ETSI TS 103 845 V1.1.1 (2024-02)

TECHNICAL SPECIFICATION

SmartM2M;
Digital Twins Communication Requirements

Reference

DTS/SmartM2M-103845

Keywords

digital twins, IoT, oneM2M, ontology

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies the definition and communication aspects for Digital Twins (DTs), defining their fundamental characteristics and the requirements for their communications and interoperability, through edge-cloud continuum deployments and with respect to their Physical and Digital Interfaces.

The purpose of the present document is to enable the use cases in ETSI TR 103 844 [i.2] (D1) and to support all the major use cases and requirements in the context of DTs. It deals with the architectural aspect of the communication and the set of information needed to ensure interoperability across installations and platforms, without specifying the specific applications that use this information.

The communication for DTs relies on existing specifications that are referenced in the present document, but the definition of the elements and the information to be exchanged is kept independent from the underlying communication framework and technology. This is to minimize the impact of the evolution of the communication framework on the information managed by the DTs. The present document focuses on the definition of DTs communication functionality and properties, as well as the standardization of DTs communication requirements.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Computers in Industry, Volume 134 (2022): "Implementation of digital twins in the process industry: A systematic literature review of enablers and barriers", Matteo Perno, Lars Hvam, Anders Haug, 103558, ISSN 0166-3615.

[i.2] ETSI TR 103 844: "SmartM2M Digital Twins and Standardization Opportunities in ETSI".

[i.3] IEEE™, vol. 108, no. 10 (October 2020): "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models," R. Minerva, G. M. Lee and N. Crespi, pp. 1785-1824, doi: 10.1109/JPROC.2020.2998530.

[i.4] Alessandro Ricci, Angelo Croatti, Stefano Mariani, Sara Montagna, and Marco Picone: "Web of Digital Twins", ACM Trans. Internet Technol. 22, 4, Article 101 (November 2022), 30 pages.

[i.5] Journal of Manufacturing Systems, Volume 58, Part A, (2021): "Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review", Diego G.S. Pivoto, Luiz F.F. de Almeida, Rodrigo da Rosa Righi, Joel J.P.C. Rodrigues, Alexandre Baratella Lugli, Antonio M. Alberti, Pages 176-192, ISSN 0278-6125.

[i.6] Manufacturing Letters, Volume 15, Part B (2018): "Digital twin - Proof of concept, Manufacturing Letters", Sebastian Haag, Reiner Anderl, Pages 64-66, ISSN 2213-8463.

[i.7] 2022 IEEE™ 42nd International Conference on Distributed Computing Systems (ICDCS): "A Digital-Twin Based Architecture for Software Longevity in Smart Homes", Bologna, Italy, 2022, pp. 669-679.

[i.8] W3C® - Web of Things - Documentation.

[i.9] ACM Trans. Internet Things 4, 1, Article 8 (February 2023): "A Flexible and Modular Architecture for Edge Digital Twin: Implementation and Evaluation", Marco Picone, Marco Mamei, and Franco Zambonelli, 32 pages.

[i.10] Smart Applications REFerence Ontology, and extensions (SAREF), European Telecommunications Standards Institute (ETSI).

[i.11] IEEE™ Access, vol. 8 (2020): "Digital Twin: Enabling Technologies, Challenges and Open Research", A. Fuller, Z. Fan, C. Day and C. Barlow, pp. 108952-108971.

[i.12] ETSI STF 641: "SAREF Digital Twins", Technical Body: SmartM2M - Project No: 641.

[i.13] oneM2M - Global IoT Technical Specifications.

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Digital Twin (DT): comprehensive software representation of an individual Physical Object

NOTE 1: Denoted also a Physical Twin.

NOTE 2: It includes the properties, conditions, relationships, events and behaviour(s) of the real-life object through models and data. A Digital Twin is a set of realistic models that can digitalize and simulate an object's behaviour in the deployed environment. The Digital Twin represents and reflects its physical twin and remains its virtual counterpart across the object's entire lifecycle [i.1].

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
CDT	Composed Digital Twin
CoAP	Constrained Application Protocol
DCA	Digital Communication Adapter
DCC	Digital Communication Channel
DT	Digital Twin
DTD	Digital Twin Description
HTTP	Hypertext Transfer Protocol

HVAC	Heating Ventilation Air Conditioning
IIoT	Industrial Internet of Things
IoT	Internet of Things
JSON	JavaScript Object Notation
mDNS	multicast Domain Name System
MES	Manufacturing Execution System
MQTT	Message Queuing Telemetry Transport
OPC UA	OPC Unified Architecture
PA	Physical Asset
PCA	Physical Communication Adapter
PCC	Physical Communication Channel
PT	Physical Twin
PTD	Physical Twin Description
SAREF	Smart Applications REFerence ontologies
SenML	Sensor Measurement Lists
UPnP	Universal Plug and Play
WoT	Web of Things
XML	eXtensible Markup Language

4 Digital Twin Communication Patterns

4.0 Introduction

In the realm of Digital Twins (DTs), communication is the fundamental cyber-physical capability that enables these virtual counterparts to digitalize, interact, and evolve in harmony with their physical counterparts. Clause 4 delves into the most important communication patterns that characterize the functioning of DTs. It explores how DTs engage in physical and digital interactions, how they replicate their real-world counterparts, and how they compose themselves to address complex challenges and also analyses the challenging field of cross-domain communication, where DTs from different domains and application scenarios come together to exchange insights and collaboratively shape a smarter, more connected world. Furthermore, this clause investigates the characteristics and challenges associated with the possibilities to have direct and indirect communication patterns between Digital Twin (DT) and Physical Twin (PT) and how the interaction can change in an edge-to-cloud DT continuum with multiple deployed DTs.

In this clause and in general also in the upcoming parts of the present document interoperability and communication functionality and requirements introduced in ETSI TR 103 844 [i.2] are recalled and elaborated together with the high-level abstraction of a DT illustrate in Figure 1 and characterized by the following main components.

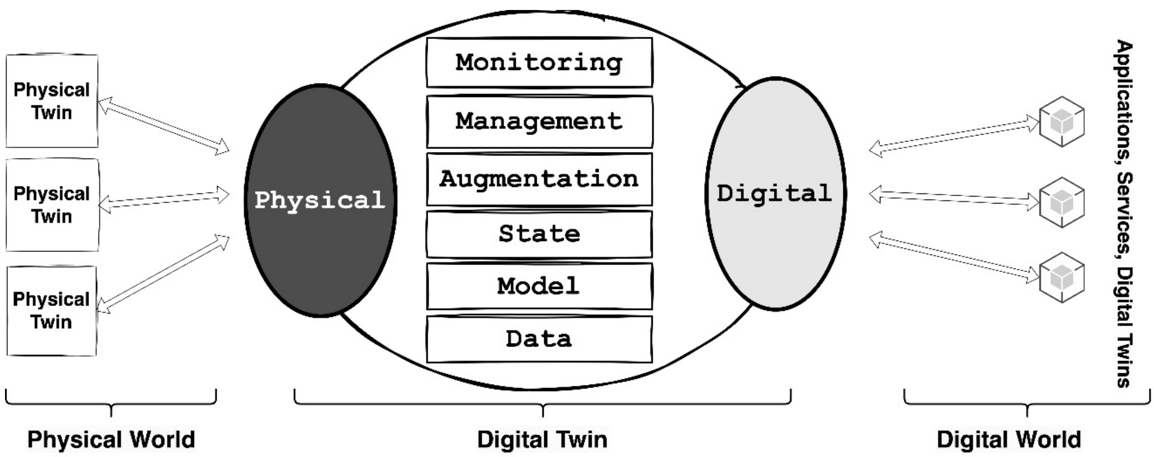

Figure 1: High level abstraction of a DT with its main cyber-physical communication channels and interaction points (Source: ETSI TS 103 844 [i.2])

In this exploration, an important element of investigation is the comprehensive analysis of the challenges and requirements associated to interoperability and communication specifications associated to the design, implementation, and deployment of interoperable DTs. As presented and highlighted in [i.2], the effective operation of DTs within existing interconnected world relies heavily on their seamless communication across various communication components and interaction flows. These layers, enveloping both the *physical* and *digital* dimensions of a DT's core, serve as the pillars upon which robust interactions and collaborations are built, illustrated schematically in Figure 1 and that can enable the vision of an ecosystem of interconnected DTs [i.4]. By underscoring the importance of structured communication channels, this clause delves into an initial analysis of the features and requisites that serve as the essential conduits for achieving interoperability, ensuring that DTs can coexist harmoniously, share critical data, and contribute to the broader operational landscape.

At its heart, a DT is a complex entity with several key responsibilities and modules that collectively enable its multifaceted functions. The *Data* module serves as the foundational component, entrusted with the handling, distribution, collection, processing, and storage of information derived from the physical asset it mirrors. This module it is also in charge of maintaining the DT's current state and potentially retains a historical record of previous states, capturing real-world data ranging from sensor readings and equipment status to environmental conditions.

The *Model* component represents the essence of a DT, embedding the intricate logic required to mirror and digitalize the target PTs into precise digital replicas. It shares a tight-knit relationship with the Data module, utilizing real-world data to craft an intricate digital representation of the physical asset. Furthermore, the Model module bears the responsibility of computing the new DT's state based on information gleaned from the physical world and the DT's design. It encompasses the physical asset's structure, behavior, and attributes, empowering the DT to replicate, analyse, and gain a profound understanding of its real-world counterpart.

Crucially, the *State* module takes on the pivotal role of maintaining a synchronized representation of the DT and its physical counterpart. It diligently captures the status and condition of the twin as a reflection of the interconnected physical world. Through the State module, the DT vigilantly tracks changes and updates, ensuring that its digital representation mirrors the real world with utmost accuracy. The State module further excels in structuring and preserving the list of properties, events, relationships, and actions available on the twin over time. This structured representation emerges as a strategic asset, enhancing interoperability by enabling multiple DTs to share a common framework, thus facilitating external services, applications, and even other DTs to discover available capabilities and engage seamlessly.

In the realm of DTs, the introduction of an architectural module focused on *Augmentation* brings forth the prospect of an adaptable layer. This innovation empowers the DT to evolve by enhancing, modifying, or extending its capabilities over time. This adaptability encompasses the addition of new properties, relationships, behaviors, and even entire components, ensuring that the DT remains in sync with changing needs and requirements.

The critical responsibility of managing the DT's lifecycle falls upon the *Management* components. These components authorize external entities to deftly oversee the creation, deployment, operation, and eventual retirement of the DT. Simultaneously, they ensure that the DT is seamlessly configured and integrated into its operational context, effectively managing interactions with other systems and services.

Lastly, the *Monitoring* module assumes a central role in the observation and evaluation of the DT's performance, health, and behavior. With a specific focus on the quality of the entanglement within the cyber-physical relationship between the DT and its associated PTs, this module excels at identifying issues, anomalies, and optimization opportunities, providing valuable insights into the DT's operation and its impact on the physical asset.

Collectively, these core components form a complex and advanced architectural goal that constitutes a DT, enabling it to bridge the physical and digital dimensions, empower decision-making processes, and deliver invaluable insights across a diverse spectrum of applications and industries. These envisioned modules are poised to play pivotal roles in both physical and digital communications, facilitating the realization of the DT's cyber-physical capabilities. The forthcoming clauses, delve deeper into the primary requirements and specifications associated with both physical and digital communications of a DT, its description and discoverability shedding light on their interactions with these core components.

4.1 Physical Interaction

One of the foundational aspects that underpin the functionality of DTs is their ability to seamlessly interact with their physical counterparts. This interaction takes place through the Physical Communication Channel (PCC), a fundamental component that a DT shall include in its design and implementation and that serves as the conduit between the digital and physical realms. The PCC is essential in the fulfillment of the DT's communication capabilities and adaptability, facilitating effective interaction among a myriad of physical twins, which can encompass various objects, devices, and target assets.

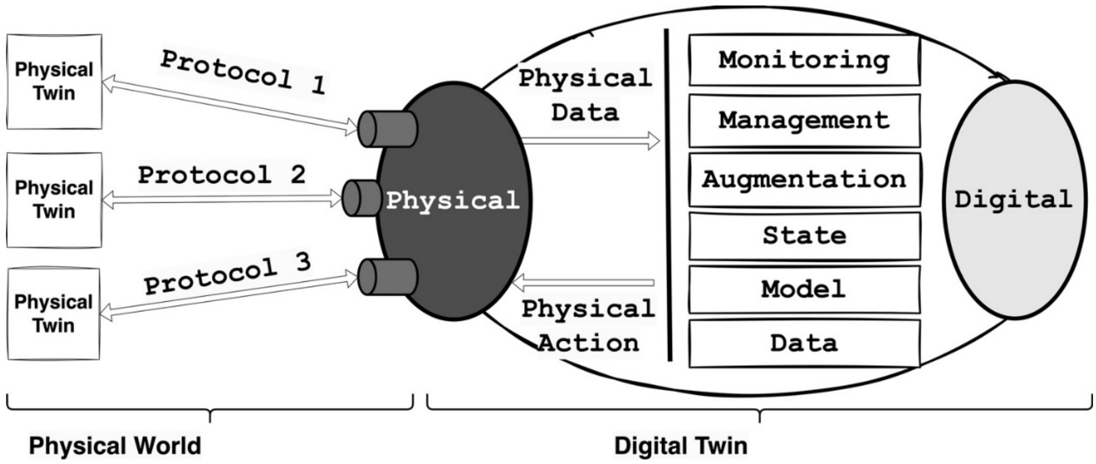

Figure 2: Handling Multiple Adapters for Diverse Communication Patterns

The cornerstone of the PCC's adaptability lies in its capability to accommodate multiple Physical Communication Adapters (PCAs). These adapters shall be used as interfaces between the DT and different physical assets, allowing the DT to communicate with various physical twins that use heterogenous protocols, data formats, or communication patterns. This modular approach (illustrated in Figure 2) enables a DT to exchange data with a wide variety of physical objects, from manufacturing machines employing multiple protocols to sensor-rich devices utilizing different data formats [i.6]. In more complex scenarios, such as an infrastructure with numerous protocols and interaction patterns, the PCC can harness a diverse array of adapters, tailor-made to suit the distinctive traits and communication requirements of different physical objects.

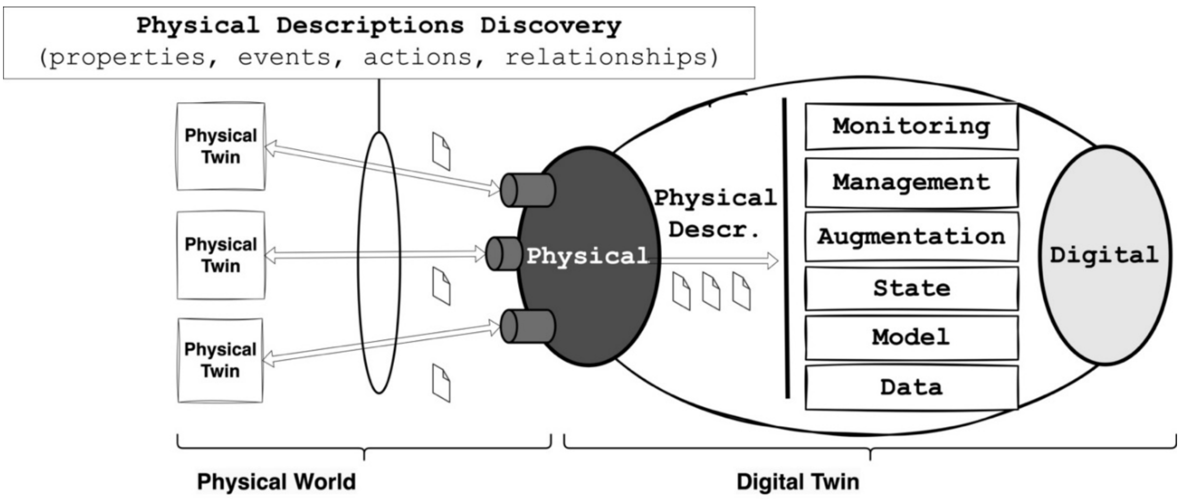

Figure 3: Receiving Descriptions to Understand Physical Twin Capabilities

One of the key challenges faced by DTs in their interaction with physical counterparts is comprehending the nature and capabilities of the associated physical twins. Each PCA shall translate the characteristics of the physical asset into a format and description that the DT can understand as schematically depicted in Figure 3. This description is invaluable, and provides the DT with details of the exposed properties, events, actions, and relationships of the physical twin.

For instance, consider two DTs associated with identical objects, such as light bulbs, but sourced from different vendors. In this scenario, the core and model of the DTs should remain the same, as these aspects reflect the essence of the twin's behavior. However, their PCC may differ significantly. Each DT shall employ one or more PCAs appropriate for the protocols, patterns, or data formats associated with their respective vendors. Thus, it is the responsibility of the entire PCC, along with its adapters, to homogenize these distinct descriptions. This decoupling of complexity, where the responsibility of conversing with a specific physical counterpart is separated from the core of the DT and its internal model, ensures that the DT remains adaptable to various physical twins without compromising its integrity.

