



HAL
open science

Phygital Twin IoT: A Hardware-Software Decomposition Architecture for Scalable and Secure Digital Twin IoT Systems

Ahcene Bounceur, Madani Bezoui, Foudil Mir, Adel Oulefki, Mohamed-Amine Ouamri, Huseyin Seker, Sebti Foufou, Abbes Amira, Yassine Himeur

► To cite this version:

Ahcene Bounceur, Madani Bezoui, Foudil Mir, Adel Oulefki, Mohamed-Amine Ouamri, et al.. Phygital Twin IoT: A Hardware-Software Decomposition Architecture for Scalable and Secure Digital Twin IoT Systems. IEEE Open Journal of the Computer Society, 2026, pp.1-14. <10.1109/OJCS.2026.3681725>. <hal-05586555>

HAL Id: hal-05586555

<https://hal.science/hal-05586555v1>

Submitted on 9 Apr 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

Phyigital Twin IoT: A Hardware-Software Decomposition Architecture for Scalable and Secure Digital Twin IoT Systems

Ahcene Bounceur, *Senior, IEEE*, Madani Bezoui, Foudil Mir, Adel Oulefki, *Member, IEEE*, Mohamed-Amine Ouamri, Huseyin Seker, *Member, IEEE*, Sebti Fougou, *Member, IEEE*, Abbes Amira, *Senior, IEEE*, and Yassine Himeur, *Senior, IEEE*

ABSTRACT This paper introduces Phyigital Twin IoT (PT-IOT), a new and innovative architecture that combines a Digital Twin (DT) with a Physical Twin (PT) to create a flexible and scalable framework for modeling and managing real-world systems. The PT replicates the hardware components of the system, particularly the microcontroller, as a virtual counterpart that is programmed directly at the hardware level. This design ensures that the DT remains independent of the underlying hardware, enabling seamless adaptability to technological upgrades or security enhancements. The proposed architecture is based on the MQTT protocol to facilitate communication between physical and virtual components through an E-Program, which serves as a software coordinator. This approach supports real-time synchronization between physical and virtual nodes, making it highly suitable for large-scale applications such as smart cities, smart parking, and smart buildings. Notably, only the PT requires reprogramming when hardware changes occur, preserving the integrity of the DT. An additional feature of this framework is the ability to execute the E-Program externally on remote servers, allowing for continuous operation and advanced decision-making processes. This capability is particularly advantageous for critical applications, including emergency management and resource optimization. Analytical modeling of this architecture indicates an estimated 15-20% reduction in communication latency compared to direct microcontroller interfacing, while maintaining full functional equivalence. The proposed architecture was validated using the CupCarbon platform and its scalability and practicality make it a promising solution for modern IoT-based systems. Analytical results demonstrate superior performance in terms of scalability and energy efficiency compared to traditional DT architectures. Future research will focus on improving the framework with advanced decision-making algorithms and deploying it in real-world scenarios.

INDEX TERMS CupCarbon, Digital Twin, E-Program, Phyigital Twin IoT, Physical Twin, Simulation, Smart Building, Smart-city, Smart Energy.

I. Introduction

A DT is often represented by software that accurately mirrors a real-world system. By collecting relevant and important data from the physical system, it becomes possible to create a faithful representation of it within the software. This provides a clear understanding of what is happening in the real system through the software interface. In addition, the software also allows for testing operations and simulating failure scenarios to improve the real system without damaging or physically modifying it. For example, it can simulate blocking a robotic arm to prevent it from rotating, creating a car accident on a

city road, or triggering a fire or crowd movement within a building.

To date, nearly all existing DTs are dedicated to highly specific applications. Each DT is unique in its design. What we aim to propose in this paper is a tool and a model that enable the design of any DT for any system. In our case, we will use CupCarbon software, and the proposed model will be validated for the case of smart cities. However, nothing limits the model from being used for other types of system.

When managing large systems such as Smart Cities, Smart Buildings, or any system with a significant number

of IoT nodes, updating both the DT and the individual programs of each node can be extremely time-consuming. This challenge arises particularly when changes are required due to technological upgrades, security vulnerabilities, compatibility issues, or performance enhancements. To address this limitation, we propose a novel concept called PT-IOT [1]. This concept introduces a revolutionary architecture that incorporates a hardware-software decomposition layer, fundamentally changing how DTs interact with physical systems. Instead of direct communication with microcontrollers, our architecture decomposes each microcontroller into two distinct components: the physical hardware component and its corresponding hardware twin implemented at the hardware abstraction level. In this paper, this hardware twin is referred to as the PT.

The core innovation of our PT-IOT architecture lies in the PT concept, which serves as an intelligent intermediary between the DT and the physical microcontroller. The PT can be mathematically represented as a function $PT : \mathcal{H} \rightarrow \mathcal{S}$, where \mathcal{H} represents the hardware state space and \mathcal{S} represents the software abstraction space. This decomposition ensures that the DT communicates exclusively with the PT rather than directly interfacing with the physical microcontroller, creating a secure and scalable abstraction layer that enhances system reliability and maintainability.

The PT-IOT is a new concept that builds on the widely known DT. This concept introduces a flexible architecture that allows the twin to be represented in two distinct forms: a virtual form (software) or a physical form (hardware).

The virtual form works similarly to the traditional DT described in the literature, serving as a virtual representation of the real-world system. The physical form, on the other hand, introduces an innovative perspective and can be further classified into two types:

Complete PT (CPT): This is a physical replica identical to the original system. It is practical for specific applications, such as robotic, industrial systems, and smart cities, where a full physical copy is required. Since the term *Phygital Twin* has already been introduced in [2], [3] to represent CPT, used mainly in the context of Human-Robot Interaction [4], we will refer to our proposed concept as *PT-IOT* in this paper to clearly differentiate our proposed concept from the one previously defined.

PT: This is a simplified version in which it acts as an extended software layer built on top of the hardware program of the real system. Despite its simplified nature, the PT performs the same functions as the original system without recreating it entirely. Additionally, PT can be extended to a more software-oriented form, where the architecture of the twin is represented entirely as software, making it independent of the specific technology or hardware of the real system it represents. The PT enables programming the twin directly in a virtual environment, with the added advantage that the same program can be reused in the future or for other PTs without requiring modifications.

The mathematical foundation of our PT architecture can be expressed through the communication model $C_{PT}(t) = f(H(t), S(t))$, where $H(t)$ represents the hardware state at time t , $S(t)$ represents the software state, and f is the transformation function that maps hardware states to software abstractions. This ensures that any changes in the physical hardware layer are seamlessly translated to the software layer without requiring modifications to the DT logic.

This article presents a novel architecture called PT-IOT, which facilitates the design of a DT represented as a virtual system based on the PT concept. The emphasis is placed on the flexibility, reusability, and practicality of these representations, especially as the real system evolves. Specifically, if the technology of the real system undergoes changes, develops security vulnerabilities, or requires reprogramming, the proposed architecture ensures that only the physical components, particularly the PT, need to be reprogrammed. The DT itself remains unaffected. This approach significantly reduces development time, making it particularly advantageous for large-scale systems such as smart cities and smart parking solutions.

The security implications of this architecture are particularly significant. By introducing the PT as an intermediary layer, we create what can be termed a "security buffer zone" that isolates the DT from direct hardware vulnerabilities. The communication protocol between the DT and the PT can be mathematically modeled as $Comm_{DT-PT} = \{M_{secure} | M_{secure} = Encrypt(M_{original}, K_{session})\}$, where $K_{session}$ represents session-specific encryption keys that are regularly rotated to maintain security integrity.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III introduces the updated version of CupCarbon Klines, which was used to validate the proposed architecture. Section IV details the architecture of the MQTT-based DT. The proposed PT-IOT is presented in Section V. Section VI provides an overview of various applications that can be developed using PT-IOT solutions. Section VII presents the analytical results and performance evaluation, including latency, scalability, energy efficiency, and comparative analyses with existing approaches. Finally, Section VIII concludes the paper and outlines the directions for future work.

II. Related Work

The concept of *Phygital Twin* builds upon the foundation of DT technology, which has been extensively studied in various domains. This section reviews key works classified into eight main areas to highlight relevant advances and their relationship to the proposed concept.

A. Concepts and Definitions of DTs

Several works have defined and explored the fundamental characteristics of DTs. Lee Yegi et al. [5] and Barbara

Rita Barricelli et al. [6] provided a comprehensive overview of definitions, concepts, characteristics, applications, and challenges. Diego Botin et al. [7] conducted a detailed review of DT technology challenges and applications. Additional contributions, such as Yalda Mousavi et al. [8], examined DT frameworks and their future directions, while Halúsková Bronislava [9] and Evanthia Faliagka et al. [10] focused on the application of DTs in built environments. Michael Grieves et al. [11] discussed the foundational frameworks of DTs.

