.

[235 J. Voas, “Toward a Usage-Based Software Certification Process,” IEEE Computer, 33(8), pp. 32-37, August 2000.

[36] J. Voas & P. Laplante, “The IoT Blame Game,” IEEE Computer, 2017.

[37] J. Voas & P. Laplante, “IoT’s certification quagmire,” IEEE Computer, April 2018.

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

[39] J. Voas, F. Charron, & K. Miller, “Tolerant Software Interfaces: Can COTS-based systems be trusted without them?” Proceedings of the 15th International Conference on Computer Safety, Reliability and Security (SAFECOMP’96), Springer-Verlag, pp. 126-135, October 1996.

Appendix A—Glossary

digital twin	The digital representation of a real-world entity, concept, or notion, either physical or perceived.

Appendix B—Acronyms

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

2D	Two dimensional
3D	Three dimensional
AI	Artificial Intelligence
AR	Augmented Reality
A/V	Audio/Visual
CNC	Computer Numerical Control
COTS	Commercial Off-the-Shelf
DT	Digital Twin
GUI	Graphical User Interface
HTML	HyperText Markup Language
IC	Integrated Circuit
IoT	Internet of Things
IT	Information Technology
MVC	Model-view-controller
NIST	National Institute of Standards and Technology
PCB	Printed Circuit Board
PDF	Portable Document Format
SDO	Standards Developing Organization
UAV	Unmanned Aerial Vehicle
UI	User Interface
VR	Virtual Reality
WYSIWYG	What-you-see-is-what-you-get

WiFi	Wireless Fidelity[1]
XML	eXtensible Markup Language

[1] A family of wireless network protocols.