The physical communication and interaction patterns encapsulate the ability of DTs to interface with diverse physical entities through an adaptable PCC interface. The versatile combination of PCAs handle myriad communication patterns and protocols. Additionally, each PCA provides a structured description of the physical twin's capabilities and characteristics to the DT's core and consequently the reception of multiple descriptions from the PCAs enables the DT to gain insights into the capabilities of its physical counterpart. This pattern empowers DTs to thrive in a world of heterogeneity, where multiple vendors, protocols, and data formats coexist, ensuring that DTs remain versatile and robust in their quest to bridge the digital-physical divide.

4.2 Digital Interaction

In the intricate world of Digital Twins (DTs), the Digital Interaction Pattern emerges as a fundamental component responsible for facilitating communication between the DT and the external digital universe. The Digital Communication Channel (DCC) is one of the core components of a DT instance that enables interoperability and the ability to seamlessly engage with a wide spectrum of applications and services. However, this bridge between the DT and the digital domain presents a set of distinctive challenges.

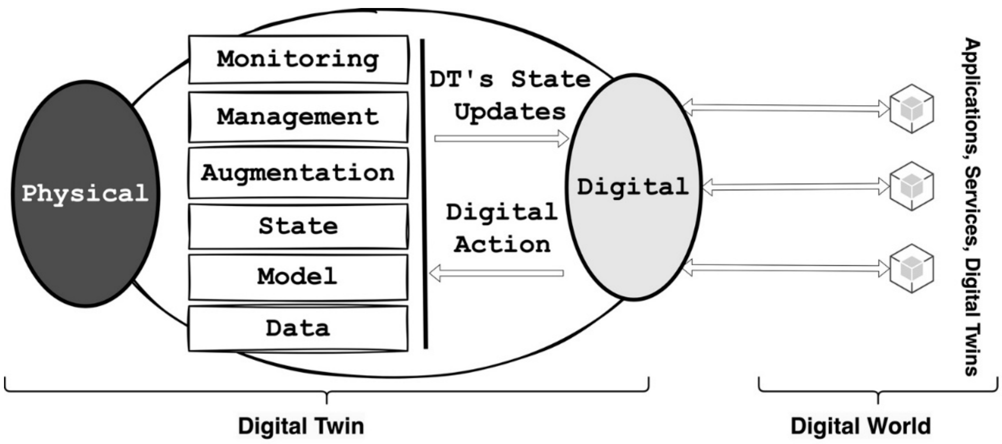

Figure 4: Interoperable and Uniform Information Flow between DT Core and the Digital Interface

A central challenge faced by DTs in the realm of digital interaction is ensuring a seamless flow of homogeneous information from the DT Core to the DCC. The DT Core shall serve as the epicenter of the twin, housing its fundamental behaviors, properties, events, relationships, and actions. For effective communication with the external digital world, this core shall distill and represent the digital twin's internal variations in a format that external applications can readily comprehend.

The DCCs role in this context shall be to accept this homogeneous information associated to DT's state update and transform it into various protocols and data formats, depending on the needs of external digital entities. Additionally, the DCC shall support the reception, management and forwarding of incoming commands targeting the DT instance and coming from external digital application and/or other twins. This adaptation should ensure that the DT remains versatile and adaptable, with the DCC acting as a bridge that connects the DT Core with a multitude of digital applications and services. Thus, the challenge is to efficiently manage this translation process while maintaining the core's independence from both physical and digital contexts.

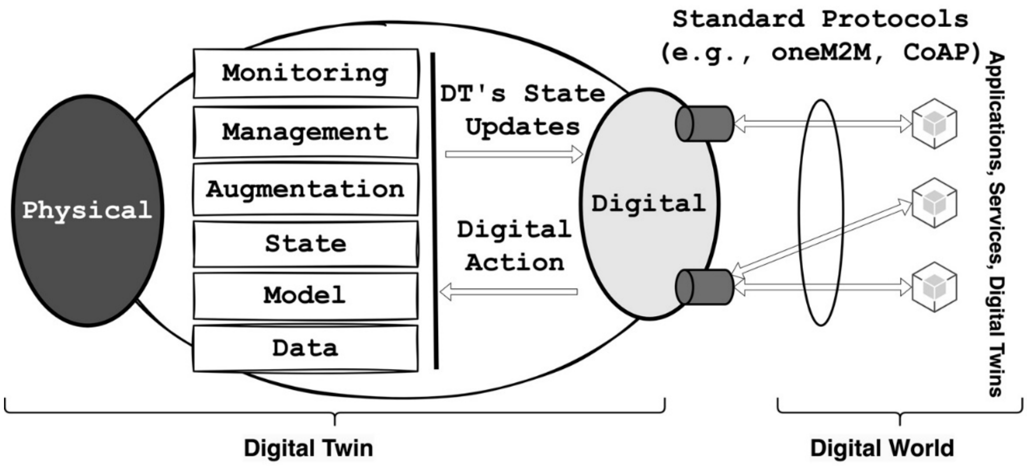

Figure 5: Using Digital Adapters of the Digital Interface for Flexible Deployment

One of the strengths of the DCC is the flexibility in accommodating various Digital Communication Adapters (DCAs). These adapters serve to bring modularity to the DCC and act as the linchpin for molding the behavior of the DT according to its deployment context embedding the responsibility to talk and interact with external digital services, applications, and other twins. Depending on where the DT is situated and which digital ecosystem it interfaces with, different adapters can be employed to tailor its behavior as schematically represented in Figure 5. For instance, consider a DT deployed in a manufacturing environment and another in a smart city infrastructure. These two contexts necessitate distinct interactions and responses. The DT shall support one or more DCCs, enabling adaptive functionality, isolating the core data model from the custom needs of the application and services. This modular approach enhances the DT's versatility, allowing it to thrive in diverse application scenarios.

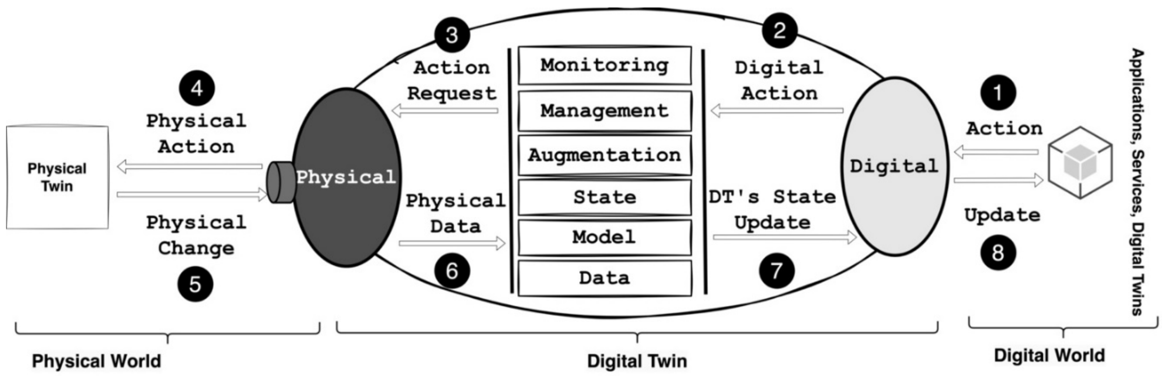

Figure 6: Handling Actions on the Digital Communication Channel and Interact with the DT Core

A key aspect of the DCC is the management and handling of actions originating from the digital interface. When external digital entities request specific actions to be performed by the DT, the DCC shall update the DT's properties, triggering events, or invoking relationships with other entities.

As illustrated in Figure 6 the DCC component acts as an intermediary, receiving these action requests and forwarding them to the DT Core for validation and adaptation. The DT core shall process these actions to ensure they align with the digital twin's behavior and context. If validated, the core then decides how to respond to the action. This response can range from manipulating the DT's internal state to triggering events and, in some cases, instructing the physical interface to carry out actions on the physical twin. With respect to the illustrated sequence of events the following steps are involved:

1) An external application, digital service or twin sends an action request on the target DT with the aim to invoke a specific action (e.g. associated to trigger a change in the status of a light bulb).

2) The DCC shall receive and process the incoming request on the associated DCA and if necessary apply required transformation or enrichment in the data format to forward to effective action request.

3) The digital action shall be received by the DT's core that validates accordingly to its internal model and forward the request to the PCC and the associated PCA responsible of handling the target exposed action.

4) The PCA shall send the actual action request to the associated PT adopting the protocol and the data format required by the specific target action and the specifications of the physical counterpart.

5) Once the action has been correctly executed by the PT, the associated variation (e.g. the change of the status of the light bulb) is detected by the same PCA.

6) The PCC, through the target PCA, shall forward the notification associated to the variation of the PT to notify the DT's core.

7) The core apply the twin's model following its current context, the new received notification and generates the new value for its state that shall be forwarded to the DCC and the available DCAs.

8) The target DCA (the one that triggered the original action request) receives the new variation of the DT's state and notify the external component the initiated the process.

The complexity of this process lies in managing the bidirectional flow of actions between the digital and physical interfaces. The DCC shall ensure that actions requested by external digital entities are validated, adapted, and then, if applicable, sent to the physical interface using the appropriate physical adapter for execution on the corresponding physical twin. This seamless orchestration of actions is a crucial aspect of the Digital Interaction Pattern, enabling DTs to participate in digital ecosystems with precision and reliability.

In essence, the Digital Interaction Pattern highlights the pivotal role played by the DCC that should be in charge of establishing effective communication between the DT and the external digital world. Challenges and responsibilities include maintaining a homogeneous information flow from the DT Core, leveraging adapters to adapt behavior based on deployment context, and handling actions requested from the digital interface with precision and reliability. This pattern positions DTs as dynamic entities that can seamlessly interact with diverse digital environments, fostering interoperability and adaptability.

4.3 Digital Twin Replication

As introduced in [i.3], the *Replication* within the realm of DTs represents a transformative concept, offering the remarkable ability to reproduce and relocate physical objects into diverse environments, both virtually and within digital ecosystems. This dynamic capability essentially for softwarizes physical entities should be included in DT architectures in order to allow with to be cloned, transformed, and distributed across various spaces, each instantiation embodying the essence of the original.

At its core, replication signifies the softwarization of physical objects, enabling the creation of multiple digital representations or "clones" within virtualization spaces. This process involves translating the physical into the digital, providing the foundation for the ensuing replication endeavors. The replication concept extends beyond merely duplicating physical entities. Each logical object, representing a DT, can also undergo replication, leading to an intricate web of interconnected digital counterparts that may also include and execute different models to implement different behaviors and representation of the same associated PT. This hierarchical approach allows for increased versatility and adaptability.

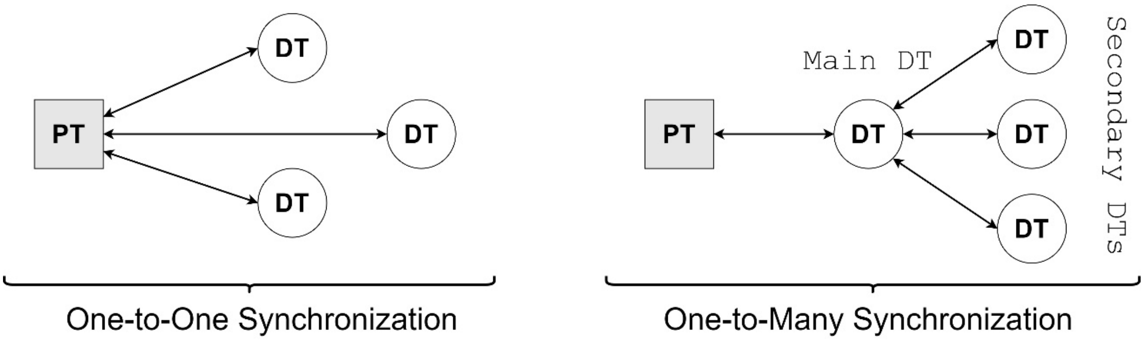

Figure 7: High-level representation of replication patterns in DTs modeling and deployments

To facilitate replication, two different communication synchronization patterns (illustrated in Figure 7) denoted as *One-to-One Synchronization* and *One-to-Many Synchronization* are envisioned and identified according to the state of the art. In the first approach, each DT maintains a direct and synchronized connection with its corresponding PT. This method ensures parity between the physical and digital realms, fostering precise representation without the use of any intermediate component or other DTs. Within the latter paradigm, a designated logical object assumes the role of a "main replica". This main replica not only should synchronize with the original physical counterpart but also assumes the responsibility of maintaining synchronicity among a group of connected replicas, referred to as "secondary twins". This arrangement optimizes resource utilization and scalability, particularly in scenarios with numerous interconnected DTs. A DT shall support at least one of the synchronization patterns.

The replication of DTs introduces distinctive communication patterns tailored to synchronize virtual replicas with their physical counterparts. The two identified synchronization modes, one-to-one and one-to-many, encompass these patterns, each posing unique challenges and opportunities for seamless interaction. These patterns, deeply intertwined with both physical and digital interfaces, orchestrate the harmonious coexistence of digital twins within complex ecosystems.

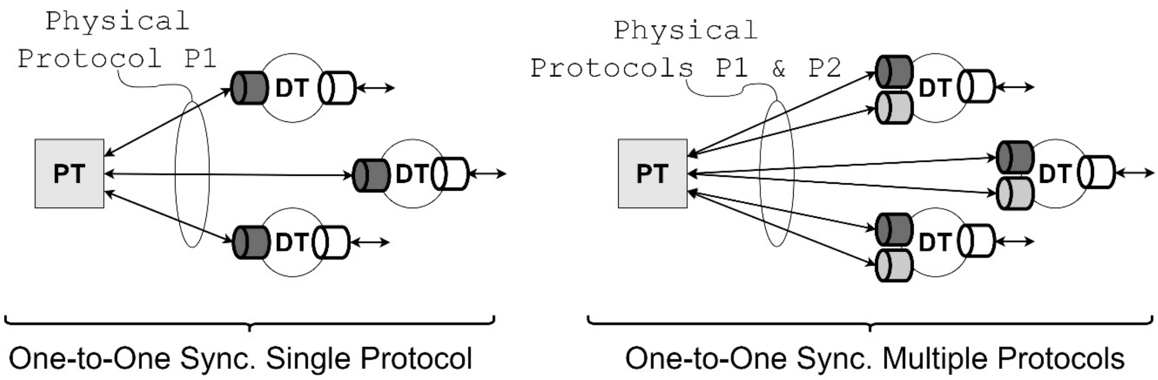

Figure 8: One-to-One Synchronization Pattern and DT's Communication Interfaces

In a one-to-one synchronization model, every DT maintains a direct and exclusive correspondence with its physical source, reflecting live changes in lockstep as reported in Figure 8 considering the presence of one or multiple physical communication capabilities and consequent need for having multiple adapters on the DT to interact with the PT. The PCC, comprised of its PCAs, assumes the pivotal role of bridging the gap between the digital and physical realms. Each adapter is dedicated to a specific physical entity, responsible for interpreting its characteristics, properties, and events. On the other hand, the DCC, equipped with its DCAs, primarily focuses on exposing the internal variations of the DT to external digital entities. It acts as a conduit for digital-to-digital interactions, allowing external applications to interact with the digital replica. Achieving one-to-one synchronization demands maintaining a direct connection between the physical and digital interfaces. Live and fresh data exchange is critical, requiring the adapters to swiftly capture and convey changes from their respective physical entities to the digital twins. This ensures that the digital replica remains a faithful representation of the original, reflecting its every nuance.

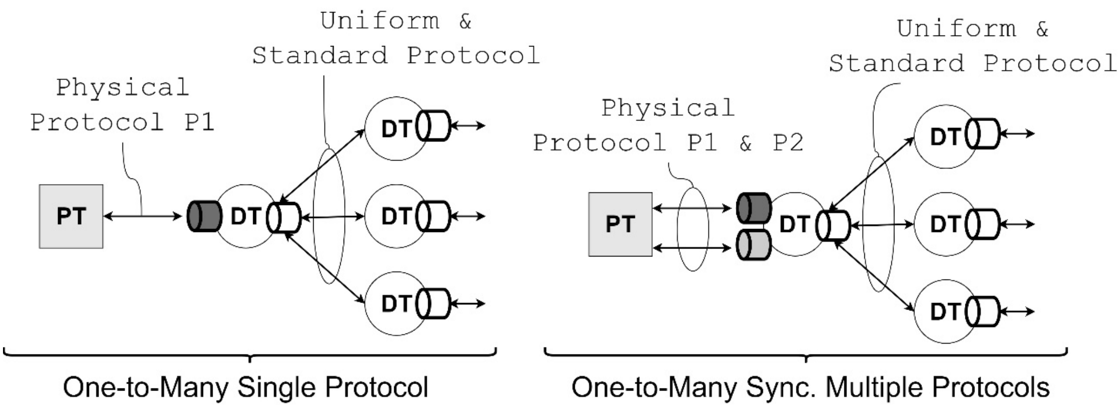

Figure 9: One-to-Many DTs Synchronization Patterns and DT's Communication Flows

One-to-many synchronization introduces on the other hand the possibility to have a hierarchical approach, where a main DT assumes the role of synchronization for a group of connected replicas and the responsibility to be the only one in charge of communicating with the PT. This model involves both digital and physical communication channels in a distinctive manner according to their architectural position on the main DT or on the secondary replica twins. Within this synchronization model, the PCC continues to interface with individual physical assets as it does in the one-to-one model. However, the master replica may interact with multiple physical assets or a consolidated view of them. The DCC's role is amplified in one-to-many synchronization. It interfaces with the primary replica, which acts as a central hub for interactions with the physical world. It is also responsible for maintaining synchronicity among the group of secondary replicas. The one-to-many model necessitates efficient data aggregation and distribution. The PCA, while interfacing with multiple physical entities, should relay relevant changes to the primary DT. The primary replica, via the DCC, orchestrates the dissemination of these changes to secondary replicas.