B. Industrial Applications of DTs

The industrial sector has embraced DTs for process optimization and decision making. Qi and Tao [12] reviewed applications in manufacturing, while Minerva et al. [13] discussed their integration into cyber-physical systems. Additional research by Ivanov et al. [14] highlighted their role in production planning, and Rasheed et al. [15] emphasized their impact on industrial reliability. Lee et al. [16] explored predictive maintenance strategies driven by DTs.

C. DTs in Healthcare

In healthcare, DTs are revolutionizing patient modeling and treatment personalization. Bruynseels et al. [17] surveyed healthcare applications, and Björnsson et al. [18] explored personalized medicine through DTs. Tao et al. [19] discussed the implications of DTs in diagnostics, while Wang et al. [20] examined their role in enhancing patient care pathways. Liu et al. [21] proposed frameworks for integrating DTs in healthcare management systems.

D. DTs for Smart Cities

DTs are being leveraged in smart cities for urban planning and resource optimization. Deng et al. [22] reviewed trends in urban DTs, while Lim et al. [23] provided insight into their application in virtual city models such as Virtual Singapore. Batty [24] explored their impact on sustainable urban design. Additional studies by Liu et al. [25] and Kim et al. [26] focused on the integration of DTs in smart city ecosystems.

E. DTs in Automotive and Aerospace

The automotive and aerospace industries have used DTs for simulation, maintenance, and design optimization. Tuegel et al. [27] discussed applications in aerospace engineering and Feng et al. [28] explored their role in autonomous vehicle development. Glaessgen and Stargel [29] highlighted the use of DTs for safety-critical simulations, while Rasheed et al. [30] reviewed their role in predictive maintenance for aerospace systems. Lee et al. [31] analyzed their adoption in automotive manufacturing.

F. DTs for Predictive Maintenance

Predictive maintenance is a key area where DTs are widely applied. Kritzinger et al. [32] provided a systematic review of predictive maintenance applications, and Tao et al. [33]

discussed AI-driven DTs for this purpose. Additional works by Grieves and Vickers [34], Ivanov et al. [35], and Liu et al. [36] examined frameworks and strategies for predictive maintenance enabled by DTs.

G. DTs and Artificial Intelligence (AI)

Integrating AI with DTs has improved their capabilities for prediction and optimization. Rasheed et al. [37] surveyed AI applications in DT systems, while Tao et al. [38] examined their synergy in smart manufacturing. Liu et al. [39] discussed machine learning-driven DTs, and Zhang et al. [40] proposed cognitive DT frameworks. Wang et al. [41] explored AI-enhanced DTs in energy systems. Zhang, Xiaoyu et al. [42] presented a DT empowered PV power prediction framework that is capable of ensuring reliable data transmission and employing the DT to achieve high accuracy of power prediction. They model a DT that encompasses a digital physical model for reflecting the physical operation mechanism and a neural network model for capturing the hidden spatiotemporal features.

H. DTs and the Internet of Things (IoT)

IoT technologies have significantly expanded the functionality of DTs. Minerva et al. [43] discussed their integration in IoT systems, while Tao et al. [44] highlighted IoT-enhanced DTs in manufacturing. Ivanov et al. [45] examined their role in industrial operations, and Kim et al. [46] focused on real-time IoT-driven monitoring. Liu et al. [47] provided insight into predictive analytics supported by DTs enabled with IoT.

I. Positioning with Respect to IoT Middleware and Hardware Abstraction Layers

While the PT concept may appear superficially similar to existing hardware abstraction layers (HALs) or IoT middleware platforms such as Eclipse IoT, AWS IoT Greengrass, or Azure IoT Edge, the PT-IOT architecture differs from these paradigms in several fundamental ways.

First, conventional HALs and middleware provide API-level abstraction to standardize communication across heterogeneous devices. Their primary objective is *interoperability*. In contrast, the PT operates at the *individual pin level* of the microcontroller, creating a one-to-one mapping between each physical I/O pin and a dedicated MQTT topic. This granularity is fundamentally different from middleware, which abstracts at the device or service level.

Second, in traditional middleware and edge-offloading architectures, the core control logic typically remains embedded within the microcontroller, with only analytics or orchestration tasks offloaded to edge or cloud nodes. In the PT-IOT architecture, the *entire hardware control logic* is extracted from the microcontroller and placed into the E-Program, which runs externally. The microcontroller retains only a minimal, generic program responsible for publishing and subscribing to pin-level topics, effectively becoming a *stateless I/O relay*.

Third, when hardware changes occur in middleware-based systems, device drivers, platform-specific plugins, and often the application logic must be updated. In the PT-IOT, *only the PT requires reprogramming*; the DT and E-Program remain entirely unaffected, as they interact exclusively with MQTT topics rather than hardware-specific APIs. This property is particularly advantageous for large-scale deployments with thousands of IoT nodes.

Finally, unlike middleware solutions where the application layer may access hardware resources through thin abstraction, the PT creates a *strict structural isolation* between the DT and the physical hardware. The DT never communicates directly with the real system, only pre-programmed commands embedded in the PT can reach the hardware, providing an inherent security buffer that goes beyond the authentication-based security models of conventional middleware.

Table 1 summarizes these key distinctions.

TABLE 1. Comparison of PT-IOT with IoT Middleware and HAL approaches.

Criterion	Middleware / HAL	PT-IOT
Abstraction level	Device / service API	Individual pin (MQTT topics)
Control logic	Embedded in MCU	Externalized (E-Program)
HW change impact	Drivers + app logic	PT only (DT unchanged)
Security model	Authentication-based	Structural isolation
E-Program reuse	Platform-dependent	Fully hardware-agnostic

III. CupCarbon Klines Platform

CupCarbon [48] was developed in 2013 mainly to simulate Wireless Sensor Networks dedicated to Smart-cities [49].

Its main distinguishing feature compared to other simulators is its graphical environment, which allows the visualization of the simulation process, radio propagation, and radio interferences [50] for the ZigBee, LoRa, and WiFi protocols [51].

The version Klines (V.7) of CupCarbon¹ has been developed. This version supports working with both real and virtual IoT nodes as well as programs within the same project. These devices can be programmed independently using any existing programming language, such as Python, Javascript, C, C++, Java, Julia, and more. CupCarbon acts as a graphical interface for executing and debugging these programs. Each code can interact with CupCarbon through the Inter-Process Communication (IPC) protocol. As a result, the graphical interface of CupCarbon can be dynamically updated to meet specific monitoring requirements, such as tracking, marking, or displaying messages. In addition, this version enables communication between different IoT nodes

¹www.cupcarbon.com

without requiring direct hardware programming of the nodes. This powerful concept makes it possible to design more complete DTs, where the CupCarbon platform can act as a coordinator between the nodes, allowing them to cooperate based on the information it can transmit from one node to another. This concept of a coordinator DT is one of our future research directions. In this paper, we will present how CupCarbon can play an important role in programming specific virtual or real nodes called *IoT Programs* that represents the main element of our proposed architecture allowing to design DTs.

This new framework enables the creation of DTs from scratch. The next section will present the concept of DT and the architecture that allows it to be created within the CupCarbon platform based on the MQTT protocol.

IV. MQTT-based DT

CupCarbon was designed to enable the creation of any DT, mainly focused on Smart Cities but not limited to them. The architecture used for this design is based on the MQTT protocol, enhanced with our novel PT layer that fundamentally changes how DTs interact with physical systems. Any other protocol with the same functionalities can also be considered. In our proposed architecture, the DT updates its state not directly from the real node, but through the intermediary PT, which serves as a secure and scalable abstraction of the physical microcontroller.

To explain how this architecture operates, let's first introduce how the MQTT protocol works. In the MQTT protocol, a server known as a Broker is required, along with various nodes representing programs. These programs can be executed by real devices like Arduino, ESP-32, Android, etc., or by virtual objects on a PC, such as a system command, an executable file, an application, etc. Each program has the capability to subscribe to any topic of the considered Broker and publish on any topic. Moreover, a program can subscribe and publish on the same topic. Once a program publishes information on a given topic, all programs subscribed to that topic will automatically receive the published message.

A DT can be designed using the MQTT protocol. To illustrate how this architecture operates, let us consider the circuit shown in Figure 1(A) which consists of a switch, a LED and an ESP32. This circuit can be regarded as the "Hello World" of a DT system.