Summarizing, in both synchronization patterns, the relationship with physical and digital interfaces is fundamental:

- **Physical Interface Harmony:** The PCC, regardless of the synchronization mode, remains instrumental in translating physical characteristics into digital representations. The PCC shall ensure that every alteration in the physical world is promptly captured and conveyed, aligning with the chosen synchronization model.

- **Digital Interface Adaptability:** The DCC plays a critical role in adapting to the selected synchronization mode. For one-to-one synchronization, it primarily handles digital-to-digital interactions, while in one-to-many synchronization, it should manage not only external digital interactions but also internal synchronization among replicas.

In essence, communication patterns for DT replication rely on the harmonious interplay of physical and digital interfaces taking into account also the presence and adoption of different adapters at the same time on both physical and digital sides as depicted in Figure 9. Whether it is the live precision of one-to-one synchronization or the scalability of one-to-many synchronization, these patterns pave the way for digital twins to flourish within dynamic ecosystems while staying impeccably synchronized with their physical sources.

In the context of DT replication, the ability to describe and structure the hierarchical digital framework takes center stage. This feature holds immense significance as it enables external applications to navigate and comprehend the intricate web of digital twins. By providing a clear digital structure, stakeholders gain insights into the replication hierarchy, empowering them to gauge the level of entanglement between the original physical twin and the observed or connected digital twins. The flexibility to define multiple levels of replication offers tailored solutions for different use cases. While it enhances comprehensibility for certain applications, such as those focused on analytics and historical data, it might introduce undesirable delays in live or even real-time scenarios. Therefore, the capacity to customize the digital structure according to specific needs emerges as a pivotal attribute in the realm of DT replication.

4.4 Digital Twin Composition

As introduced in [i.3], DT's *Composition* (or *Composability*) is a dynamic capability that allows the digitalization and visualization of complex systems as compositions of sub-parts or the amalgamation of individual objects as depicted in Figure 10. Take, for example, a car; it can be perceived as a single entity or as the result of combining various objects like the brake system, transmission, and power production components. This composability powerfully abstracts the intricacy of large systems, permitting a focus on specific, relevant aspects tailored to particular applications. Through DTs, composability is achievable whether a DT directly connects to multiple physical devices or if it establishes connections with other DTs, effectively representing intricate relationships, like an IoT device within a room within a building.

To effectively manage these aggregations and compositions of DTs, orchestration is paramount. It ensures that these complex digital ecosystems function seamlessly. In this intricate landscape, simulation theory and agent-based modeling technologies shine, offering the ability to simulate and model the behavior of large systems comprising interacting DTs. The challenge here is the simultaneous use of multiple technologies and representations to characterize, identify, manage, and enhance internal DT processes, as well as its interactions with other twins, external applications, and services.

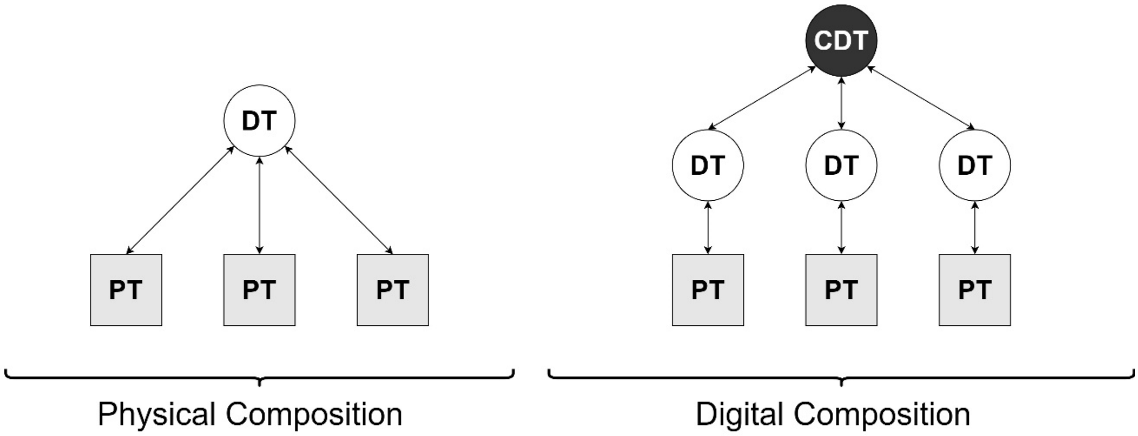

Figure 10: Schematic representation of DTs composition patterns

At its zenith, a substantial composition of DTs gives rise to a large system of systems, introducing a new frontier in managing complexity - an open challenge that beckons innovative solutions to navigate the intricate web of interconnected twins and systems.

The support of DT Composition (and of course its integration with Replication) shall be integrated in DT architectures and platforms in order to enable the possibility of designing and building complex digital systems by aggregating multiple DTs. This process can occur in two main scenarios:

a) Physical Composition: synchronizing multiple physical twins with their respective DTs.

b) Digital Composition: composing DTs into a higher-level twin (denoted as parent twin) digitalizing a low layer of twin (denoted as child twins).

Of course, this approach can be combined to build a complex hierarchical structure of DTs virtualizing a specific target application domain or use case. In both scenarios, communication patterns play a pivotal role, and they are intricately tied to the functioning of the PCC and DCC, alongside their adapters as schematically illustrated and represented in Figure 11.

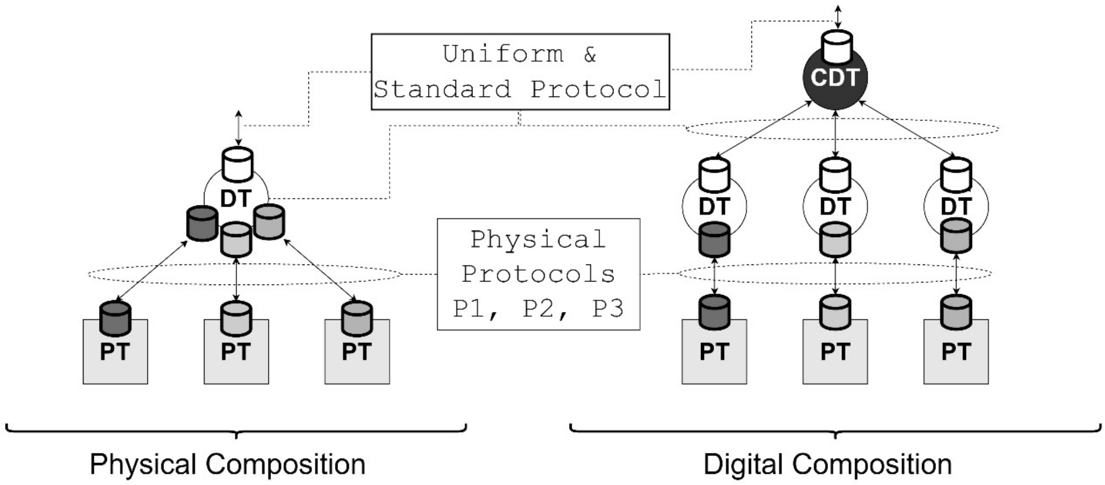

Figure 11: Synchronization of Multiple Physical Twins with DTs with a focus on Physical and Digital Communication Channels

In the case of Digital Composition, several physical objects (e.g. machines in a factory) can be replicated into the digital realm. Each physical twin is associated with a DT that mimics its behavior. To enable synchronization, each DT is equipped with a PCC and its associated adapters PCAs.

DT composition introduces a layer of complexity that extends beyond the foundational synchronization approaches previously introduced. It involves orchestration and combination of different solutions at the same time to handle interactions between digital and physical entities, relying on the adept integration of PCAs and DCAs. The two primary synchronization patterns, one-to-one and one-to-many, set the stage for exploring an advanced landscape of DT composition. Through the one-to-one pattern, each DT establishes a direct synchronization link with its physical counterpart. The responsible party here is the PCC and its PCAs, seamlessly translating real-world events into digital data. Ensuring effective communication, the PCC governs these adapters, bridging the gap between the tangible and digital realms.

Taking a more hierarchical approach, the one-to-many pattern designates one DT as the primary that shall synchronize with its physical twin and manage all connected replicas. The primary DT, equipped with its PCC and adapters PCAs, directly interacts with the physical entity. Concurrently, secondary DTs replicate the master's state through their communication channels, receiving updates in harmony potentially through homogeneous protocols, data formats and interaction patterns handling the "physical" source the DCC of their main twin. The primary DT's PCC acts as a conduit for synchronized data, with adapters meticulously mapping and transferring data to the secondary DTs.

As complexity escalates, multiple DTs can coalesce to represent intricate systems like manufacturing plants or smart cities. Communication hinges on both PCC and DCC of each DT, catering to interactions within the composite structure and external interfaces. The amalgamation of DTs may results in a hierarchical structure, where a parent DT oversees interactions among its child DTs through the interaction with their DCC and the available DCAs. The DCC of the parent DT orchestrates communication, with adapters facilitating understanding and interaction with diverse protocols and formats used by its children.

Within complex compositions, child DTs may need to communicate internally. The DCC of each child DT takes charge of these interactions, while adapters ensure seamless data translation and routing between the child DTs, fostering a collaborative environment. The parent DT acts as the gateway for external applications and services seeking interaction with the composition. The DCC of the parent DT presents a unified view of the entire structure, translating external requests into actions that fanout across the child DTs. Adapters on the DCC validate and adapt external requests, ensuring a harmonious integration into the composition.

In both synchronization patterns, the pivotal role of adapters on the PCC and DCC cannot be overstated. These adapters act as crucial bridges, navigating diverse protocols, data formats, and communication patterns. On the PCC, they guarantee accurate translation of physical events into digital data together with the management of incoming action requests and their execution. On the DCC, they facilitate seamless data flow between DTs, building an homogeneous layer of digital and interoperable twin and fostering effective collaboration and interaction with external entities.

The adaptability and versatility of these channels and their modular adapters are paramount, offering the flexibility to integrate various protocols and data formats and enabling an effective composition and composability of twins through hierarchical and synchronized deployments. This adaptability is the linchpin for DTs to communicate seamlessly within complex digital ecosystems, embodying the essence of interconnected and interoperable realms.

4.5 Replication & Composition in Cross-Domain Scenarios

Cross-domain communication in DT ecosystems is a pivotal capability that shall be implemented to enable interoperability and collaboration between DTs belonging to different domains or application scenarios. This capability unlocks a wide range of opportunities and applications, facilitating the exchange of information, insights, and actions among diverse digital and physical entities.

In many applications, different domains often use distinct protocols, data formats, and communication patterns. Cross-domain communication allows DTs to bridge these gaps, ensuring that information can flow seamlessly between them. This interoperability is essential for creating holistic solutions that encompass various aspects of the physical and digital worlds as schematically illustrated in Figure 12.

Furthermore, cross-domain communication is critical for scalability. In large-scale applications, the DT ecosystem can span multiple domains. For instance, in a smart city project, DTs may cover transportation, energy, and public services, each with its own domain-specific DTs. Cross-domain communication enables the aggregation and coordination of these DTs, creating a unified view of the entire city.

Collaboration between different stakeholders and systems is another key benefit of cross-domain communication. For instance, in an industrial setting, manufacturing machines (DTs in one domain) might need to collaborate with inventory management systems (DTs in another domain) to optimize production and logistics. This collaboration is only possible through effective cross-domain communication.

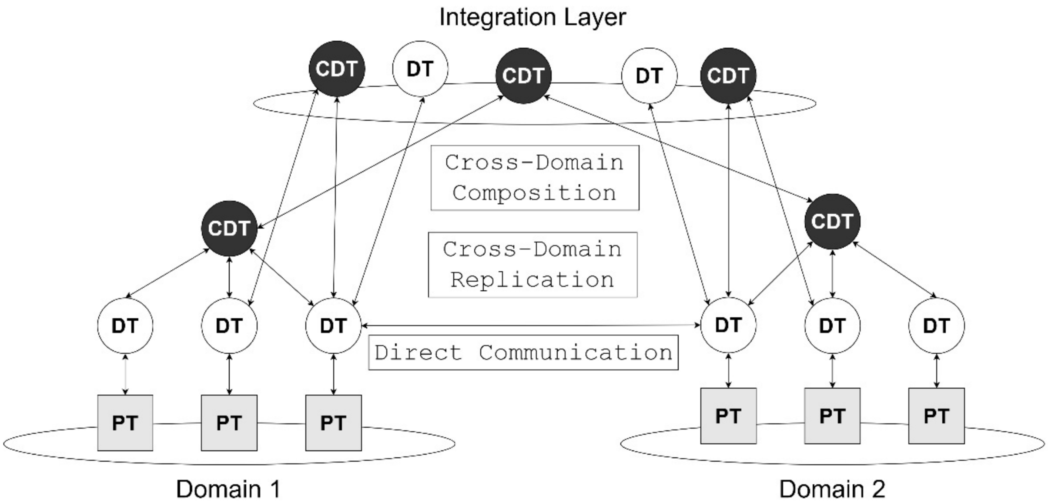

Figure 12: DTs modeling through cross-domain applications

Both the PCC and DCC are instrumental in enabling cross-domain communication as represented and structured in the schema in Figure 13. In particular, the PCC in cross-domain scenarios serves as the gateway for physical interactions with diverse entities. It accommodates various PCA capable of communicating with different types of physical assets. These adapters ensure that data from physical entities is translated and normalized, making it understandable by the DT and compatible with the communication protocols used in the target domain allowing also the possibility to share and reuse the same DT's models and implementation through different domain simply changing and adapting the adopted physical adapters.

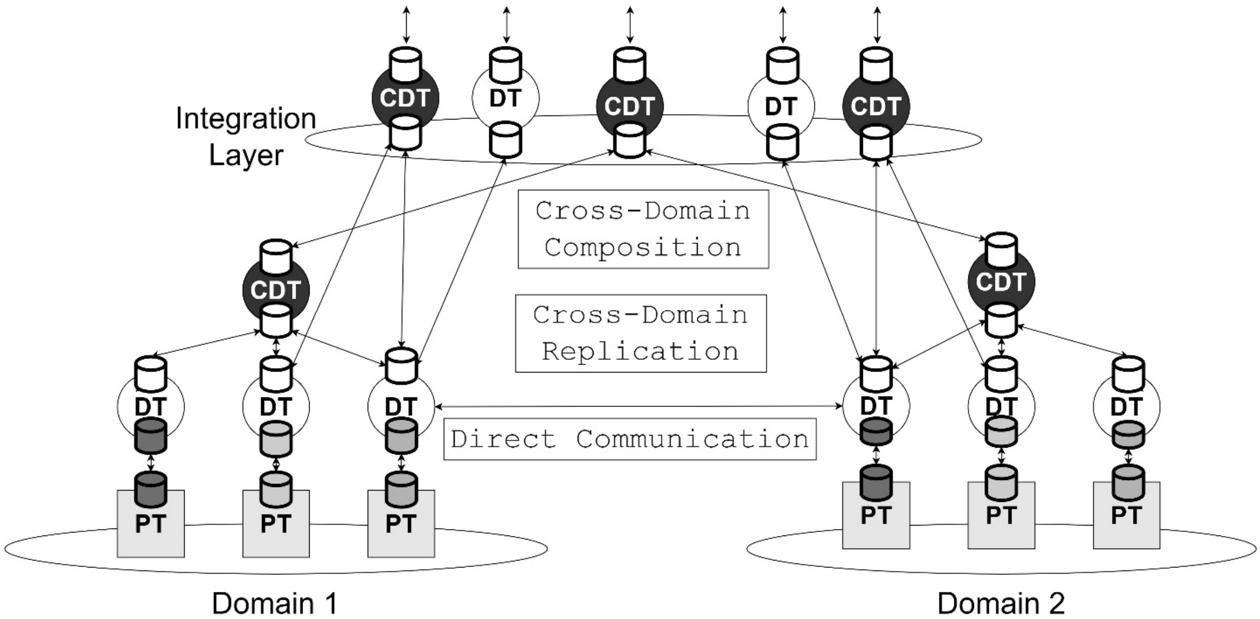

Figure 13: The Role of Physical and Digital Communication Channels in Cross-Domain DTs

The DCC, on the other hand, manages digital interactions and communication with external entities. It exposes a unified and abstracted view of the DT's capabilities and state, shielding external systems from domain-specific intricacies. The DCAs are crucial in translating digital requests and responses, ensuring that interactions with DTs from other domains are coherent and efficient.

In this context, previously analysed *Replication* and *Composition* are powerful techniques that can be harnessed to facilitate cross-domain communication. On the one hand, the replication pattern is strategic when dealing with cross-domain scenarios, replicating a DT can be a valuable approach. A DT from one domain can be replicated to communicate with DTs in another domain. This replication allows for the creation of specialized intermediary DTs according to the communication patterns and requirements of each domain involved, bridging the gap between them. On the other hand, DT Composition can create hierarchical structures where a parent DT oversees the coordination of child DTs from different domains. This hierarchical structure streamlines cross-domain communication, as the parent DT can act as a mediator and translator between the child DTs and external entities.

In summary, cross-domain communication in DT ecosystems is a fundamental enabler of interoperability, scalability, and collaboration across various application scenarios. It relies on the adaptability of both the Physical and Digital communication channels, along with techniques like Replication and Composition, to ensure that DTs from different domains can effectively communicate, share insights, and cooperate in the broader digital landscape. This capability paves the way for innovative solutions that span multiple domains and deliver comprehensive value to diverse stakeholders.

4.6 Cyber-Physical Awareness

In the context of DTs, the possibility to model and monitor over time the cyber-physical relationship between the PT and the DT (denoted also as *entanglement* in [i.3]) has emerged as a strategic property that should characterize the nature of communication and the fundamental aspect of a DT driven architecture or implementation. This concept emphasizes that the relationship between the digital representation and its physical counterpart is far from static; instead, it is dynamic and ever-changing.

At its core, this relationship and its characterization should be taken into account in the design of DT architecture and framework in order to enable and extended awareness of the supported cyber-physical relationships. Aspects (illustrated also in Figure 14) that might be considered to structure the DT cyber-physical awareness and the binding between a PT and its DT, are the following:

- **Connection:** The connectivity dimension scrutinizes how robustly the DT is linked to its PCC and the associated PCA. This spectrum extends from scenarios featuring no direct connection-a state of disconnection where the DT operates autonomously-to instances characterized by a robust and direct link between the DT and the PT.

- **Speed:** The speed aspect revolves around the temporal dynamics of communication. This spectrum spans from non-real-time scenarios, where updates from the PT to the DT may experience delays and are not immediate, to the other end of the spectrum involving hard real-time communication, where changes are instantaneously transmitted, ensuring swift and synchronous updates between the PT and DT.

- **Direction:** The directional facet delves into the nature of the relationship. It contemplates whether the association is uni-directional, signifying that the primary function of the DT is to monitor the PT, or if it is bi-directional, indicating that the DT possesses the capability to both monitor and exert control over the PA. In a uni-directional association, the DT primarily observes and gathers data from the PT. Conversely, in a bi-directional relationship, the DT not only monitors the PA but also holds the capacity to influence and control the actions of the physical counterpart. This dynamic interaction adds a layer of sophistication, allowing the DT not only to reflect the state of the PT but also to actively participate in shaping and influencing its behavior.

By dissecting the relationship between PTs and DTs along these dimensions-connection, speed, and direction are the fundamental components to gain a comprehensive understanding of how these entities interplay and communicate within the broader digital ecosystem, enriching the conceptualization and implementation of DT across diverse application domains.

The concept of entanglement in DTs communication is essential, reflecting how tightly the DT is connected to its physical counterpart, with factors like connectivity, promptness, and association playing key roles. These aspects of entanglement are highly dynamic and contingent upon various factors. The nature of the PT, be it a connected device or a passive resource, influences entanglement, as does the available infrastructure, including network speed and bandwidth. Additionally, domain-specific requirements, such as real-time demands or data freshness, contribute to the variability of entanglement.

These entanglement factors have a profound impact, shaping the design of the PCC and its adapters, determining the suitability of DTs for live or real-time applications, and influencing the reliability of the digital representation.

Recognizing the dynamic nature of entanglement is crucial for informed decision-making in DT implementations, ensuring alignment with specific use case requirements.

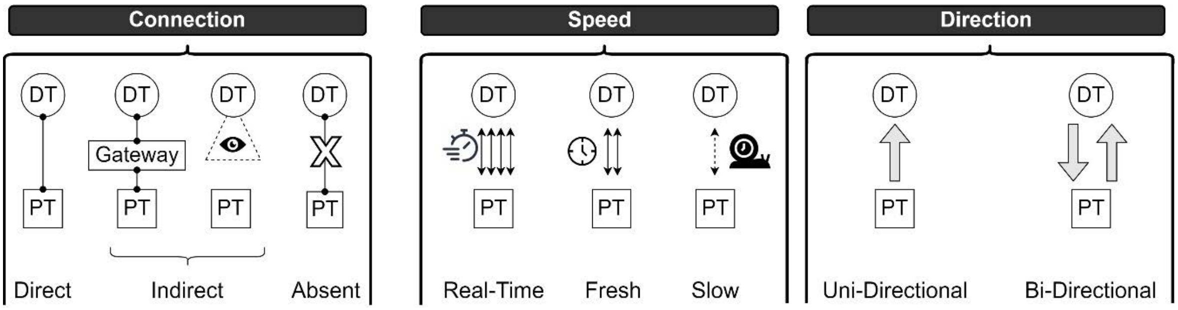

Figure 14: A schematic representation of the characteristics of the cyber-physical relationship between PT and DT

The entanglement level influences the design of the PCC and its adapters denoted as PCAs. For instance, in scenarios with high connectivity and promptness requirements, the PCC and adapters may be optimized for live data exchange and reliability. On the other hand, in situations with lower entanglement, the emphasis might shift to energy efficiency or data compression.

Applications built on DTs vary widely, from control systems in industrial settings to non-time-critical analytics. The level of entanglement dictates the suitability of DTs for specific applications. For instance, in direct and live or even real-time communication scenarios, DTs can be the foundation for control and monitoring systems. In contrast, in situations with indirect communication and lower promptness, DTs may serve more for historical data analysis or predictive modeling.

The DT's ability to faithfully mirror the state of its PT is highly influenced by the entanglement level. In direct and live communication, the DT's representation remains synchronized with the PT, offering a reliable reflection of the physical world. In indirect communication, where updates may be delayed or less frequent, the reliability of the digital representation may be reduced.

In essence, understanding and managing entanglement in DT communication patterns is pivotal and should be integrated in DT architectures and frameworks. It directly impacts the design and implementation of the PCC, the suitability of DTs even for real-time applications, and the reliability of the digital representation. Acknowledging the dynamic nature of entanglement empowers organizations to make informed decisions about their DT implementations, ensuring they align with the specific requirements of their use cases.

The significance of an external observer monitoring and observing the variations over time in cyber-physical relationships lies in the depth of insights gained into the dynamic interplay between the digital and physical realms. By continuously tracking the evolution of these relationships, external observers can discern patterns, trends, and anomalies, providing a comprehensive understanding of how digital twins synchronize with their physical counterparts. This observational capability extends beyond mere static snapshots, offering a nuanced perspective on the intricacies of changes and adaptations in the cyber-physical landscape. Additionally, monitoring these variations enables the identification of potential challenges, performance fluctuations, or emerging opportunities, fostering a proactive approach to decision-making and optimization. The ability to observe the continuous evolution of cyber-physical characteristics of connection, speed and direction shall be included in a description of the capabilities of a DT to empower external observers with actionable intelligence, contributing to the enhancement of twin functionalities and the refinement of their integration into diverse operational contexts.

4.7 Edge-to-Cloud Communication

The concept of the Edge-to-Cloud compute continuum involves a distributed architectural pattern that spans from edge devices to cloud infrastructure. This approach is highly relevant to DTs scenarios as it accommodates multiple instances deployed in both edge and cloud environments and more in general in any required distributed architectural layers according to application requirements, business logic and goals.

In the edge deployment scenario, DT instances reside close to the physical assets they represent. This proximity allows for live data processing and low-latency interactions, making it ideal for applications that demand immediate decision-making, such as industrial automation or autonomous vehicles [i.9]. Edge DTs collect and process data locally, reducing the need for continuous communication with remote cloud resources.

On the other hand, cloud (or Edge) deployment of DT instances offers scalability, composability, extensive storage capacity, and advanced analytics capabilities. These cloud-based DTs are well-suited for tasks like long-term data storage, complex analytics, and applications that do not require live or real-time responses.

The DT's replication and composability capabilities plays a significant role in modeling the Edge-to-Cloud continuum for DTs. It involves creating multiple instances of DTs, some of which are deployed at the edge, while others reside in the cloud or on edge facilities. Replication ensures redundancy, load balancing, and data availability, particularly in situations where edge devices may experience failures. Additionally, composition is essential when DT instances need to collaborate to represent a larger system or complex relationships. For example, an edge-based DT representing an individual industrial machine can be composed with a cloud-based DT representing the overall factory's efficiency. This composition provides a holistic view of the entire manufacturing process.

To enable seamless communication within this continuum, specific communication patterns can be defined:

- **Local Edge Communication Enhancement:** Efficient communication among DT instances at the edge is strategic, especially in collaborative scenarios within factories or industrial settings. This is where a group of edge devices collaborate on a task, requiring seamless interaction. Communication adapters integrated into both the PCC and DCC communication channels play a pivotal role in facilitating this local interaction. These channels and the associated adapters not only ensure a smooth flow of information but also enhance the interoperability of diverse edge devices, promoting a cohesive and synchronized operational environment.

- **Edge-to-Cloud Communication Refinement:** When considering the transmission of data and insights from edge DT instances to the cloud for long-term storage and in-depth analysis, establishing reliable communication channels is imperative. This process might involve incorporating edge servers or gateways to guarantee the integrity of data reaching the cloud. Adapters on the DCC of edge DTs contribute significantly by translating data into formats understandable by cloud-based DTs. This not only streamlines the communication process but also ensures that valuable insights from the edge are seamlessly integrated into the broader analytics framework within the cloud.

- **Cloud-to-Edge Communication Optimization:** The flow of decisions or insights from the cloud back to edge devices for immediate actions demands a robust communication mechanism. Here, the adapters on the DCC of the cloud-based DT play a crucial role in facilitating this bidirectional communication. The involved adapters are instrumental in ensuring that instructions and insights generated in the cloud are compatible with the diverse edge devices. This optimization of cloud-to-edge communication is pivotal for quick and live responsiveness and agile operational adjustments at the edge.

- **Edge Integration Streamlining:** In scenarios involving edge facilities, communication patterns should be meticulously designed to account for the proximity of edge servers to edge devices. Available facilities act as intermediaries, strategically optimizing data exchange between edge and cloud-based DT instances. This integration enhances the efficiency of the entire ecosystem by leveraging the computational capabilities of edge servers to process data closer to the edge, reducing latency and enhancing the overall responsiveness of the distributed digital twin network.

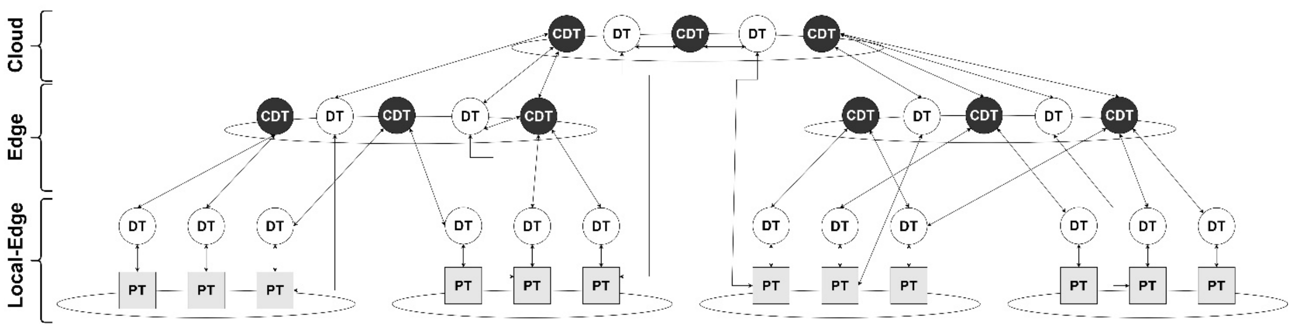

Figure 15: Illustrative architecture depicting the distribution of multiple DTs through an Edge-to-Cloud Continuum

The effective implementation of these solutions relies heavily on the proper utilization of both PCC and DCC, along with their adapters. The PCC and its adapters manage the communication with physical assets at the edge, ensuring data acquisition and device control. On the other hand, DCC and its adapters handle the translation and compatibility of data and commands between edge and cloud-based DT instances. Moreover, they facilitate composition by providing standardized interfaces for interactions between different DTs.

In summary, the Edge-to-Cloud continuum for DTs leverages replication and composition to create a flexible and adaptive architecture. Effective communication patterns, along with the intelligent use of physical and digital communication capabilities and channels through a modular and dynamic usage of adapters, ensure that DTs can seamlessly operate across this continuum, catering to a wide range of applications and requirements.

5 Digital Twin Communication Requirements

5.0 Foreword

The upcoming clause 5.1 delves into a comprehensive analysis of the communication requirements that are associated with the design and implementation of a DT. Communication lies at the heart of enabling twins to fulfill their intended functionalities and seamlessly interact within the intricate landscapes of both the physical and digital realms. These communication requirements are vital in ensuring that DTs can effectively share data, insights, and actions while contributing to the broader operational landscape. Exploration will encompass various facets of communication, including communication flows, interoperability, adaptability, and more, shedding light on the pivotal role communication plays in shaping the capabilities and effectiveness of DTs.

5.1 Digital Twin Communication Flows

As previously anticipated and introduced, in the realm of DTs, communication flows are pivotal for enabling seamless interactions and collaborations within the complex landscapes of both the physical and digital domains. These interaction flows (depicted in Figure 16) can be structured into two primary categories associated with physical and digital interaction flows and the associated responsibilities in terms of digitalization procedures and cyber-physical management and synchronization.

DTs are tasked with bridging the gap between the digital and physical worlds, which necessitates robust and versatile communication capabilities. In the context of *Physical Communication* and interaction flows, DTs face the challenge of accommodating multiple communication protocols (e.g. MQTT, HTTP, CoAP), data formats (standard or custom) and interaction patterns (such as request/response or publisher/subscriber) to effectively interact with the physical twins. This multifaceted requirement involves not only reading data from physical entities but also sending actions to influence and interact with the environment. Additionally, an essential aspect of this physical interaction flows and the associated design and implementation within a DT is the ability to decouple the inherent complexity of physical communications. DTs achieve this by forwarding the content of information while abstracting away the specific and custom characteristics of interaction protocols and patterns, ensuring that the core of the DT remains streamlined and adaptable.

At the same time, the digital layer of a DT should be responsible for communicating its internal state and insights to the digital world, as well as receiving and translating digital actions. The DCC is fundamental for serving as a dynamic gateway through which DTs interact with external digital entities, offering a spectrum of interaction patterns, protocols, and data formats tailored to the deployment scenario or operational context. On one hand, this digital interaction layer receives data from the DT's model, reflecting the evolution of its state, and exposes this information to the digital realm. On the other hand, it processes incoming digital actions and translates them back into the DT's model for analysis and potential forwarding to the physical world. However, the role of digital communication flows extends beyond merely exchanging data and actions. DTs also rely on these flows for their management and orchestration through active deployments together with the possibility of monitoring their activities over time within the digital space. This multifaceted functionality allows the digital interaction layer to manage the DT according to its nature and implementation while continuously monitoring its behavior throughout its lifecycle, thereby facilitating manageability, accountability, and overall effectiveness.