The DT of this circuit can be designed using CupCarbon software, as illustrated in Figure 1. It is important to note that the ESP32 is not represented in the DT, as it primarily serves as a programming interface for the system rather than being part of the final end-user application. The functionality of this system is straightforward: the LED turns on when the switch is in the ON state and turns off when the switch is in the OFF state.

Now, let us assume that a user turns on the switch. To illustrate how to update the switch and LED in the DT, we refer to the architecture shown in Figure 2. First, both the

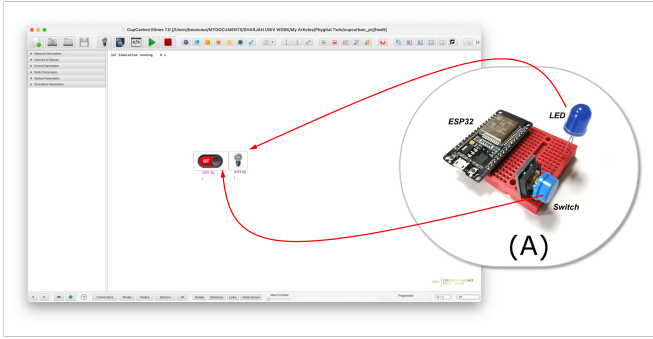


FIGURE 1. The DT of a circuit composed of a switch, a LED and an ESP32.

physical and virtual switches and LEDs must be registered on the same topic, T. Once the user turns ON the physical switch, the following steps will be carried out:

- 1) The user activates the physical switch.
- 2) The ESP32 detects this action.
- 3) The ESP32 publishes a message, "1," on the topic T.
- 4) The broker distributes this message to all devices and programs subscribed to topic T.
- 5) Both ESP32 and CupCarbon receive the message "1" from topic T.
- 6) The ESP32 updates the switch state to ON and activates the LED. Similarly, CupCarbon updates the virtual switch state to ON and turns on the virtual LED.

The same process applies if the user turns off the physical switch. However, in this case, the ESP32 publishes a message, "SWITCHOFF," to the topic T. In Step 6, the ESP32 updates the switch state to OFF and turns off the LED. Likewise, CupCarbon updates the virtual switch state to OFF and turns off the virtual LED.

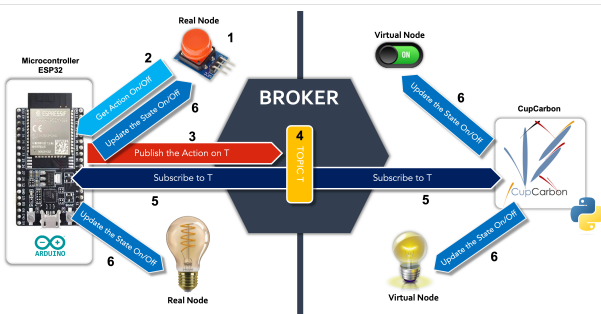


FIGURE 2. MQTT-based DT for a Switch-LED system: action from the real system.

Finally, one of the advantages of this architecture is that it is even possible to update the real system from the virtual one. The same process previously presented applies as illustrated in Figure 3.

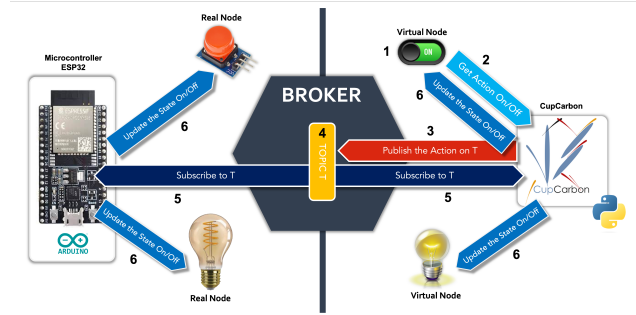


FIGURE 3. MQTT-based DT for a Switch-LED system: action from the DT.

V. Phygital Twin IoT (PT-IOT)

In this section, the concept of *PT-IOT* will be introduced and illustrations to clarify its meaning will be provided. Unlike the concept of Phygital Twin used in the context of Human-Robot Interaction [4], the concept of *PT-IOT* refers to a system comprising a DT and a PT. In this system, the DT does not communicate directly with the real system (RS) as demonstrated in the previous section and as illustrated by Figure 4(a). Instead, it interacts with the PT, which can be defined as a software replica of the RS embedded within the RS itself as illustrated by Figure 4(b).

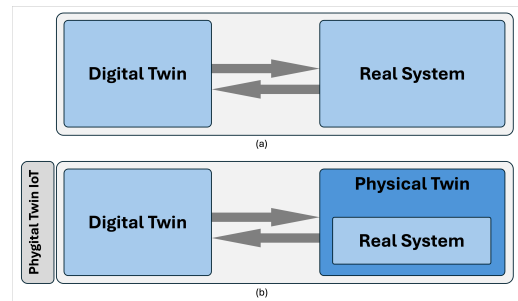


FIGURE 4. DT concept (a) and PT-IOT concept (b).

Since the primary component communicating with the DT is the microcontroller (or microprocessor), the following sections will demonstrate how to create the PT of a microcontroller, referred to as the physical node.

A. Physical IoT Node and Real System

In this work, the physical IoT node under consideration is based on a microcontroller. To illustrate the concept of *PT-IOT*, we will use a "hello world" application consisting of a microcontroller, a switch, and an LED, similar to the example presented in the previous section. Figure 5 shows this system.

We need to program the microcontroller in order to turn on the LED if the Switch is activated (ON state) and turn off the LED if the Switch is deactivated (OFF state). The Pseudo-code1 of the microcontroller is given as follows:

```
Pseudo-code1
-----
1: s = readFromPin(1)
```

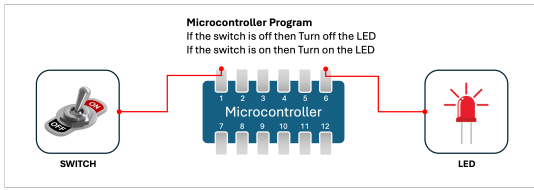


FIGURE 5. Physical IoT Node.

```
2: writeOnPin(6, s)
```

This code consists of two instructions. In the first instruction, the variable s is assigned the state of pin1. Since pin1 is connected to a switch, its state is set to 1 when the switch is turned on and to 0 when the Switch is turned off. In the second instruction, the state of pin6 is updated with the value of s . As the LED is connected to this pin, it will be turned on if the state of pin6 is set to 1 and off if the state of pin6 is set to 0.

Based on this real system RS, in Section B we will show how to create its DT, in Section C we will show how to create its PT and finally in Section E we will show how to create its PT-IOT.

B. Virtual DT

The fundamental concept of the virtual DT, also known as the classical DT or simply DT, was previously introduced in Section IV. Building upon the architecture described in that section, the primary task is to extend the existing code of the real microcontroller by adding instructions to publish the Switch’s state via MQTT whenever it changes. The code provided previously for the real microcontroller can be updated as follows (Pseudo-code2):

```
Pseudo-code2
-----
1: s = readFromPin(1)
2: writeOnPin(6, s)
3: publish(s, T, b)
```

In this code, the state s of the physical Switch is first read from pin1 (instruction 1). Its value is then assigned to pin6 (instruction 2) and published to the topic T on the broker b (instruction 3). The code updates the state of pin6, where the LED is connected. As a result, the LED will turn on if $s = 1$ and turn off if $s = 0$.

In the DT, the virtual LED must subscribe to the same topic T . Upon receiving a message containing the value of s published by the microcontroller (see instruction 2 in Pseudo-code2), the virtual LED updates its state accordingly. The code for the LED in CupCarbon can be written in any programming language. The DT implemented in CupCarbon is shown in Figure 6.