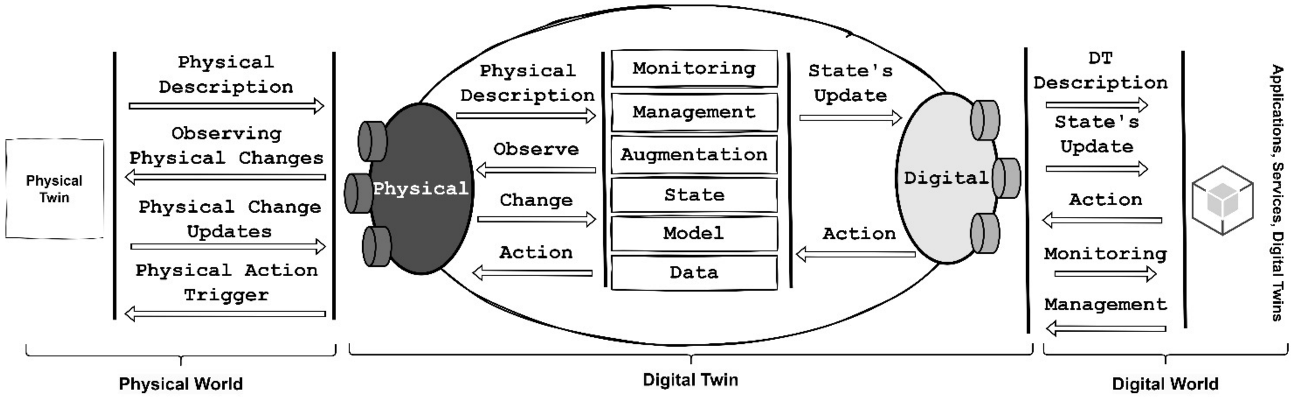

Figure 16: Schematic representation of the main characteristic of Physical and Digital communication and interaction flows in a DT

Clause 5.2 presents and analyses all the requirements associated with both Physical and Digital worlds. Additionally, it delves into an in-depth examination of advanced Digital Twin (DT) capabilities, such as replication, composition, entanglement management, cross-domain deployment, and the seamless integration of edge-to-cloud continuum functionalities, shedding light on how these advanced capabilities augment the existing requirements and specifications of DTs.

5.2 Physical Communication Requirements

This clause delves into the key requirements for the design, implementation, and deployment of the PCC of a DT. This layer serves as the vital bridge between the DT and the physical world, enabling seamless communication with various physical assets and entities. The main capabilities and requirements that should be taken into account when designing and implementing a DT framework and architecture are summarized and detailed in the following clauses.

Structured Physical Twin Description: The PCC through the deployed PCAs shall provide the DT core and model with a structured and standardized description of the associated physical twin, as schematically reported in Figure 17, and that can be denoted as Physical Twin Description (PTD). This description shall encompass details about the physical asset's characteristics, properties, events, actions, and relationships. By providing this structured representation, the layer enables the DT model to comprehensively understand the capabilities and attributes of the physical counterpart. This fundamental operation should work without any specific bound to the adopted protocols or interaction patterns. It plays a pivotal role in guiding the digitalization and replication process by clearly defining which aspects of the physical twin should be mirrored in the digital realm. This structured description forms the foundation for effective interaction, synchronization, and adaptability between the DT and its physical twin. Involved fields that should be integrated in the PTD are those that can help the core of the twin to build, maintain and describe a digital representation of the PT according to its fundamental capabilities (properties, events, actions and relationships) integrated with specific characteristics extracted and associated to the target communication protocol and integration pattern used by the PCC and its PCAs that might include additional information related to capabilities data format (e.g. JSON, XML or SenML), ontology references (such as [i.10] and [i.12]), and interaction patterns support such as observability or polling.

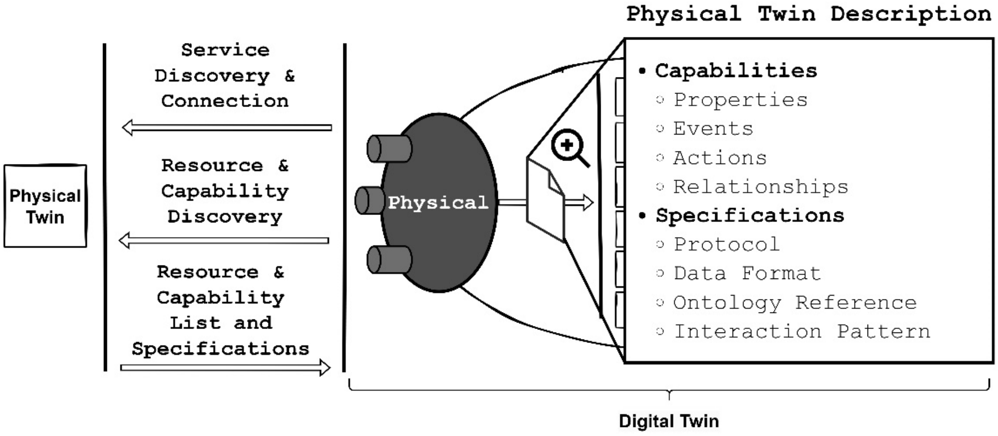

Figure 17: High-level representation of the process of providing a structured description of the physical twin for the DT's model

Protocol Flexibility: The PCC of a DT through its modular PCAs should exhibit a high degree of protocol flexibility and modularity. It should be capable of accommodating a diverse range of communication protocols to interact patterns effectively with various physical assets, including sensors, devices, and machinery. This flexibility ensures compatibility with protocols such as MQTT, CoAP, Modbus, OPC UA, and more. It is fundamental that the PCC shall support multiple communication protocols (PCAs) concurrently. This capability allows the DT to engage with a diverse range of physical assets, devices, and sensors within a single deployment, enhancing versatility and adaptability and also the possibility of digitalizing a single PT using different protocols at the same time (e.g. to communicate heterogeneous information with different priorities) using independent adapters without the need to create custom connector and specific development for the target physical counterpart. Furthermore, for each protocol support security measures shall be supported and configured to protect communication with physical assets. The layer following protocols specification shall include configurable and dynamic authentication, authorization, encryption, and secure key management to prevent unauthorized access and data breaches in the cyber-physical relationship between the DT and the associated PTs. The PCC's capability to support multiple communication protocols concurrently enhances the versatility and adaptability of the DT. For example, a DT deployed in a smart manufacturing environment may need to communicate with diverse assets, such as sensors using MQTT, robotic machinery employing Modbus, and industrial equipment adhering to OPC UA standards. The layer's ability to seamlessly engage with this heterogeneous mix ensures the comprehensive integration of the DT into complex industrial ecosystems:

- **Bidirectional Communication:** The PCC shall facilitate bidirectional communication between the DT and the physical twin. This bidirectional capability is essential to support a twofold purpose: *Continuous Monitoring and Synchronization:*

  - *Live Data Retrieval:* Bidirectional communication ensures the DT's ability to retrieve live and fresh data, properties, telemetry, and events from the PT. This continuous stream of information is indispensable for the DT's capacity to monitor the ever-evolving state of its physical counterpart.

  - *Synchronization Precision:* The bidirectional flow facilitates the synchronization process, where the DT mirrors the physical twin's state dynamically. This precision in synchronization is critical for maintaining an accurate and up-to-date digital representation, aligning seamlessly with the current state of the physical entity.

- *Empowering Dynamic Interaction:*

  - *Action Requests Initiation:* Beyond passive observation, bidirectional communication empowers the DT to initiate and execute action requests directed towards the PT. These actions may stem from the DT's internal processes or be triggered by external entities in the digital domain. For instance, a DT monitoring an industrial machine may actively trigger maintenance actions based on predictive analytics.

  - *Proactive Adaptability:* The ability to influence the physical environment quickly and with low latency fosters adaptability. The DT, armed with bidirectional communication, can actively respond to changes in the physical world, ensuring that its digital representation reflects the most relevant and responsive state. This proactive adaptability is crucial in scenarios where timely interventions are strategic.

- *Fast Insights and Control:*
  - *Fostering Quick Insights:* Bidirectional communication is the linchpin for fostering quick insights into the physical twin's behavior, performance, and condition. This continuous flow of information enables the DT to make informed decisions promptly, contributing to enhanced operational efficiency.
  - *Active Control Mechanisms:* Beyond insights, bidirectional communication equips the DT with active control mechanisms. Whether adjusting parameters, initiating preventive measures, or responding to emergent situations, the DT can actively influence its physical environment. This active control facilitates a more responsive and intelligent cyber-physical system.

By providing bidirectional communication, the PCC through its deployed and active PCAs empowers the DT with the means to both passively observe and actively influence its physical environment, fostering live and fresh insights, control, and adaptability.

Standardized Interfaces: A fundamental requirement that shall be taken into account designing and implementing the PCC and its PCAs structure is the provision of standardized interfaces and communication protocols that facilitate smooth integration with diverse physical entities. These interfaces should serve as abstraction layers, shielding the core DT from the intricacies of underlying communication protocols. Standardization promotes interoperability and ensures a consistent method of interaction. This requirement is integral to the harmonious interaction between the digital and physical realms, ensuring not only a smooth integration process but also fostering interoperability across varied communication protocols and devices:

- *Facilitating Seamless Integration:*
  - *Abstraction of Communication Protocols:* Standardized interfaces act as powerful abstraction layers, shielding the core DT from the intricacies of diverse communication protocols employed by different physical entities. This abstraction simplifies the integration process by providing a uniform, standardized way for the DT to interact with various devices, sensors, and machinery.
  - *Unified Method of Interaction:* By adhering to standardized interfaces, the DT can engage with physical entities using a unified method of interaction. This uniformity eliminates the need for the DT to adapt to the idiosyncrasies of each communication protocol, streamlining the integration workflow and expediting the deployment of the DT across diverse environments.
- *Promoting Interoperability:*
  - *Universal Communication Standard:* Standardization ensures a consistent method of interaction, akin to a universal language that all integrated physical entities can understand. For instance, consider a scenario where a DT, standardized to interact using the MQTT protocol, seamlessly integrates with a variety of sensors and devices that also adhere to the MQTT standard. This interoperability is vital for creating a cohesive cyber-physical ecosystem.
  - *Compatibility Across Devices:* Standardized interfaces promote compatibility across a spectrum of devices, regardless of their specific communication protocols. This enables the DT to effortlessly communicate with sensors using protocols like CoAP, devices using Modbus, or machinery leveraging OPC UA. The DT's ability to speak the language of different devices is crucial for achieving a comprehensive understanding of the entire physical environment.
- *Ensuring Consistency in Communication:*
  - *Predictable Communication Patterns*: Standardized interfaces ensure that the DT can predictably understand and respond to communication patterns. This predictability is crucial for developing robust communication strategies within the cyber-physical system, minimizing uncertainties, and enhancing the overall reliability of the DT's interaction with its physical counterpart.
  - *Consistency in Data Exchange:* Whether receiving telemetry data from sensors or issuing commands to actuators, standardized interfaces ensure consistency in data exchange. This consistency is paramount for creating a reliable and stable cyber-physical relationship, where the DT can confidently rely on standardized interfaces for efficient communication.

For example, considering a smart building where the DT is tasked with monitoring and managing various IoT devices, including temperature sensors, lighting systems, and security cameras. Standardized interfaces implemented in the PCC allow the DT to seamlessly integrate with these diverse devices, each using its specific communication protocol. The DT interacts uniformly with temperature sensors using MQTT, lighting systems using CoAP, and security cameras using HTTP. This standardized approach ensures that the DT comprehensively understands and controls the entire smart building ecosystem, promoting interoperability and ease of integration. In conclusion, the provision of standardized interfaces in the PCC establishes a foundation for the DT's successful integration with diverse physical entities. It not only simplifies the integration process but also promotes interoperability across varied communication protocols. Through standardized interfaces, the DT can engage in a universal language with different devices, ensuring a harmonious and consistent interaction that underpins the reliability and efficiency of the entire cyber-physical system.

Customization & Configuration: The DT shall offer the flexibility for customization and configuration of communication patterns and protocols to align with specific use cases and the unique requirements of individual physical assets. This includes the ability to define custom protocols or configure their usage when needed to address specific scenarios. Aspects that should be taken into account within this strategic requirement are:

- *Flexibility for Unique Use Cases:*
    - *Adaptation to Varied Environments:* The ability to customize and configure communication patterns allows the DT to adapt to a plethora of environments and use cases. For instance, consider a DT deployed in both an industrial manufacturing plant and a smart office building. Customization enables the DT to seamlessly adjust its communication protocols based on the unique requirements of each environment, accommodating diverse sensors, machinery, and devices.
    - *Use Case-Specific Requirements:* Different applications may demand specific communication patterns. In a scenario where a DT is employed for predictive maintenance in an industrial setting, the communication patterns may need to prioritize live data transmission and low-latency interactions. On the other hand, a DT managing energy consumption in a smart building might prioritize energy efficiency and periodic data updates. Customization ensures that the DT can finely tune its communication strategies to cater to the distinct needs of each use case.
- *Defining Custom Protocols for Specialized Scenarios:*
    - *Addressing Custom Requirements:* In certain situations, predefined communication protocols might not perfectly align with the specialized requirements of a given use case. The customization capability enables the DT to define and implement custom protocols tailored to address these niche requirements. For example, in a healthcare setting where a DT monitors medical equipment, the communication protocol might need to adhere to stringent privacy and security standards, necessitating the creation of a custom protocol.
    - *Ensuring Security and Compliance:* Custom protocols can be designed to meet specific security and compliance standards. For instance, a DT deployed in a financial institution, handling sensitive data, might require a custom protocol that incorporates advanced encryption and authentication measures. The ability to define such custom protocols ensures that the DT can adhere to industry-specific regulations and security guidelines.
- *Ensuring Adaptability to Evolving Requirements:*
    - *Changing Physical Twin Configurations:* Physical assets within a deployment may evolve over time, necessitating changes in communication strategies. The customization and configuration capabilities empower the DT to seamlessly adapt to changes in the configuration of physical assets. For instance, when new sensors are introduced or machinery is upgraded, the DT can be reconfigured to accommodate these changes without disruption.
    - *Scalability and Future-Proofing:* As the DT deployment scales or encounters new requirements, the ability to customize and configure communication patterns becomes a strategic advantage. It future-proofs the DT, ensuring that it remains adaptable and scalable, capable of accommodating evolving communication standards and emerging technologies.
- *Configurable Authentication and Security Measures:* Security requirements may vary across different use cases and assets. The customization and configuration capabilities extend to security measures, allowing the DT to dynamically configure authentication, authorization, encryption, and secure key management based on the specific security needs of each scenario.

Imagine a DT deployed in an agricultural setting to monitor and optimize irrigation systems. In this dynamic environment, the water requirements for crops can vary based on factors such as weather conditions, soil moisture levels, and crop types. The DT employs various sensors and actuators to gather live data and control irrigation. The DT's PCC, configured for dynamic customization, allows for quick and live adjustments in communication patterns. When adverse weather conditions are detected, the DT dynamically configures communication to prioritize rapid updates from weather sensors, ensuring timely irrigation adjustments. In periods of stable weather, the DT might switch to a more conservative communication approach, conserving energy and bandwidth. This dynamic customization ensures that the DT can efficiently manage irrigation, responding to the evolving needs of the agricultural landscape. Another configuration-oriented example can be associated to a DT utilized in a healthcare environment where patient monitoring devices are integrated. The DT can be configured to implement stringent security measures, such as encrypted communication channels and multi-factor authentication, ensuring compliance with healthcare data protection regulations. This level of customization is critical to maintaining the integrity and privacy of sensitive healthcare information.

In essence, the customization and configuration capabilities within the PCC and its PCAs empower the DT to tailor its communication strategies to the unique demands of diverse use cases. Whether adapting to changing environments, defining custom protocols, or ensuring dynamic security measures, this flexibility is instrumental in maximizing the effectiveness and relevance of the DT across a spectrum of applications.

Modularity and Reusability: The design and implementation of the PCC should prioritize modularity and reusability of components. This entails creating a framework where support for multiple communication protocols, data formats, and bidirectional communication patterns is structured as modular, interchangeable modules. Each module should encapsulate the logic and functionality required to handle a specific protocol or communication pattern. By adopting a modular approach, efforts spent on implementing support for a particular protocol or pattern can be leveraged across multiple DT instances without redundant development work. This not only minimizes development effort but also enhances code maintainability and scalability. Moreover, it allows for swift adaptation to new communication standards or requirements, as new modules can be introduced, or existing ones updated independently. The modularity and reusability of the PCC and the presence of modular, configurable and interoperable PCAs embedding a shared set of common capabilities such as discoverability, PT description, reading and observation of physical variations and actions execution (as illustrated in Figure 18) are key in achieving flexibility and efficiency, enabling DTs to seamlessly communicate with diverse physical twins and adapt to evolving communication needs across various deployment scenarios. Aspects that should be taken into consideration in terms of modularity and reusability are:

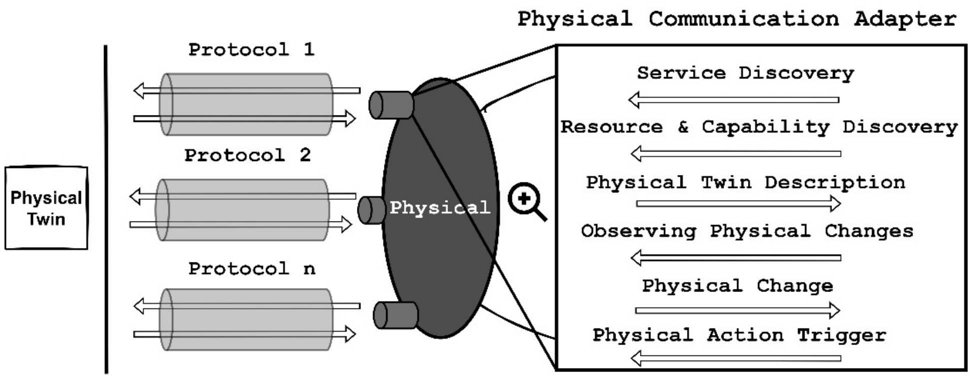

Figure 18: A representation of a DT's physical communication layer characterized by multiple modular and reusable components

- *Framework for Interchangeable Modules:*
    - *Strategic Design Philosophy:* Modularity and reusability dictate the overarching design philosophy of the PCC. The layer is conceived as a dynamic framework where components are organized into modular, interchangeable units, akin to building blocks. This strategic design choice ensures that the PCC can seamlessly evolve and adapt to the diverse and ever-changing landscape of communication protocols, data formats, and bidirectional patterns.
    - *Encapsulation of Logic:* Each module encapsulates the logic and functionality required to handle a specific protocol or communication pattern. For instance, a module might be dedicated to MQTT, another to CoAP, and so forth. This encapsulation ensures that the intricacies of individual communication protocols remain contained within their respective modules, preventing the entanglement of complexities.