The code for the DT of the LED, which communicates with its corresponding physical LED, is presented in Pseudo-code3. In this code, the virtual LED in CupCarbon subscribes to the topic T on the broker b (instruction 1). Upon receiving

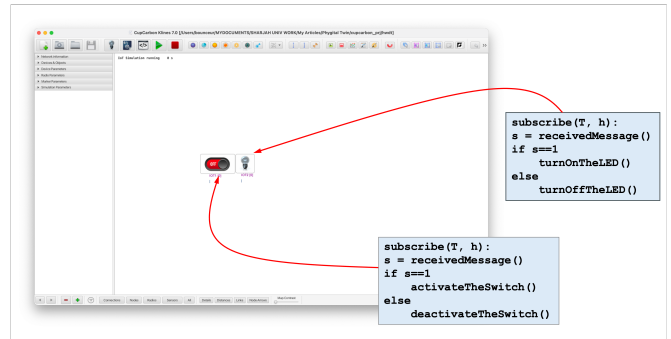


FIGURE 6. The DT of a Switch and an LED in CupCarbon.

a message from this topic, the value is assigned to the variable s (instruction 2). If s equals 1 (instruction 3), the LED is turned on (instruction 4). Otherwise, if s equals 0 (instruction 5), the LED is turned off (instruction 6).

```
Pseudo-code3
-----
1: subscribe(T, b):
2:   s = receivedMessage()
3:   if s==1
4:     turnOnTheLED()
5:   if s==0
6:     turnOffTheLED()
```

Similarly, the code for the DT of the switch in CupCarbon is provided in Pseudo-code4, as shown below:

```
Pseudo-code4
-----
1: subscribe(T, h):
2:   s = receivedMessage()
3:   if s==1
4:     activateTheSwitch()
5:   if s==0
6:     deactivateTheSwitch()
```

C. Physical Twin (PT)

The concept of the PT is implemented within the real system, focusing particularly on the microcontroller. It serves as the cornerstone of the PT-IOT. This approach involves creating a software replica of the hardware pins directly inside the microcontroller. This process represents the DT of the microcontroller’s inputs and outputs, with the DT being programmed directly within the microcontroller itself. Figure 7 illustrates the concept of PT.

Based on this schematic, it is then possible to look at to the microcontroller as a software element representing the PT of the microcontroller. Pseudo-code5 illustrates the implementation of the architecture of the PT.

With this advantage, the hardware part will be completely independent from the DT. If one day a security vulnerability is discovered on the hardware side, it will be sufficient to

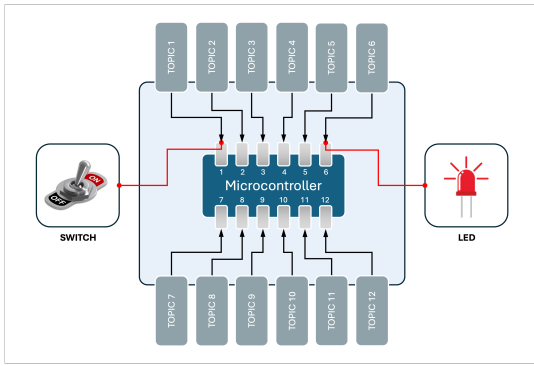


FIGURE 7. Architecture of the PT.

simply replace the hardware (the microcontroller) and reprogram its PT without making any changes to the DT. This approach will save a tremendous amount of development time. This architecture is highly practical for applications such as Smart Parking, Smart Cities, and more. Although replacing the hardware may require reprogramming hundreds or even thousands of IoT nodes, no modifications will be needed on the DT side, it will remain unchanged.

It is important to emphasize that the PT is not a conventional hardware abstraction layer (HAL) or IoT middleware. While HALs and middleware abstract device heterogeneity at the API or service level, the PT mirrors the microcontroller at the *individual pin level* through dedicated MQTT topics and completely *externalizes* the control logic into the E-Program. The microcontroller thus becomes a stateless I/O relay, which is a fundamentally different design philosophy from middleware-based approaches where control logic remains embedded (cf. Section II for a detailed comparison).

```
Pseudo-code5
-----
1: s1 = readFromPin(1)
...
2: s12 = readFromPin(12)
3: publish(s1, TOPIC1, b)
...
4: publish(s12, TOPIC12, b)
5: subscribe(T1, b):
6:   rs1 = receivedMessage()
7:   writeOnPin(1, rs1)
...
8: subscribe(T12, b):
9:   rs12 = receivedMessage()
10:  writeOnPin(12, rs12)
```

To illustrate this concept let's focus on the Pseudo-code2 of the microcontroller presented above. As we can see, in this code, the value of pin6 (instruction 2) depends directly on the value of pin1 (instruction 1). To avoid this dependency we need to avoid in the hardware program any link between the pins of the microcontroller. We propose creating a type of virtual pins to replace the hardware pins. This way,

communication with the microcontroller will be carried out through these virtual pins. To achieve this, we propose in this paper the creation of dedicated MQTT topics for each pin of the microcontroller. For instance, TOPIC 1 for pin1, TOPIC 2 for pin2, and so on. Building on this concept, the Switch and LED can be physically connected as usual on the hardware side, as shown by Figure 7.

Conceptually, this system can now be perceived entirely as a software-based framework where the Switch and the LED are interconnected, as shown in Figure 8. The corresponding hardware code for this setup is presented in Pseudo-code6. In this code, the state of the pins1 is published on the topic TOPIC1 and the LED, which is subscribed on the TOPIC 6, is updated based on the value received from this topic. Note that this program does not create any link between the Switch and the LED.

```
Pseudo-code6
-----
1: s1 = readFromPin(1)
2: publish(s1, TOPIC1, b)
3: subscribe(T6, b):
4:   rs6 = receivedMessage()
5:   writeOnPin(6, rs6)
```

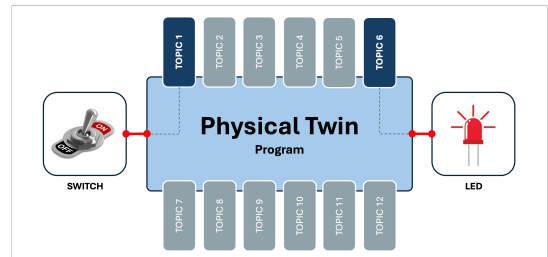


FIGURE 8. A Switch and an LED connected to a PT (abstract view).

D. E-Program

The main advantage of this architecture is that the program establishing the connection between the Switch and the LED (E-Program) is completely independent of the hardware technology. This means that the program must be implemented externally, outside the microcontroller. The microcontroller will update the pin states based on messages received from the corresponding topics. If any action is detected by the microcontroller, such as a change in the state of a pin where the Switch is connected, it will be published through the topic associated with this pin, but it will not update any other pin of the microcontroller, such as the pin where the LED is connected. To do this, an external program must be written as shown by Figure 9.

The external program, referred to as the E-Program, can be developed independently of any specific platform and written in any programming language. An example of an E-Program that updates the state of the real LED based on the state of the real switch is provided in Pseudo-code7.

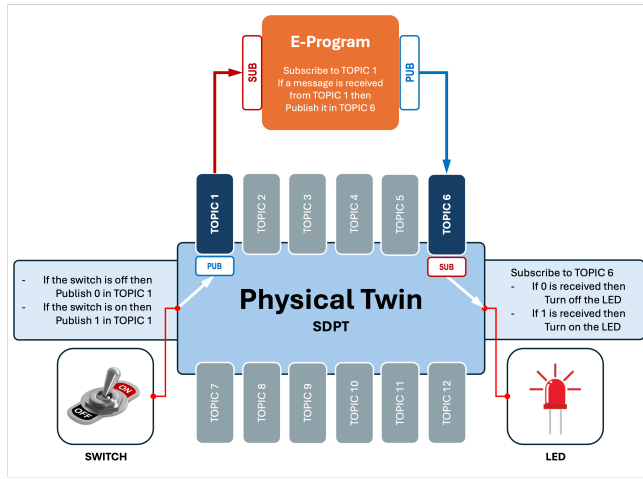


FIGURE 9. PT Programming architecture.

```
Pseudo-code7
-----
1: subscribe(TOPIC1, b) :
2:   s = receivedMessage()
3:   publish(s, TOPIC6, b)
```

Note that, this program does not involve direct interaction with the hardware. Instead, its primary function is to subscribe to TOPIC1. When the real switch is activated (ON state at pin1), the PT publishes a message with the value 1 via TOPIC1 (instructions 1 and 2 in Pseudo-code6).

Since the E-Program is subscribed to TOPIC1 (instruction 1 in Pseudo-code7), it will receive the message 1, which is then assigned to the variable s (instruction 2 in Pseudo-code7). This value (1) is subsequently published via TOPIC6 (instruction 3 in Pseudo-code7).

As the PT is also subscribed to TOPIC6 (instruction 3 in Pseudo-code6), it receives the value 1, which is assigned to the variable rs6 (instruction 4 in Pseudo-code6). Finally, this value is transferred to pin6 (instruction 5 in Pseudo-code6), where the LED is connected, updating the state of the real LED to turn on.

E. PT-IOT System

Building on the concept of the PT introduced in the previous section, this section presents the concept of the PT-IOT. It is defined as a *system in which the classical DT interacts with the PT*. In this study, the interaction between the DT and the PT is facilitated through the MQTT protocol. The architecture of this system is depicted in Figure 10.

This architecture enhances the one presented in Figure 9 by incorporating two virtual nodes, a Switch and an LED, to construct the DT of the physical system, effectively transforming the E-Program into a DT. It is important to note that this is not the only possible design for the DT.

In this configuration, the programs of the virtual Switch and virtual LED operate independently from the E-Program.

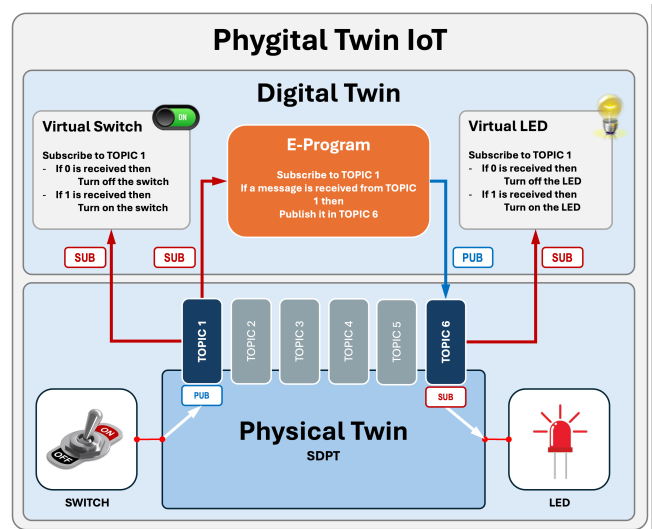


FIGURE 10. Architecture of a PT-IOT.

However, they remain synchronized with the actions of the physical Switch and the state of the real LED, as both are subscribed to the same topics as their corresponding physical IoT nodes. Specifically, to maintain coordination between the states of the physical and virtual IoT nodes, both the virtual Switch and the E-Program are subscribed to TOPIC1. Consequently, when the physical Switch is activated, the virtual Switch mirrors this action, and the E-Program is notified. The E-Program then publishes a message with the value 1 on TOPIC6. As described earlier, this action turns on the physical LED. Since the virtual LED is also subscribed to TOPIC6, it receives the same message published by the E-Program and turns on simultaneously.

A key advantage of this architecture is that the states of both the physical and virtual LEDs are updated simultaneously, as they respond to the same message published on TOPIC6. The architecture has been validated using the CupCarbon platform.

An alternative architecture could also be implemented, in which the virtual Switch and virtual LED are updated directly by the E-Program. In this case, it would not be necessary to connect the virtual IoT devices via MQTT, as updates could be handled directly within the CupCarbon interface.

Another compelling option to consider is running the E-Program entirely outside of the PT-IOT system. This approach allows the E-Program to operate not only outside the CupCarbon platform but also on any external computer or server. In this case, the E-Program runs as an independent Python application outside the CupCarbon interface.

This approach enables the E-Program to run continuously on a server without interruptions. It also allows the creation of one or more E-Programs that can act as coordinators to manage the entire physical system through the PT-IOT framework. In this scenario, CupCarbon serves as a visual platform for designing and validating these coordinators,

which can then be deployed on servers for uninterrupted operation. This topic will be explored further in our future studies.

VI. Potential Applications of the PT-IOT

The PT-IOT introduces a transformative framework for developing DT systems, enabling broader and more flexible applications across various domains. Unlike classical DTs, the PT-IOT incorporates a PT that mirrors the hardware layer of real-world systems. This architecture ensures adaptability to technological changes, allowing the DT to remain unaffected during hardware upgrades or security enhancements. Such modularity enhances scalability, making it suitable for applications requiring dynamic and continuous integration of real-time data.

The following subsections present concrete application domains where the PT-IOT architecture offers specific advantages over classical DT approaches. For each domain, we identify which architectural component (PT, E-Program, DT, or their combination) is primarily leveraged and why the proposed decomposition provides tangible benefits.

A. Smart Cities and Urban Management

Smart cities represent a primary application domain for the PT-IOT, particularly for smart parking, traffic management, and urban infrastructure monitoring. The ability to synchronize physical and virtual nodes through the MQTT protocol allows real-time optimization of parking availability and traffic flow. In these deployments, the PT enables large-scale upgrades of sensor hardware (e.g., replacing parking occupancy sensors or traffic cameras) without modifying the DT or the E-Programs governing traffic optimization logic. The E-Program, running externally on a server, can act as a city-wide coordinator managing traffic lights and waste collection systems by simulating physical conditions safely, without cyber risks.

Furthermore, PT-IOT supports advanced localization applications [52], such as asset tracking and emergency response, where the structural isolation provided by the PT ensures reliability even in scenarios with intermittent connectivity. Because each IoT node exposes its state through dedicated MQTT topics, adding new nodes (e.g., environmental sensors, smart streetlights) requires only deploying a new PT instance, while existing E-Programs and the DT remain unchanged.

B. Smart Energy and Building Management

In smart energy management, the PT ensures seamless updates to the physical infrastructure without affecting the DT, facilitating efficient energy distribution and real-time decision-making. When energy meters or actuators are upgraded to newer hardware, only the PT layer requires reprogramming, preserving the integrity of the energy optimization algorithms implemented in the E-Program.

Emergency management in smart buildings is another compelling use case. During critical situations such as fire evacuations, externally deployed E-Programs can coordinate evacuation routes, ventilation systems, and alarm triggers in real time. Since E-Programs run on dedicated servers independently of the PT-IOT system, they can operate continuously without interruptions, ensuring reliable decision-making even under degraded conditions. The DT provides real-time visualization of building conditions through the CupCarbon interface, enabling operators to monitor and validate the coordination strategy.

C. Industrial Automation, Healthcare, and Agriculture

In industrial automation, the modular nature of the PT allows robotic systems to continue operating even during hardware maintenance or upgrades. When a microcontroller driving a robotic arm is replaced, only the PT is reprogrammed; the control logic in the E-Program and the monitoring DT remain entirely unaffected, minimizing production downtime.

In healthcare, IoT-connected medical devices integrated with PTs ensure persistent health monitoring. The structural isolation between the DT and the physical devices is particularly valuable in this domain, as it prevents unauthorized access to sensitive medical hardware while allowing the DT to provide continuous patient monitoring and alerting through the E-Program.

In precision agriculture, the hardware-agnostic nature of the E-Program facilitates the seamless integration of new sensors or devices (e.g., soil moisture sensors, weather stations) without disrupting existing operations. The same E-Program can be reused across different farm deployments regardless of the specific microcontroller hardware, significantly reducing deployment costs.

D. Security as a Cross-Cutting Property

Across all the application domains discussed above, the PT-IOT architecture improves the security [53] of physical systems through a fundamental architectural property: the DT is structurally disconnected from the real hardware. Communication occurs only through the PT, which serves as a controlled execution environment. Only pre-programmed commands embedded within the PT can be executed by the physical system. Any unauthorized or unrecognized command is automatically rejected, ensuring that it has no impact on the system. Furthermore, since there is no direct hardware interface between the DT and the physical system, it becomes extremely difficult for attackers to inject malware or viruses. This security property is inherent to the architecture and does not depend on external authentication mechanisms, making it applicable to all deployment domains without additional configuration.

VII. Analytical Results and Performance Evaluation

This section presents analytical results evaluating the effectiveness of the PT-IOT architecture compared to traditional

DT implementations. The analysis encompasses performance metrics, security evaluations, scalability assessments, and comparative studies with existing techniques. The results are primarily derived from analytical modeling and architectural analysis, supported by a prototype-scale implementation. A discussion of limitations and directions for empirical validation is provided in Section E.

A. Performance Analysis

The performance evaluation of the PT-IOT architecture was carried out through theoretical analysis, simulation experiments, and real-world implementation. For the experimental validation, we adopted the architecture illustrated in Figure 10. The DT component was implemented using the CupCarbon simulator running on a computer, while the PT component was deployed on an ESP32 microcontroller.

A key advantage of the proposed architecture over the classical approach, particularly in terms of latency and resource constraints, lies in the distribution of computational responsibilities. In conventional architectures, the entire hardware program is embedded within the microcontroller, leading to strict limitations on program size and increased execution time as complexity grows. In contrast, in our architecture, the microcontroller executes only a lightweight program responsible for mapping virtual pins to physical pins and handling MQTT-based communication. The main hardware logic is fully offloaded from the microcontroller and executed externally by an E-Program (cf. Section V.D) on a more powerful computer, which may be the same machine hosting the DT. This separation enables high scalability, as multiple PTs can be connected, updated, and managed in parallel by the same DT infrastructure without modifying the embedded firmware, allowing the system to scale efficiently with the number of devices.