- *Cross-Instance Leverage and Effort Optimization:*
    - *Minimizing Redundant Development:* By adopting a modular approach, development efforts invested in supporting a particular protocol or bidirectional pattern can be leveraged across multiple instances of DTs. For instance, if a module is crafted to handle MQTT communication, it can be reused across various DT instances that require MQTT support. This not only minimizes redundant development work but also accelerates the implementation of communication standards.
    - *Enhanced Code Maintainability:* Modularity amplifies code maintainability. Updates or improvements to a specific protocol module can be made independently of others. This granular approach to maintenance ensures that modifications are confined to the relevant module, minimizing the risk of unintended consequences elsewhere in the PCC. It also simplifies debugging and troubleshooting processes.
- *Adaptability to Emerging Standards:*
    - *Swift Adoption to New Standards:* The ever-evolving landscape of communication standards necessitates a DT architecture that can swiftly adapt. Modular components enable the PCC to seamlessly integrate new communication standards or requirements. Introducing a new module or updating an existing one allows the DT to stay abreast of emerging technologies without undergoing a cumbersome overhaul.
    - *Agile Response to Technological Shifts:* Consider the emergence of a novel communication protocol that promises enhanced efficiency. With a modular PCC, integrating this protocol becomes a matter of introducing a dedicated module, ensuring that DTs can readily embrace technological shifts without disruptions to their core functionalities.
- *Efficiency and Scalability:*
    - *Streamlining Development Processes:* The modular and reusable nature of components streamlines development processes. Development teams can focus on optimizing the functionality of individual modules, resulting in more efficient workflows. This efficiency extends to scalability, allowing DT ecosystems to grow seamlessly without incurring proportional increases in development complexity.
    - *Scalable Implementation Across DT Instances:* As the DT ecosystem expands, the modular PCC accommodates the scaling requirements effortlessly. Each new instance can leverage existing modules, promoting consistency in communication implementations and contributing to a unified and scalable architecture.

Consider a scenario where a new data format gains prominence in the IoT landscape. With a modular PCC composed by multiple PCAs, introducing support for this format involves the creation of a dedicated module. Existing DT instances can incorporate this module as needed, enabling them to comprehend and interact with the new data format without necessitating alterations to the core PCC logic. In an industrial setting, where diverse machinery adheres to different communication protocols, a modular PCC facilitates the seamless integration of industry-specific protocols. Whether it is a shift from Modbus to OPC UA or the introduction of a proprietary protocol, the PCC's modularity ensures that DTs remain agile and adaptable.

Summarizing, the strategic adoption of modularity and reusability within the PCC emerges as a linchpin in achieving flexibility, efficiency, and agility in DT ecosystems. This approach ensures that DTs can dynamically communicate with diverse physical twins, adapt to evolving standards, and scale seamlessly across various deployment scenarios. As the digital and physical landscapes continue to evolve, the modularity of the PCC stands as a testament to the resilience and foresight embedded in the architecture of DTs.

Decoupling Complexity: An essential requirement is the decoupling of physical communication characteristics, such as communication protocols and data formats, from the core of the DT and its model responsible for the replication of the associated Physical Twin. It should ensure that the core DT receives information in a standardized and normalized format, abstracting the specifics of various communication protocols and patterns. This decoupling simplifies the core DT's processing and enhances its adaptability. Core aspects that should be considered are:

- *Standardized Information Flow:*
    - *Abstraction of Protocols:* The decoupling mechanism ensures that the core DT is shielded from the specifics of diverse communication protocols. For instance, whether the PT communicates via MQTT, CoAP, or Modbus, the core DT remains agnostic to these intricacies.

- *Normalized Data Formats:* Similarly, the decoupling extends to data formats. The core DT receives information in a standardized and normalized format, abstracting the variability in how different physical assets might present their data. This normalization facilitates uniform processing, irrespective of the data's origin.

- *Enhanced Adaptability:*

  - *Simplified Processing Logic:* Decoupling complexity simplifies the processing logic embedded in the core DT. It allows the core to operate on information presented in a consistent format, reducing the need for intricate protocol-specific parsing and decoding logic.

  - *Rapid Integration of New Protocols:* When a new communication protocol gains prominence or an existing one evolves, the core DT's adaptability shines. The introduction of a new protocol involves updating the corresponding module within the PCC, leaving the core DT untouched. This adaptability ensures that the core remains resilient to the dynamic nature of communication technologies.

- *Adaptation to Evolving Standards:*

  - *Evolutionary Nature of Protocols:* Communication protocols are subject to evolution and standardization. The decoupling mechanism allows the core DT to adapt seamlessly to these changes. Whether there's a transition from an older version of a protocol to a newer one or a migration to a completely different standard, the core DT remains insulated from these transitions.

  - *Future-Proofing the DT*: In the context of cybersecurity, where encryption and authentication standards evolve, the decoupling of the core DT ensures that updates to security protocols can be implemented in the PCC without necessitating modifications to the core logic. This future-proofs the DT, allowing it to embrace the latest security measures without disrupting its foundational processes.

- *Reduced Development Overheads:*

  - *Focused Development Efforts:* The separation of concerns facilitated by decoupling complexity allows development efforts to be focused. Protocol-specific or data format-specific modifications are confined to the PCC, minimizing the risk of unintended consequences in the core DT. This focused development enhances efficiency and accelerates adaptation to new communication standards.

- *Incremental Updates and Maintenance:* Updating or modifying communication protocols becomes an incremental process. Developers can introduce updates to the PCC modules independently of the core DT, ensuring that maintenance tasks are targeted and do not disrupt the ongoing operations of the DT.

Consider an Industrial Internet of Things (IIoT) [i.5] scenario where various machines within a manufacturing plant communicate using different protocols, such as Modbus and OPC UA. By decoupling the communication specifics from the core DT, the DT seamlessly integrates diverse data streams. The core DT, abstracted from protocol intricacies, can process and synchronize information without being entangled in the diverse communication methods employed by different machines. On the other hand, in an agricultural setting, where sensors on different types of equipment generate data, decoupling allows the core DT to receive and process this data in a standardized manner. Whether it is soil moisture levels communicated in JSON or temperature readings in XML, the core DT remains focused on understanding and replicating the essence of the PT without concern for the data's original format.

Decoupling complexity within the PCC should be a cornerstone principle in sculpting agile, adaptable DT. By shielding the core DT from the intricacies of communication protocols and data formats, this principle streamlines processing, fortifies adaptability, and future-proofs DTs against the dynamic landscape of evolving standards. In a world where digital dexterity is paramount, decoupling complexity stands as a beacon illuminating the path to a resilient and responsive DT architecture and framework.

Live Data Processing: The PCC should enable and support when required live or even real-time data processing, enabling the DT to respond swiftly to changes in the physical environment building and maintaining an accurate replica of the PT when needed and accordingly to the target application scenario. This capability should be included in the design of DT architecture and framework but might be enabled and deployed only when it is needed following both PT capabilities and DT design requirements and application goals. This includes the ability to detect and react to physical events, synchronize states, transmit relevant information and execute actions with a low delay allowing live and quick interactions. Aspect that should be taken into account in the design are the following:

- *Swift Responsiveness to Physical Changes:*
  - *Dynamic Adaptability:* The core essence of real-time data processing lies in enabling the DT to adapt dynamically to changes in the physical world. This includes responding promptly to events, fluctuations in sensor readings, or alterations in the state of physical entities.
  - *Synchronized State Maintenance:* Real-time data processing ensures that the DT's internal state remains synchronized with the Physical Twin. This synchronization is pivotal in maintaining an accurate representation of the physical counterpart, aligning the digital model with the ever-evolving conditions of the physical environment.
- *Conditional Activation of Live Processing:*
  - *Contextual Activation:* While the capability for real-time data processing is a fundamental design consideration, its activation should be contextually driven. Not all scenarios demand real-time responsiveness. Therefore, the DT architecture should provide the flexibility to enable or disable this feature based on the specific requirements of the PT, the goals of the DT design, and the application scenario.
  - *Adapting to Varying Application Needs:* For instance, in scenarios where the PT exhibits dynamic and rapidly changing conditions, such as in autonomous vehicles or smart city grids, real-time processing may be crucial. Conversely, in scenarios where periodic or batch processing is sufficient, such as in certain industrial monitoring applications, real-time processing may be selectively activated.
- *Dynamic Detection and Reaction to Events:*
  - *Event-Driven Architecture:* Real-time data processing equips the DT with the capability to operate on an event-driven paradigm. Events could range from sudden changes in sensor readings, equipment malfunctions, to external triggers initiated by the digital realm or external applications.
- *Seamless Information Transmission:*
  - *Latency Reduction:* Real-time data processing minimizes latency in transmitting information between the DT and the PT. This is particularly crucial in applications where timely decision-making or immediate actions are imperative.
- *Low-Latency Action Execution:*
  - *Timely Decision-making:* The capability to execute actions in real-time extends the DT's ability to influence its physical environment promptly. This involves not only observing and processing real-time data but also orchestrating actions that impact the PT.
- *Aligning with Application Goals:*
  - *Scenario-Specific Implementation:* The implementation of real-time data processing should align with the overarching goals of the application. In scenarios where real-time insights are critical, such as in healthcare monitoring, the DT's ability to process data in real-time enhances its utility.
  - *Balancing Resources and Requirements:* However, in scenarios where real-time processing might impose resource constraints or is unnecessary, the DT architecture should provide the flexibility to balance the need for real-time responsiveness with resource optimization.

In a smart building environment, real-time processing allows the DT to swiftly respond to events like sudden temperature spikes, triggering actions such as adjusting HVAC systems or alerting facility managers. This dynamic response ensures that the digital representation aligns with the real-time dynamics of the building's physical attributes. Another example can be a predictive maintenance scenario for manufacturing machinery, real-time processing enables the DT to detect anomalies, transmit alerts, and initiate maintenance actions without delay. This seamless transmission ensures that the physical machinery's condition is accurately reflected in the DT, guiding timely intervention. In autonomous vehicles, real-time data processing facilitates swift decision-making based on the current state of the vehicle and its surroundings. This includes instant adjustments to navigation, speed, or even emergency responses, ensuring that the DT remains in harmony with the dynamic conditions of the physical vehicle.

In essence, Real-time Data Processing within the PCC emerges as a pivotal enabler for DTs to embody dynamic responsiveness. By facilitating swift adaptation to changes, synchronized state maintenance, and seamless interaction with the physical realm, this capability ensures that the DT remains attuned to the real-time nuances of its Physical Twin. Conditional activation, event-driven architecture, and alignment with application goals further underscore the nuanced implementation of real-time data processing in crafting versatile and efficient DTs.

Scalability: The PCC together with its PCAs should be scalable to accommodate the growth of physical assets and their interactions. Scalability is crucial for handling larger and more complex DT deployments effectively. Fundamental aspects that should be considered with respect to scalability are:

- *Dynamic Expansion of Physical Assets:*
  - *Unpredictable Growth:* The nature of physical assets in diverse domains often involves unpredictable growth patterns. For instance, in a smart city infrastructure, the addition of new sensors, devices, or infrastructure components may occur organically over time.
  - *Scalable Adapters:* Scalability in the PCC and PCAs entails a design that allows for the seamless integration of new physical assets without necessitating a fundamental overhaul of the existing architecture. Adapters should be modular and adaptable, ensuring they can efficiently engage with a varying number of physical entities.
- *Interactions Across an Expansive Ecosystem:*
  - *Interconnected Networks:* Scalability extends beyond the sheer number of physical assets to encompass the complexity of interactions within a broader ecosystem. In industrial settings, for instance, as the number of machines increases, the interconnections between these machines and their associated DTs amplify.
  - *Distributed Scalability:* The PCC and PCAs should be architected to support distributed scalability, allowing for the seamless incorporation of new nodes, whether they be sensors, machinery, or entire facilities. This ensures that the DT ecosystem can gracefully grow to meet the demands of an expanding industrial landscape.
- *Resource Optimization and Efficiency:*
  - *Economical Resource Allocation:* Scalability is not merely about accommodating growth but also optimizing resource allocation. The PCC should be designed to efficiently utilize computing resources, bandwidth, and storage, particularly as the scale of the DT ecosystem magnifies.
  - Adaptive Resource Scaling: As an example, in an agricultural context where DTs monitor diverse aspects of a large-scale farm, scalable PCAs should dynamically allocate resources based on the specific monitoring needs of different zones. This adaptive scaling ensures that resources are judiciously utilized, enhancing efficiency.
- *Challenges in Scalability Implementation:*
  - *Data Volume Management:* As the number of physical assets and their interactions burgeon, managing the volume of data generated becomes a critical challenge. Scalability in the PCC necessitates robust data management strategies, including data aggregation, compression, and prioritization.
  - *Network Traffic Considerations:* In scenarios where DTs communicate over networks, scalability requires careful consideration of network traffic. Efficient data routing, load balancing, and network optimization become imperative to prevent congestion and latency issues.

- *Ensuring Consistency in Scalable Deployments:*
  - *Maintaining Coherence:* Scalable deployments should not compromise the coherence of the DT ecosystem. The PCC and PCAs should ensure that the state of the DT, reflecting the physical reality, remains consistent across the entirety of the scalable deployment.
- *Adaptability to Evolving Communication Needs:*
  - *Future-Proofing Communication Protocols:* Scalability is not just about current needs but also anticipating future communication requirements. The PCC should be agile enough to embrace new communication protocols, standards, and data formats that may emerge over time.
  - *Intelligent Routing for Future Integration:* As an illustration, in a manufacturing setting where robotic systems are continually evolving, scalable PCAs should employ intelligent routing mechanisms. This allows the DT to incorporate new robotics technologies seamlessly without disrupting existing communication pathways.

In smart grid applications for example, scalability demands the addition of new sensors and energy distribution nodes. The PCC should orchestrate this scalability while ensuring that real-time data from these new assets integrates seamlessly into the broader DT, maintaining a unified view of the entire energy grid. In essence, scalability in the PCC and PCAs of a Digital Twin transcends numerical expansion; it encapsulates the holistic readiness to embrace a burgeoning ecosystem of physical assets, interactions, and evolving communication dynamics. By addressing challenges, optimizing resources, and ensuring adaptability, a scalable PCC becomes the linchpin in navigating the expansive horizons of Digital Twin deployments, ensuring their effectiveness in diverse and growing domains.

Communication Awareness: Tailored to the specific application requirements, the PCC together with the implemented and active PCAs shall incorporate mechanisms for real-time performance monitoring, management, and control of communication parameters such as latency, packet loss, and other pertinent metrics. The PCC shall provide comprehensive techniques for monitoring the health and performance of physical communication according to the nature of the supported communication protocols and interaction patterns. This becomes especially pivotal in scenarios where instantaneous or near-instantaneous communication holds critical importance or more generally where digital service relies on the quality of the DT and in its capabilities of reflecting the physical world. The layer's ability to maintain awareness of these factors is essential for enhancing the DT's interaction with its physical counterpart and ensuring the creation of a digital replica aligned with the precise needs of the target application domain. Furthermore, external applications shall be empowered with the capability to gauge the quality of the cyber-physical connection and its potential variations over time. In essence, communication awareness in the PCC through its PCAs transforms the cyber-physical relationship from a static connection to a dynamic and responsive interplay. By tailoring monitoring techniques, facilitating real-time adjustments, and empowering external applications, the PCC becomes a conduit for not just data transfer, but for orchestrating an intricate performance that aligns precisely with the evolving needs of the twin's application domain.

Reliability, Resilience: Ensuring reliable communication with physical entities is paramount. According to the context and the nature of DTs the PCC shall incorporate mechanisms for redundancy, failover, and error handling to maintain consistent communication, especially in mission-critical applications. Robust error handling and recovery mechanisms are essential to maintain communication integrity, particularly in scenarios where communication disruptions can occur. Resilience in the PCC involves designing redundant pathways that can adapt to varying scenarios. For example, in a smart building DT, if communication with environmental sensors is disrupted due to network issues, redundant pathways can include local communication within the building infrastructure. Resilience is further bolstered by dynamic failover mechanisms. In agriculture, where DTs monitor irrigation systems, the PCC can employ failover strategies that dynamically switch to alternative communication channels if primary channels experience disruptions due to weather conditions or equipment malfunctions. Consider for example also a smart grid scenario where DTs oversee energy distribution. A robust failover mechanism in the PCC ensures that if communication with a specific energy node is interrupted, the DT swiftly redirects communication to an alternative node, preventing data gaps and ensuring real-time monitoring. In industrial automation, where communication with machines is pivotal, the PCC may incorporate redundancy not only in network pathways but also in communication protocols. If one protocol encounters issues, the DT can seamlessly switch to an alternative protocol to ensure continuous communication.

By comprehensively addressing these multifaceted requirements, the PCC of a DT, in tandem with its indispensable PCAs, emerges as the linchpin that masterfully orchestrates the convergence of the digital and physical worlds. This nuanced orchestration is rooted in the layer's structured provision of a detailed and standardized description of the associated physical twin. This serves as the foundational schema that meticulously captures the intricacies of the physical asset's characteristics, properties, events, actions, and relationships. The PCC's protocol flexibility stands out as a pivotal trait, enabling seamless interaction with an extensive array of communication protocols, from MQTT and CoAP to Modbus and OPC UA, concurrently. This flexibility not only enhances versatility but also ensures that diverse physical assets, including sensors, devices, and machinery, can be engaged within a singular deployment framework.

Bidirectional communication shall be facilitated by the PCC, empowering the DT to not only passively observe but actively influence its physical environment in real-time. Standardized interfaces act as fundamental enablers, shielding the core DT from the intricacies of underlying communication protocols, thereby promoting interoperability and consistency across diverse physical entities. Moreover, the flexibility for customization and configuration ensures adaptability to specific use cases, allowing for the definition of custom protocols and configurations tailored to unique requirements.

The modularity and reusability of components in the PCC design and implementation, showcased in the use of interchangeable modules, ensure that efforts invested in supporting specific protocols or communication patterns can be leveraged across various DT instances. This modular approach not only minimizes development efforts but also augments code maintainability and scalability. Decoupling the complexity of physical communication characteristics from the core DT's model simplifies processing and enhances adaptability, crucial for efficient functioning. The emphasis on real-time data processing, scalability, and reliability ensures the PCC's effectiveness in diverse scenarios, from industrial IoT settings to healthcare applications.

The layer's robust error handling and recovery mechanisms, coupled with its awareness of communication parameters, fortify the reliability and resilience of communication with physical entities, making it particularly apt for mission-critical applications. As a result, the PCC's comprehensive techniques for monitoring the health and performance of physical communication in real-time become essential for maintaining a high-quality cyber-physical connection. This heightened awareness not only serves the DT's immediate requirements but also empowers external applications to gauge the quality of the connection and anticipate potential variations over time.

In essence, the PCC, fortified by a spectrum of meticulously addressed requirements, stands as a dynamic conduit that not only connects the digital and physical realms but also crafts an adaptive and responsive bridge. It ensures the efficient, reliable, and adaptable communication of DTs with their physical counterparts while navigating the evolving intricacies of diverse communication needs and deployment scenarios. DT as a robust and agile entity within the broader cyber-physical landscape.

5.3 Digital Communication Requirements

The DCC of a DT together with its modular structure and the presence of different adapters DCAs serves as a crucial bridge between the twin's core model and the broader digital environment. This layer should meet a set of distinct requirements to ensure effective communication, interoperability, and seamless integration within digital ecosystems.

DT's Digital Representation: The DT's DCC shall abstract raw data and information from its core model, providing a simplified representation of the DT's state (properties, events, action and relationships), capabilities and actions to external digital entities through the representation through one or multiple format of a Digital Twin Description (DTD) as illustrated in Figure 19. This abstraction shields external systems from unnecessary complexity, making interactions more efficient and straightforward. To support interoperability, the layer shall be capable of translating the DT's internal data into various formats and protocols commonly used in the digital world and enable the call of actions available on the PT and replicated on the DT. This flexibility enables the DT to communicate effectively with a wide range of applications and services.

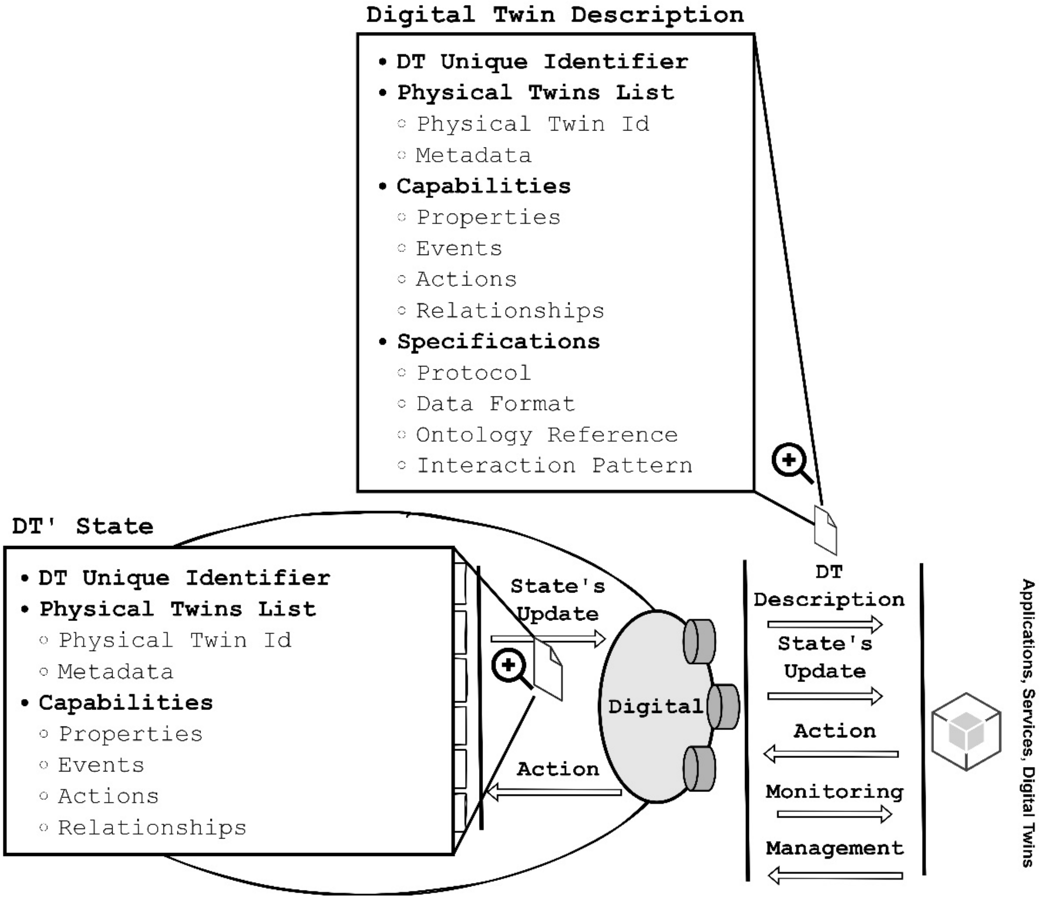

Figure 19: Schema depicting the relationship between DT's state and its digital representation

The primary objective of this abstraction is to shield external systems from unnecessary complexity, ensuring that the information presented is concise, relevant, and easily digestible. By presenting a structured and interoperable version of the DT's state, capabilities, and available actions, the DCC enhances the efficiency and effectiveness of interactions between the DT and external digital entities through one or multiple DCAs according to the target deployment and the associated digital applications interacting with the twins. This becomes particularly crucial in scenarios where live or even real-time or near-real-time communication is essential, such as in industrial automation or healthcare applications.

To support seamless interoperability, the DCC shall exhibit a high degree of flexibility in translating the DT's internal data into various formats and protocols commonly used in the digital world. This includes widely recognized standards such as JSON, XML, or even protocol-specific representations like MQTT or RESTful APIs. The layer's capability to interpret and adapt the DT's information into diverse formats ensures compatibility with a broad spectrum of applications and services. For instance, if a manufacturing plant employs a Manufacturing Execution System (MES) that communicates via RESTful APIs, the DCC can adeptly translate the DT's data into the required format, enabling effective communication and coordination between the DT and the MES.

Furthermore, the DCC should facilitate the invocation of actions available on the PT and replicated on the DT. This means that external digital entities can trigger actions on the DT, which are then mirrored in the physical world through the associated PT. For instance, in a smart building scenario, an external application controlling the HVAC (Heating Ventilation Air Conditioning) system can send a command to the DT via the DCC to adjust the temperature settings. The DCC translates this digital command into a format understood by the DT's core model, ensuring the corresponding action is executed on the PT, thereby influencing the physical environment.

In summary, the abstraction capabilities of the DT's Digital Communication Channel play a pivotal role in making the DT's information accessible and comprehensible to external digital entities. By providing a simplified representation of the DT's state and supporting diverse digital formats and protocols, the DCC fosters efficient and versatile communication, enabling the DT to seamlessly integrate and collaborate with a wide array of applications and services in the digital ecosystem.

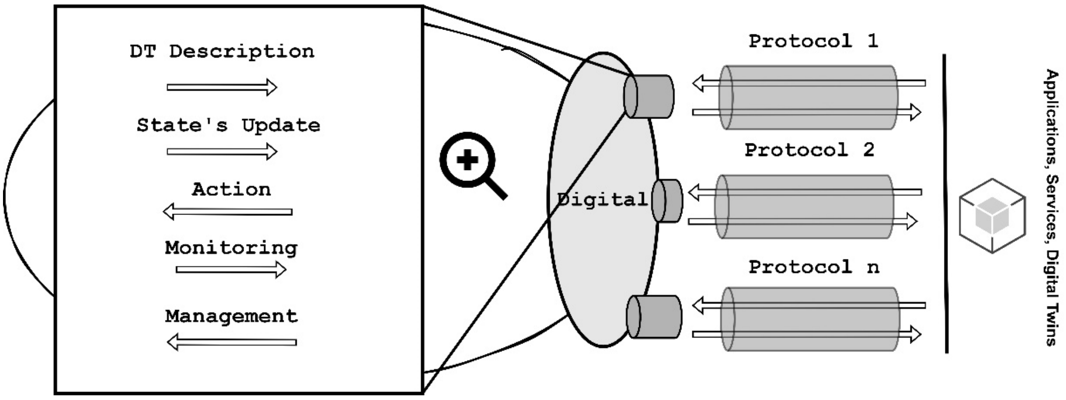

Figure 20: Modularity to support extensibility and reusability in DT's interaction with the digital space

Bidirectional Communication: The DCC shall enable external applications to access and consume DT data, including properties, telemetry, events, and other relevant information as reported through its adapter base structure in Figure 20. In addition to data retrieval, the layer shall support bidirectional communication by allowing external applications to trigger actions within the DT, affecting its behavior and physical counterpart if necessary. To ensure data security and privacy, the DCC and its DCAs shall support encryption protocols and secure communication channels, safeguarding sensitive information during transmission. Access control mechanisms, including user authentication and authorization, shall be in place to regulate external entities' interactions with the DT, preventing unauthorized access.

Consider an industrial IoT scenario where a DT monitors and manages a manufacturing plant's machinery. The DCC, through its bidirectional communication capabilities, allows an external maintenance application to retrieve real-time telemetry data, such as machine temperatures and vibration levels, for predictive maintenance. Simultaneously, the maintenance application can leverage bidirectional communication to trigger an action within the DT, such as adjusting the operating parameters of a machine to optimize performance or scheduling preventive maintenance routines.

To ensure the integrity and security of the bidirectional communication, the DCC and its DCAs should support robust encryption protocols and the establishment of secure communication channels. In the context of a healthcare application, where a DT represents a patient monitoring system, bidirectional communication might involve a remote healthcare provider accessing vital sign data and remotely adjusting the monitoring thresholds in response to the patient's evolving health condition. Encryption ensures that sensitive patient information remains confidential during transmission, safeguarding privacy and complying with healthcare data protection regulations.

Access control mechanisms are paramount in regulating external interactions with the DT. Authentication and authorization protocols become crucial in scenarios where a smart home DT communicates bidirectionally with external applications like a voice-activated assistant [i.7]. This ensures that only authorized entities can trigger actions within the DT, preventing unauthorized access or manipulation. For instance, an authorized user might remotely instruct the DT to adjust the home thermostat, providing a seamless and secure bidirectional communication experience.

Moreover, bidirectional communication enhances the collaborative capabilities of the DT in diverse applications. In a smart grid deployment, the DCC enables external applications to not only retrieve energy consumption patterns but also instruct the DT to implement load-shedding strategies during peak demand. This bidirectional interaction optimizes energy distribution and contributes to grid stability.

The bidirectional communication capabilities of the DCC are especially valuable in dynamic environments. In autonomous vehicle systems, for instance, the DCC allows external traffic management applications to retrieve real-time data about vehicle locations and conditions while also providing a means to send directives to alter routes or adjust driving behavior based on real-time traffic conditions.

In summary, Bidirectional Communication within the DCC empowers external applications to seamlessly access and retrieve relevant data from the DT while also triggering actions within the DT. Examples from diverse domains, such as industrial IoT, healthcare, smart homes, energy management, and autonomous vehicles, highlight the versatility and significance of bidirectional communication in enhancing the functionality and responsiveness of DTs in a wide array of applications. The layer's support for encryption, secure channels, and access controls ensures the reliability, security, and privacy of these bidirectional interactions.

Extensible Protocols & Interaction Patterns: The DCC through its DCAs shall support multiple protocols and interaction patterns, such as Publish/Subscribe through MQTT, RESTful APIs with HTTP and CoAP, WebSocket, and more. This diversity allows the DT to engage with different digital applications, each employing its preferred interaction method. To accommodate various digital protocols (e.g. HTTP, MQTT, CoAP), the layer should remain protocol-agnostic, ensuring that it can seamlessly adapt to the communication requirements of different scenarios supporting for example multiple interaction patterns at the same time.

Consider a smart city deployment where a DT is responsible for monitoring and optimizing various urban services. The extensible nature of the DCC enables the DT to communicate with a city traffic management application using the Publish/Subscribe model via MQTT for real-time updates on traffic conditions. Simultaneously, it interacts with a city services portal using RESTful APIs over HTTP, allowing administrators to retrieve aggregated data for long-term planning. The DCC 's support for diverse interaction patterns accommodates the varied communication needs of different digital applications within the smart city ecosystem.

In an IIoT context, the DT could be part of a smart manufacturing system. The DCC 's extensibility allows it to utilize WebSocket communication for low-latency control signals exchanged with a centralized manufacturing execution system. Meanwhile, it interacts with a supply chain management application using CoAP for efficient and lightweight communication. This flexibility in supporting multiple protocols and interaction patterns ensures that the DT seamlessly integrates into a diverse IIoT landscape, where different applications may have distinct communication preferences.

The protocol-agnostic nature of the DCC is crucial for interoperability. In a smart home scenario, the DT communicates with a voice-controlled assistant using WebSocket for real-time command responses, while simultaneously interacting with a home automation platform through CoAP for resource-efficient data exchange. The layer's ability to accommodate these different protocols concurrently ensures that the DT can adapt to the specific communication requirements of each application, contributing to a cohesive and integrated smart home environment.

Furthermore, the DCC 's extensibility is exemplified in a healthcare application where a patient monitoring DT communicates with a cloud-based health analytics platform. Here, the DCC supports both MQTT for instantaneous telemetry updates and RESTful APIs for retrieving historical patient data during routine check-ups. The layer's capacity to seamlessly incorporate various interaction patterns allows the healthcare system to leverage the strengths of each protocol based on the immediacy and nature of the data being exchanged.

In summary, the DCC 's Extensible Protocols & Interaction Patterns empower the DT to engage with a diverse range of digital applications using protocols such as MQTT, HTTP, CoAP, and WebSocket. Examples from smart cities, industrial IoT, smart homes, and healthcare illustrate how this extensibility fosters interoperability, adaptability, and efficient communication in different application domains. The DCC's protocol-agnostic design ensures that it can concurrently support multiple interaction patterns, accommodating the varied communication preferences of external digital entities.

Real-time Communication: For applications demanding real-time data updates, the DCC should facilitate the instantaneous transmission of DT state changes to digital applications, enabling timely decision-making and actions. In scenarios where asynchronous communication is more suitable, the layer should allow external systems to retrieve DT information at their convenience, fostering flexibility in data retrieval.