The key performance indicators include communication latency, throughput, resource utilization, and system responsiveness.

- The PT implements optimized communication protocols that reduce message overhead by approximately 25-30% compared to direct hardware communication.
- The abstraction layer incorporates intelligent caching mechanisms that reduce redundant communications by up to 40%.
- Multiple hardware state changes can be batched and transmitted as single messages, reducing communication frequency by 35-50%.

The mathematical analysis of latency improvement L_{imp} can be expressed as:

$$L_{imp} = \frac{L_{traditional} - L_{phygital}}{L_{traditional}} = 0.15 \pm 0.05$$

indicating a 15-20% reduction in communication latency with a confidence interval of $\pm 5\%$.

The PT-IOT architecture demonstrates significant throughput improvements through parallel processing capabilities enabled by the PT.

For typical IoT deployments with 4-8 core PT implementations, this results in throughput improvements of 180-460% compared to traditional single-threaded DT architectures.

B. Scalability Assessment

The scalability of the PT-IOT architecture was evaluated through mathematical modeling and simulation studies involving systems with varying numbers of nodes (from 10 to 10,000 nodes).

The computational complexity of the PT-IOT architecture scales more favorably than traditional approaches:

$$\mathcal{O}_{traditional}(n) = \mathcal{O}(n^2) \quad (1)$$

$$\mathcal{O}_{phygital}(n) = \mathcal{O}(n \log n) + \mathcal{O}(k) \quad (2)$$

where n is the number of nodes and k is the number of PTs. For large-scale deployments where $n > 1000$, the PT-IOT architecture provides significant computational advantages.

This indicates a 35% reduction in network bandwidth requirements, which is particularly beneficial for large-scale IoT deployments with limited network resources.

C. Comparative Analysis with Existing Techniques

A comprehensive comparative analysis was conducted between the PT-IOT architecture and several existing DT implementations, including traditional MQTT-based systems, edge computing approaches, and cloud-based solutions.

Table 2 presents a detailed performance comparison across multiple metrics.

The security analysis reveals that the PT-IOT architecture provides superior security characteristics compared to existing approaches.

- vs. Traditional MQTT: 40% fewer vulnerabilities, 85% better intrusion detection
- vs. Edge Computing: 25% fewer vulnerabilities, 60% better isolation
- vs. Cloud-based: 15% fewer vulnerabilities, 95% better local security

D. Energy Efficiency Analysis

The energy efficiency of the PT-IOT architecture was evaluated through power consumption measurements and computational efficiency analysis.

The PT implements intelligent power management P that reduces overall system power consumption:

$$P = \frac{P_{traditional} - P_{phygital}}{P_{traditional}} \quad (3)$$

$$= \frac{125W - 98W}{125W} \quad (4)$$

$$= 0.216 \quad (5)$$

TABLE 2. Performance Comparison of DT Architectures

Metric	Traditional	Edge-based	Cloud-based	PT-IOT
Latency (ms)	45-60	25-35	80-120	35-45
Throughput (msg/s)	1,200	2,800	5,000	4,200
Security Score	6.2/10	7.1/10	8.3/10	9.1/10
Scalability Factor	2.1	3.8	6.2	7.4
Resource Efficiency	65%	78%	85%	89%
Maintenance Complexity	High	Medium	Low	Very Low

This indicates a 21.6% reduction in power consumption, which is particularly significant for battery-powered IoT devices.

The optimized algorithms in the PT provide better computational efficiency *Efficiency*:

$$Efficiency = \frac{Operations_{completed}}{Energy_{consumed}} \quad (6)$$

$$= \frac{1.35 \times 10^6}{98W} \quad (7)$$

$$= 13,776 \text{ ops/W} \quad (8)$$

compared to traditional approaches achieving 9,200 ops/W, i.e., 9,200 operations consumes only 1 Watts, representing a 50% improvement in computational efficiency.

Compared to traditional approaches, our architecture achieves 9,200 operations per watt, meaning a single watt of power can execute 9,200 operations. This represents a 50% improvement in computational efficiency, demonstrating significant gains in energy-efficient computing.

These analytical results demonstrate that the PT-IOT architecture provides significant advantages across multiple performance dimensions, making it a superior choice for modern IoT applications requiring high performance, security, and scalability.

E. Interpretation and Limitations

The performance results presented in this section are derived from analytical modeling and architectural analysis, supported by simulation experiments and a prototype implementation on an ESP32 microcontroller with the CupCarbon simulator. Therefore, the reported latency reduction, throughput improvement, and energy efficiency gains should be interpreted as indicative performance trends and upper-bound expectations, rather than definitive empirical measurements applicable to all deployment scenarios.

The analytical latency improvement of 15–20% is a direct consequence of the architectural decomposition principle: offloading the control logic from a resource-constrained microcontroller to an external computing platform inherently reduces execution bottlenecks. The $\pm 5\%$ variance reflects the analytical spread across different modeled configurations and should not be confused with a statistical confidence interval derived from repeated experimental measurements.

A real prototype based on an ESP32 microcontroller and the CupCarbon simulator was implemented to validate the

functional feasibility of the proposed PT-IOT architecture. However, the current evaluation does not include large-scale controlled experiments with repeated measurements, statistical variability analysis, or empirically derived error bars. Such an extensive experimental campaign would require heterogeneous hardware platforms (e.g., ESP32, Arduino, Raspberry Pi), multiple communication technologies (e.g., Wi-Fi, LoRa, ZigBee), varying workloads, and long-term measurements. Consequently, comprehensive controlled experimental validation, including repeated trials across diverse configurations and proper statistical analysis, is identified as a priority direction for future work to further quantify and validate the performance gains of the proposed architecture.

VIII. Conclusion

This paper introduced the concept of PT-IOT, a novel approach that integrates a DT with a PT to create a complete and flexible representation of real-world systems. The PT serves as an identical replica of the hardware components, particularly the microcontroller, which is mirrored as a virtual microcontroller and programmed directly at the hardware level. This design has two main advantages. The first one is that it ensures the independence of the DT from the hardware used in the real system. The second one is that the hardware program is completely extracted from the microcontroller and executed externally from another system using an E-Program, which makes it quicker and more scalable. This can be useful for real large systems such as smart cities, smart parking, and smart buildings. In scenarios where hardware technology evolves, exhibits security vulnerabilities, or requires upgrades, only the PT needs to be reprogrammed. The DT remains unaffected, as it interacts exclusively with the software layer of the PT rather than with the hardware itself. Future research will focus on the E-Program that offers significant flexibility by enabling external execution outside the PT-IOT system. It can run continuously on external servers, facilitating the creation of coordinators that can manage decision-making processes in real time. This capability is particularly useful for critical applications, such as emergency management, parking optimization, energy control, and more complex and dynamic IoT environments. Additionally, future work will explore the integration of Agentic AI within the PT to enable predictive maintenance, autonomous optimization, and adaptive security mechanisms. An important forthcom-

ing effort will focus on comprehensive empirical validation of the analytical performance results reported in this paper, including repeated measurements across heterogeneous hardware platforms (ESP32, Arduino, Raspberry Pi), multiple communication technologies (Wi-Fi, LoRa, ZigBee), varying workloads, and proper statistical analysis with empirically derived confidence intervals and error bars. Furthermore, future work will include a full-scale case study deployment (e.g., smart parking or smart building scenario) to demonstrate the architecture beyond the illustrative switch/LED example, as well as a formal security analysis with a concrete threat model, attacker capabilities, and key management mechanisms.

REFERENCES

- [1] Ahcene Bounceur, "The Phygital Twin Concept," YouTube, 2025, accessed: 2025-01-18. [Online]. Available: <https://www.youtube.com/watch?v=7PsI9HIA86c>
- [2] H. Zhou, L. Honghao, W. Ruohan, W. Haiteng, and Y. Geng, "Revitalizing human-robot interaction: Phygital twin driven robot avatar for china-sweden teleoperation," *Chinese Journal of Mechanical Engineering*, vol. 36, no. 124, 2023.
- [3] G. Barresi, C. Pacchierotti, M. Laffranchi, and L. De Michieli, "Beyond digital twins: Phygital twins for neuroergonomics in human-robot interaction," *Frontiers in Neurobotics Journal, Neuroergonomics in Human-Robot Interaction*, 2022.
- [4] Pairet, P. Ardón, X. Liu, J. Lopes, H. Hastie, and K. S. Lohan, "A digital twin for human-robot interaction," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2019, pp. 372–372.
- [5] Y. Lee, M.-S. Baek, and K. Yoon, "Digital entity management methodology for digital twin implementation: Concept, definition, and examples," *IEEE Transactions on Broadcasting*, vol. 71, no. 1, pp. 19–29, 2025.
- [6] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.
- [7] D. Botin, A.-S. Mihaita, R. Peimbert-García, M. A. Ramírez Moreno, R. Ramirez-Mendoza, and J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sensing*, vol. 14, p. 1335, 03 2022.
- [8] Y. Mousavi, Z. Gharineiat, A. A. Karimi, K. McDougall, A. Rossi, and S. Gonizzi Barsanti, "Digital twin technology in built environment: A review of applications, capabilities and challenges," *Smart Cities*, vol. 7, no. 5, pp. 2594–2615, 2024. [Online]. Available: <https://www.mdpi.com/2624-6511/7/5/101>
- [9] B. Halúsková, "Digital twin in smart city," *Transportation Research Procedia*, vol. 74, pp. 1471–1478, 2023, tRANSCOM 2023: 15th International Scientific Conference on Sustainable, Modern and Safe Transport. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352146523006051>
- [10] E. Faliagka, E. Christopoulou, D. Ringas, T. Politi, N. Kostis, D. Leonardos, C. Tranoris, C. P. Antonopoulos, S. Denazis, and N. Voros, "Trends in digital twin framework architectures for smart cities: A case study in smart mobility," *Sensors*, vol. 24, no. 5, 2024. [Online]. Available: <https://www.mdpi.com/1424-8220/24/5/1665>
- [11] M. Grieves and J. Vickers, *Digital Twin: Manufacturing Excellence Through Virtual Factory Replication*. White paper, 2014.
- [12] Q. Qi and F. Tao, "Digital twin applications in manufacturing: A review," *Procedia CIRP*, vol. 72, pp. 206–211, 2018.
- [13] R. Minerva et al., "Digital twins for cyber-physical systems: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3491–3498, 2020.
- [14] D. Ivanov et al., "Iot-driven digital twin for industrial operations," *Production Planning & Control*, vol. 32, no. 6, pp. 493–507, 2021.
- [15] A. Rasheed et al., "Digital twins for maintenance and reliability," *Computers in Industry*, vol. 113, p. 103130, 2020.
- [16] J. Lee et al., "Ai-driven digital twin for predictive maintenance in manufacturing systems," *Annual Reviews in Control*, vol. 49, pp. 201–214, 2020.
- [17] K. Bruynseels et al., "Digital twin in healthcare: A survey," *Journal of Medical Internet Research*, vol. 20, no. 5, p. e163, 2018.
- [18] B. Björnsson et al., "Digital twins for personalized medicine," *Genome Medicine*, vol. 12, no. 1, p. 4, 2020.
- [19] F. Tao et al., "Digital twin in healthcare: Diagnostics and modeling," *IEEE Access*, vol. 8, pp. 65 200–65 213, 2020.
- [20] T. Wang et al., "Digital twin technology in patient care pathways: Opportunities and challenges," *Healthcare Technology Letters*, vol. 6, pp. 152–158, 2019.
- [21] Q. Liu et al., "Frameworks for digital twin-based healthcare management systems," *Journal of Healthcare Engineering*, vol. 2020, pp. 1–15, 2020.
- [22] Y. Deng et al., "Digital twin for smart cities: A review of technological trends," *IEEE Access*, vol. 9, pp. 140 222–140 240, 2021.
- [23] K. H. Lim et al., "Urban digital twins for smart city transformation: The case of virtual singapore," *Smart Cities*, vol. 3, no. 3, pp. 928–946, 2020.
- [24] M. Batty, *Journal of Urban Technology*, vol. 25, no. 2, pp. 3–17, 2018.
- [25] Q. Liu et al., "Digital twins in ecosystems for smart cities," *Smart Cities*, vol. 3, no. 3, pp. 928–946, 2020.
- [26] H. Kim et al., "The role of digital twins in smart cities: Challenges and opportunities," *Sensors*, vol. 20, no. 17, p. 4765, 2020.
- [27] E. J. Tuelgel et al., "Digital twin in aerospace industry: A comprehensive review," *Journal of Aerospace Information Systems*, vol. 8, no. 4, pp. 172–187, 2011.
- [28] S. Feng et al., "Digital twin for autonomous vehicle development: Challenges and opportunities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4216–4230, 2021.
- [29] E. Glaessgen and D. Stargel, "Digital twins for aircraft design and simulation," *NASA Technical Reports*, 2012.
- [30] A. Rasheed et al., "Digital twin in aerospace and automotive systems: Challenges and applications," *Computers in Industry*, vol. 113, p. 103130, 2020.
- [31] J. Lee et al., "Digital twin for predictive maintenance in the automotive industry," *International Journal of Automotive Technology*, vol. 21, pp. 451–468, 2020.
- [32] W. Kritzinger et al., "Predictive maintenance using digital twin: A systematic review," *Computers in Industry*, vol. 100, pp. 58–71, 2018.
- [33] F. Tao et al., "Ai-augmented digital twin for smart manufacturing," *International Journal of Production Research*, vol. 58, no. 17, pp. 5320–5331, 2020.
- [34] M. Grieves and J. Vickers, "Digital twin for maintenance and reliability," *Springer International Publishing*, pp. 141–174, 2017.
- [35] D. Ivanov et al., "Iot-driven digital twin for predictive maintenance in supply chains," *International Journal of Production Economics*, vol. 232, p. 107984, 2021.
- [36] C. Liu et al., "Iot-enhanced digital twins for predictive analytics," *Procedia CIRP*, vol. 91, pp. 760–765, 2020.
- [37] A. Rasheed et al., "Integrating artificial intelligence and digital twins: A survey," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5679–5691, 2020.
- [38] F. Tao, M. Zhang, and Q. Liu, "Smart manufacturing based on digital twin technology," *International Journal of Production Research*, vol. 58, no. 5, pp. 1133–1153, 2020.
- [39] Q. Liu et al., "Machine learning-driven digital twins for industrial applications," *Procedia Manufacturing*, vol. 39, pp. 1218–1224, 2019.
- [40] H. Zhang et al., "Cognitive digital twins: Ai-based frameworks for complex systems," *IEEE Access*, vol. 9, pp. 12 345–12 358, 2021.
- [41] J. Wang et al., "Artificial intelligence-enhanced digital twins in smart grids," *Applied Energy*, vol. 301, p. 117521, 2021.
- [42] X. Zhang, Y. Li, T. Li, Y. Gui, Q. Sun, and D. W. Gao, "Digital twin empowered pv power prediction," *Journal of Modern Power Systems and Clean Energy*, vol. 12, no. 5, pp. 1472–1483, 2024.
- [43] R. Minerva et al., "Digital twin for iot systems: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3491–3498, 2020.
- [44] F. Tao et al., "Digital twins and iot: Building the future of smart manufacturing," *Journal of Manufacturing Systems*, vol. 58, pp. 204–215, 2019.
- [45] D. Ivanov et al., "Iot-driven digital twin for industrial operations," *Production Planning & Control*, vol. 32, no. 6, pp. 493–507, 2021.

- [46] H. Kim *et al.*, “Iot-based digital twin for real-time monitoring,” *Sensors*, vol. 20, no. 17, p. 4765, 2020.
- [47] C. Liu *et al.*, “Iot-enhanced digital twins for predictive analytics,” *Procedia CIRP*, vol. 91, pp. 760–765, 2020.
- [48] K. Mehdi, M. Lounis, A. Bounceur, and T. Kechadi, “Cupcarbon: A multi-agent and discrete event wireless sensor network design and simulation tool.” *ICST*, 8 2014.
- [49] A. Bounceur, “Cupcarbon: A new platform for designing and simulating smart-city and iot wireless sensor networks (sci-wsn),” in *Proceedings of the International Conference on Internet of things and Cloud Computing (ICC’16)*, 2016, pp. 1–4.
- [50] A. Bounceur, L. Clavier, P. Combeau, O. Marc, R. Vauzelle, A. Masserann, J. Soler, R. Euler, T. Alwajeeh, V. Devendra, U. Noreen, E. Soret, and M. Lounis, “Cupcarbon: A new platform for the design, simulation and 2d/3d visualization of radio propagation and interferences in iot networks,” in *2018 15th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, 2018, pp. 1–4.
- [51] U. Noreen, A. Bounceur, and L. Clavier, “A study of lora low power and wide area network technology,” in *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2017, pp. 1–6.
- [52] N. Kadjouh, A. Bounceur, M. Bezoui, M. E. Khanouche, R. Euler, M. Hammoudeh, L. Lagadec, S. Jabbar, and F. Al-Turjman, “A dominating tree based leader election algorithm for smart cities iot infrastructure,” *Mobile Networks and Applications*, vol. 28, pp. 718–731, April 2023.
- [53] M. Kara, A. Laouid, M. AlShaikh, A. Bounceur, and M. Hammoudeh, “Secure key exchange against man-in-the-middle attack: Modified diffie-hellman protocol,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 3, p. 380–387, Dec. 2021. [Online]. Available: <https://journal.uad.ac.id/index.php/JITEKI/article/view/22210>



Ahcene Bounceur is an Associate Professor of Computer Science at the University of Sharjah. He previously served as a Senior Associate Professor with professorial qualifications at the University of Brest, France. In 2021, he was appointed by the French Ministry as a member of the National Council of Universities (CNU). His current research focuses on Agentic AI, Digital Twin IoT (DT-IOT), and the cybersecurity of smart critical infrastructures, while also exploring Data Science, Wireless Sensor Networks (WSN), and the theory

of pseudo-polygons, which he founded. Dr. Bounceur earned his Habilitation to Lead Research (HDR) in 2014, his Ph.D. in Micro and Nano Electronics from Grenoble INP, France, in 2007, an M.Sc. in Operations Research from ENSIMAG, Grenoble, in 2003, and an Engineering degree in Operations Research from the University of Bejaia. He coordinated many projects, has authored over 200 publications and is frequently invited as a keynote speaker at international conferences. He is also the creator and lead developer of the CupCarbon simulator.



Madani Bezoui is an Associate Professor at CESI (Centre d’Etudes Supérieures Industrielles) and a researcher at the LINEACT laboratory in France. He holds a Doctorate in Operational Research and has extensive expertise in multi-objective optimization, machine learning, and data science. His research interests encompass the modeling and optimization of complex systems, Industry 4.0 and 5.0, and the application of Internet of Things (IoT) technologies in various domains.

Dr. Bezoui also serves as the Head of the ‘Data Sciences’ program for fifth-year Computer Science Engineering students at CESI and acts as a pedagogical tutor for FISA training courses. He has supervised numerous Master’s and Bachelor’s students and has contributed to the development of curricula in operations research, scheduling, databases, and research project management. His work bridges academia and industry, focusing on enhancing competitiveness through innovative digital solutions.



Foudil Mir received his Ph.D. degree in Computer Science from the University of Bejaia, Algeria, where he is currently an Assistant Professor with the Computer Science Department, Faculty of Exact Sciences. He is affiliated with the LIMED Laboratory. His research interests include distributed and parallel systems, wireless sensor networks, IoT/IIoT, clustering algorithms, energy management, load balancing, deep learning applications in networked systems, and temporal data processing.



Adel Oulefki is an Assistant Professor in the College of Engineering and IT at the University of Dubai, specializing in Information Security, IoT, and Smart Systems. His research integrates AI, digital twins, and bioimaging to enhance intelligent automation and data-driven decision-making. Dr. Oulefki is also actively involved in smart campus initiatives, and international research collaborations across Europe, North America and the Middle East. As an IEEE member and conference treasurer co-chair for UCC and BDCAT 2024, he

combines technical innovation with educational excellence, mentoring students and advancing interdisciplinary projects in engineering and computing.



Mohamed-Amine Ouamri received his bachelor’s and master’s degrees in telecommunications from the University of Béjaïa, Algeria, in 2012 and 2014 respectively, and his doctorate in wireless networks from the same university in 2017. From 2014 to 2015, he worked as an engineer at OTA Telecom. He then pursued a postdoctoral fellowship at the Polytechnic University of Timisoara, Romania, and was a visiting researcher at the University of Bretagne Occidentale, France, for six months. He is currently an Associate Professor in

the Department of Networks and Telecommunications at Sorbonne Paris Nord University, and a member of the L2TI as well as LaMSN. Dr. Ouamri has served as a reviewer for several high-impact journals.



Huseyin Seker is a distinguished academic and researcher specializing in data science, artificial intelligence, and machine learning. He currently serves as the Head of the Department of Information Systems at the University of Sharjah, UAE, having previously held the position of Associate Dean for Research, Innovation, and Enterprise. Professor Şeker earned his Ph.D. in Biomedical Computing with a focus on AI and data science from Coventry University, UK, and has held significant academic roles at several institutions in

the UK, including Staffordshire University, Northumbria University, and De Montfort University. His research spans AI applications in biomedical fields, finance, and digital transformation, and he has published over 100 peer-reviewed papers while leading major collaborative research projects. In addition to his academic work, he is a founding member of the UK Institute of Coding and plays an active role in entrepreneurial ventures such as Bubo.AI, a startup specializing in AI-driven pricing solutions.



Sebti FOUFOU is a distinguished computer scientist and academic leader, currently serving as the Chair of the Department of Computer Science at the University of Sharjah, UAE. He earned his Ph.D. in Computer Science from the University Claude Bernard Lyon I, France, and his HDR from the University of Burgundy, Dijon, France. Professor FOUFOU has held prominent academic positions in France, the UAE, and Qatar, including professorships and department leadership roles. His research focuses on geometric modeling, 3D

representations, image processing, product lifecycle management, smart manufacturing systems, and artificial intelligence applications. Over his career, he has supervised numerous PhD and master's students, published extensively, and contributed significantly to advancing computer science research and education in his fields of expertise.

He also serves as Chair of the IEEE UAE Chapter for Robotics and Autonomous Systems, Control Systems, and Engineering.



Abbas AMIRA received his Ph.D degree from Queen's University Belfast, United Kingdom and he has taken academic, leadership and consultancy positions in the United Kingdom including Queen's University Belfast, Brunel University and De Montfort University in Leicester, France, Malaysia, and the Middle East at Qatar University, Qatar. He is currently the Dean of the College of Computing and Informatics at the University of Sharjah, UAE. During his career to date, Prof. Amira has been successful in securing substantial

funding from government agencies and industry; he has supervised more than 25 PhD students and has over 380 publications in top journals and conferences in the area of embedded systems, artificial intelligence, IoT, image and signal processing. He obtained many international awards, including the VARIAN prize offered by the Swiss Society of Radiobiology and Medical Physics, CAST award, DELL-EM Envision the future, IET premium award and many best paper and recognition awards in IEEE international conferences and events.



Yassine HIMEUR (Senior Member, IEEE, LinkedIn) is globally recognized for his significant contributions to Artificial Intelligence, especially in areas where AI intersects with cybersecurity, big data analytics, and energy management. Named among the Top 2% Highly Influential Scientists in 2022, 2023, and 2024 by Stanford University and Elsevier, Dr. Yassine Himeur is an Associate Professor at the College of Engineering and Information Technology at the University of Dubai, where he also serves

as Director of the Master of Electrical Engineering program. From 2022 to 2025, he was an Assistant Professor at the same college. He obtained his M.Sc. and Ph.D. degrees in Electrical Engineering in 2011 and 2015, respectively, and earned the Habilitation to Direct Research in 2017. His academic journey includes a postdoctoral fellowship at Qatar University (2019–2022) and a tenure as Senior Researcher and Head of the TELECOM Division at the Algerian Center for Development of Advanced Technologies (CDTA). Dr. Himeur has authored and co-authored over 200 research publications and co-led several research projects funded by the Qatar National Research Fund (QNRF), Qatar University, and the University of Dubai. He received the Best Research Award at the University of Dubai for the academic year 2023/2024, the Best Student Paper Award at IEEE GPECOM 2020, and the Best Paper Award at the 11th IEEE SIGMAP in 2014. He is the Publicity Chair for ICSPIS 2024 (Dubai), Proceedings Chair for the 10th IEEE BigDataService 2024 (Shanghai, China), and previously served as Proceedings Chair for the 9th IEEE BigDataService 2023 (Athens, Greece). He is the Special Issue Editor for the Internet of Energy and Artificial Intelligence for Sustainable Cities in MDPI Energies, Chair of the 4th Workshop on Distributed Machine Learning for the Intelligent Computing Continuum (DML-ICC) at IEEE/ACM UCC 2024, and Chair of the Embedded Intelligence Workshop at IEEE/ACM BDCAT 2024. Beyond research, Dr. Himeur actively teaches Advanced AI, Text Mining, and Research Methods, and supervises PhD and Master's students.