Consider an example in the context of predictive maintenance for industrial machinery. The DT, equipped with sensors and monitoring devices, continuously captures real-time data on the health and performance of the physical machinery. In a scenario where a critical fault is detected, the DCC enables immediate communication of this information to a maintenance application. This real-time communication allows the application to trigger instant alerts, schedule urgent maintenance tasks, and potentially prevent costly downtime.

In autonomous vehicle systems, real-time communication is essential for ensuring the safety and efficiency of operations. The DT embedded within a connected car continuously gathers data from sensors, cameras, and other sources. When the DT detects a sudden change in road conditions or identifies potential safety risks, the DCC facilitates instant communication of this information to the vehicle's control systems. This real-time exchange enables the vehicle to make rapid adjustments to its speed, trajectory, or other parameters to navigate safely through dynamic environments.

In the healthcare domain, where patient monitoring and response are critical, the DT of a medical device can utilize real-time communication through the DCC. For instance, in a remote patient monitoring system, the DT collects real-time data on vital signs. When an anomaly is detected, the DCC ensures immediate communication of this information to healthcare applications, allowing healthcare professionals to intervene promptly and provide necessary assistance.

While real-time communication is vital in certain applications, there are scenarios where asynchronous communication is more suitable. For example, in a smart city deployment where environmental sensors continuously monitor air quality, traffic conditions, and energy consumption, the DCC can support asynchronous communication. Digital applications can retrieve aggregated DT information periodically, enabling them to analyse trends, generate reports, and optimize city operations without requiring immediate updates.

In summary, the real-time communication capability of the DCC is tailored to the specific requirements of diverse applications. Examples from predictive maintenance, autonomous vehicles, and healthcare showcase how real-time communication enables timely responses and actions in critical scenarios. Additionally, the flexibility to support asynchronous communication ensures adaptability to applications where real-time updates may not be essential, providing a versatile solution for a wide range of deployment scenarios.

Dynamic Adaptation & Configuration: The layer shall be adaptive, capable of reconfiguring its communication parameters, data formats, and protocols according to the specific requirements of the DT's deployment scenario. The DT shall be able to efficiently re-configure and manage the DCC together with the available DCAs to align with evolving operational needs, ensuring seamless integration and communication.

Consider a scenario in the context of a smart building management system. The DT embedded in the building's infrastructure continuously monitors environmental conditions, occupancy patterns, and energy consumption. During a major renovation or structural modification, the building's architecture and sensor placements may undergo significant changes. The DCC 's dynamic adaptation allows it to automatically reconfigure communication parameters, ensuring that the DT seamlessly integrates with the modified infrastructure. This adaptability ensures that the DT continues to provide accurate and relevant information to applications managing lighting, HVAC systems, and overall building efficiency.

In the context of IIoT, where machinery and equipment configurations may change due to production line modifications, the DCC 's dynamic adaptation becomes essential. For instance, in a manufacturing facility deploying a DT for predictive maintenance, if a new machine is introduced or an existing one is replaced, the DCC can dynamically adjust its communication parameters and protocols to accommodate the changes. This ensures that the DT remains in sync with the physical machinery, allowing the predictive maintenance application to operate seamlessly without manual reconfiguration.

Moreover, in the field of precision agriculture, where sensor networks are deployed across vast agricultural landscapes, the DCC's dynamic adaptation is critical. Changes in crop types, weather conditions, or the introduction of new sensor technologies may necessitate adjustments in communication parameters. The DCC can autonomously adapt to these changes, allowing the DT to efficiently communicate with sensors, drones, and agricultural machinery, providing real-time insights for precision farming practices.

The dynamic adaptation capability of the DCC extends to scenarios where security protocols need to be adjusted based on emerging threats or compliance requirements.

EXAMPLE: In a healthcare application using a DT for patient monitoring, if there are updates to security standards or regulatory guidelines, the DCC can dynamically reconfigure encryption protocols and access control mechanisms to ensure compliance and safeguard sensitive patient data.

In summary, the DCC 's dynamic adaptation and configuration capabilities are exemplified in scenarios ranging from smart building management and IIoT to precision agriculture and healthcare. The ability to autonomously adjust communication parameters ensures that the DT remains agile and responsive to changes in its deployment environment, facilitating continuous, effective communication and integration with evolving operational needs.

Management and Monitoring: The layer shall provide interfaces for DT management, allowing authorized entities to control and configure the DT's behavior, including the execution of communication modules and/or adapt model behaviors and digital replication parameters. It should support live monitoring of communication performance metrics, including latency, packet loss, and other relevant indicators together with the support for providing information about cyber-physical awareness and DT's life cycle evolution potentially through different DCAs on the DCC.

Consider a smart energy grid employing a DT to monitor and manage electricity distribution. In this scenario, the DCC 's management interfaces would empower utility operators to configure the DT's behavior to optimize energy flow during peak demand hours. The operators could dynamically adjust communication modules to respond to grid fluctuations, ensuring efficient energy distribution and minimizing latency in critical scenarios.

Live monitoring is another essential facet of the DCC, providing continuous insights into communication performance metrics. This includes tracking crucial indicators such as latency, packet loss, and other relevant parameters. In the context of an autonomous vehicle fleet utilizing DTs for predictive maintenance, live monitoring enables fleet managers to assess the health of the communication channels. If there is a sudden increase in latency or a spike in packet loss, the DCC 's monitoring capabilities would trigger alerts, allowing proactive intervention to prevent potential disruptions in the fleet's operations.

Moreover, the DCC shall support the dissemination of information regarding cyber-physical awareness and the DT's life cycle evolution. This information is invaluable for stakeholders seeking insights into the DT's current state, its interactions with physical entities, and any ongoing changes in its life cycle. In a smart manufacturing setting, for instance, where a DT is employed for production optimization, monitoring of cyber-physical awareness enables plant managers to track the synchronization of the DT with physical machinery. This awareness is vital for ensuring that the DT accurately reflects the current state of the production line, fostering efficiency and minimizing downtime.

The management interfaces of the DCC should be designed to align with the specific needs of diverse applications. In a healthcare environment where a DT is utilized for patient monitoring, authorized medical professionals can leverage the DCC's management interfaces to configure communication modules and adapt DT behaviors to suit individual patient requirements. The possibility to have a live monitoring, in this context, enables healthcare providers to receive critical data promptly, supporting timely decision-making and intervention.

In summary, the DCC's management and monitoring capabilities are versatile and applicable across various domains, including smart energy grids, autonomous vehicles, smart manufacturing, and healthcare. By providing interfaces for dynamic DT management and monitoring of communication performance and awareness parameters, the DCC ensures that the DT operates optimally, adapts to changing conditions, and maintains a synchronized and accurate reflection of its physical counterpart.

Extensibility and Reusability: An extensible, modular architecture should be employed, enabling the addition of new communication adapters, protocols, and formats with ease. Once a communication adapter is developed, it should be reusable across multiple DT instances and deployments, minimizing development effort and enhancing code reusability within the DCC and in particular through the presence of modular and reusable DCAs within different twin instances.

Consider a scenario where a DT is deployed in a smart manufacturing environment. The extensible architecture of the DCC allows for the addition of a new communication adapter specifically designed to interface with advanced robotic systems introduced into the manufacturing process. This adapter, once integrated, becomes a reusable component across various DT instances within the manufacturing facility, ensuring consistent communication with different robotic systems.

Moreover, the extensibility of the DCC is crucial for incorporating support for emerging communication protocols and standards. In an evolving technological landscape, new protocols may gain prominence, and the DCC's extensibility enables the seamless integration of these protocols without requiring a complete overhaul of the communication infrastructure. For instance, the DCC can be extended to support the adoption of a novel communication protocol like a future version of MQTT or an emerging standard in the IoT or IIoT domain.

Reusability is equally paramount in achieving efficiency and reducing development effort. Once a communication adapter is developed and tested for compatibility with a specific type of device or system, it should be easily reusable across multiple DT instances and deployments. In the context of smart buildings, where a DT is employed for energy management, a communication adapter designed for interfacing with smart thermostats can be reused across various buildings, streamlining the deployment process and ensuring consistency in communication across diverse environments.

Additionally, the reusability of communication adapters contributes to the scalability of DT deployments. As the number of connected devices and physical assets grows, having a library of reusable communication adapters simplifies the integration process for new devices, accelerating the expansion of the DT ecosystem.

The extensibility and reusability of the DCC are not only about adding new DCA but also about evolving existing ones. For example, as security standards evolve, the DCC's extensible architecture allows for the incorporation of updated security protocols into existing communication adapters, ensuring that the DT remains resilient to emerging cybersecurity threats.

In conclusion, an extensible and reusable architecture for the DCC is foundational for adapting to changes in communication requirements, supporting the integration of new technologies, and efficiently managing the complexity of diverse DT deployments. This approach facilitates a future-proof communication layer that can evolve alongside technological advancements and diverse application scenarios.

Error Handling and Reporting: The DCC shall incorporate error handling mechanisms to address communication failures gracefully, ensuring the DT's stability and resilience. Comprehensive logging and reporting capabilities shall be available to record communication-related events, aiding in troubleshooting and performance optimization.

One key aspect of robust error handling is the ability to detect and respond to communication failures promptly. For instance, in a smart city deployment where a DT oversees various connected infrastructure components, the DCC's error handling mechanisms might include sophisticated algorithms that can identify network disruptions or device unavailability. When such issues arise, the DCC can trigger predefined recovery procedures, minimizing downtime and maintaining the DT's responsiveness.

Comprehensive logging is indispensable for understanding the intricacies of communication failures. The DCC shall generate detailed logs of communication-related events, capturing information such as error types, timestamps, and affected components. In a scenario where a DT manages a fleet of autonomous vehicles, the logging system could record instances of communication breakdowns with specific vehicles, providing insights into potential issues like connectivity gaps or device malfunctions.

The reporting capabilities of the DCC further enhances its utility. A well-designed reporting system enables live and synchronized visibility into the health of the communication infrastructure. For instance, in an industrial setting where a DT oversees a network of machinery, the DCC's reporting capabilities could generate alerts or notifications when certain communication metrics deviate from predefined thresholds. This proactive reporting allows operators to address potential issues before they escalate, contributing to the overall reliability of the DT-driven system.

Moreover, error handling should not only focus on technical failures but also consider scenarios where the DT receives inconsistent or inaccurate data. In an agricultural context where a DT monitors soil conditions through various sensors, the DCC 's error handling might involve data validation checks. If sensor readings fall outside expected ranges or exhibit irregular patterns, the DCC could flag these instances as potential data anomalies, prompting further investigation or recalibration.

The user interface for error handling and reporting should be user-friendly, providing clear insights into the nature of communication issues. In a smart building deployment, where a DT manages energy consumption across diverse systems, the DCC 's interface might display intuitive dashboards illustrating communication statuses, error trends, and recommended actions. This empowers users to make informed decisions swiftly, contributing to effective troubleshooting and performance optimization.

In summary, the Error Handling and Reporting capabilities of the DCC are indispensable components of a resilient and reliable DT ecosystem. By proactively addressing communication challenges, logging detailed events, and offering insightful reporting features, the DCC enhances the overall effectiveness of the DT, making it a robust and dependable tool for various applications.

Ensuring the DCC of a DT adheres to these core identified principles is paramount for establishing a robust and adaptive bridge between the DT and the digital ecosystem. By incorporating extensible protocols, bidirectional communication, and quick or even real-time capabilities, the DCC enables seamless interaction, collaboration, and information exchange within a myriad of application scenarios. Standardized interfaces and dynamic adaptation mechanisms contribute to the layer's versatility, promoting interoperability and simplifying integration with diverse digital entities.

Moreover, the DCC 's commitment to error handling, reporting, and communication awareness fortifies the DT's resilience in the face of challenges. With the ability to detect, respond to, and log communication failures, it ensures stable operations even in dynamic and uncertain environments. The layer's proactive reporting capabilities empower stakeholders to monitor and address potential issues promptly, enhancing the overall reliability of the DT.

Extensibility and reusability principles further future-proof the DCC, allowing for seamless integration of new communication adapters, protocols, and formats. This modular approach minimizes development effort, fosters code reusability, and supports scalability across various DT instances and deployment scenarios.

In essence, the adherence to these principles transforms the DCC into a dynamic, adaptable, and reliable component within the DT architecture. It not only facilitates effective communication between the DT and the digital realm but also empowers the DT to actively participate in the digital ecosystem, contributing to informed decision-making, live or even real-time control, and collaborative endeavors across diverse applications.

5.4 Replication Requirements

Replication is a transformative capability within the realm of DTs that allows for the cloning, transformation, and distribution of physical objects and their digital representations across various environments. To effectively harness replication capabilities, certain requirements should be met to ensure the successful design, implementation, and deployment of this feature within DTs and in particular with both PCC and DCC should be involved with their adapters PCAs and DCAs.

Figure 21 depicts a scenario associated with a one-to-many replication patterns (introduced in previous clause 4.3) where a main DT is replication with multiple synchronized instances. This pattern can be effectively implemented through the use of PCC and DCC and in particular with the flexible approach of PCAs and DCAs. Specifically, DCAs play a crucial role in customizing the digital representation of replicas based on the specific application domains where the twins are deployed. The PCC through its PCAs manages the physical communication between the original PT and the main DT and it is responsible for translating the physical attributes, properties, events, and relationships into digital representations. The PCAs involved adopt specific protocols to ensure accurate and efficient cloning and synchronization between physical and digital counterparts.

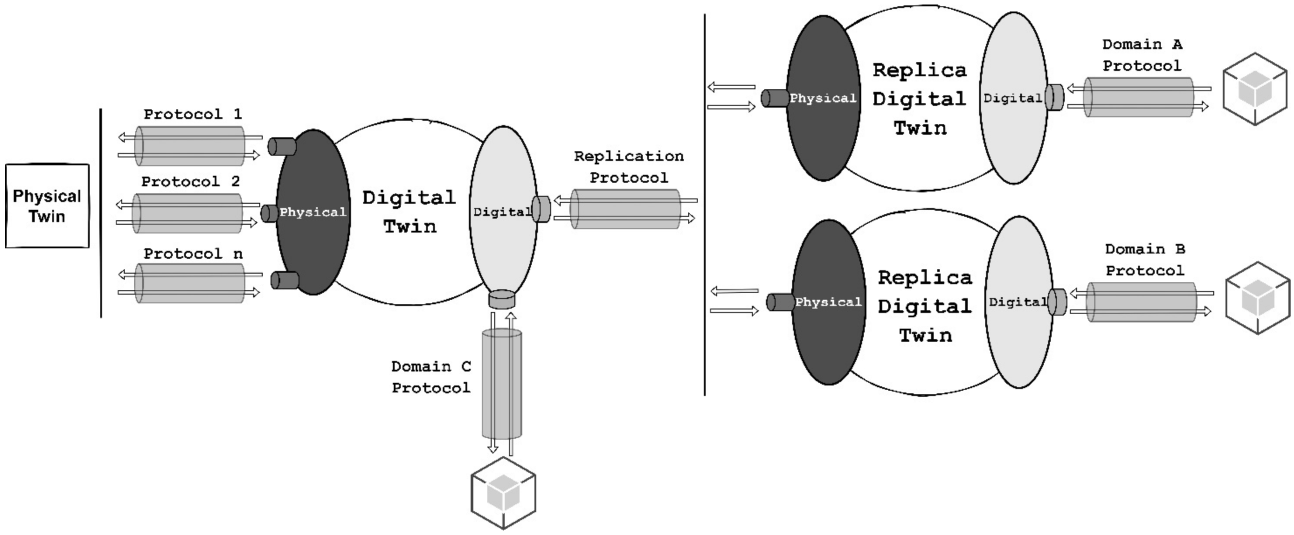

Figure 21: High-level schematic representation of DT's replications through multiple patterns

The DCC and DCA, positioned between the main DT and each replica (with their PCAs), plays a key role supporting the management and synchronization of the replicas for example through the support of a specific and optimized communication protocols dedicated just to replica management. This protocol governs how data is transmitted, synchronized, and updated across all instances. The main DT can also adopt other DCAs to expose its model and capabilities to digital application according to the domain and context where it is deployed (for example on the edge while replicas are in the cloud). Following the same principle, different DCAs can be employed for each replica, each tailored to the specific requirements of the application domain where the twin is deployed. For example, if the DT is used in a smart manufacturing environment, one DCA might focus on live monitoring of machinery, while another might emphasize quality control metrics. These DCAs are responsible for adapting the digital representation to the unique characteristics and needs of each domain, ensuring that the replicas are not generic copies but customized to operate effectively in their respective environments. The modular design of DCAs ensures scalability. New DCAs can be added or existing ones updated independently to accommodate changes in the application domain or to enhance functionality without affecting the entire system.

By implementing the one-to-many replication pattern through PCA and DCA, DTs can effectively extend their presence and adapt to diverse application domains. The customization offered by DCAs ensures that each replica serves its specific purpose within its unique operational context, enhancing the versatility and value of DT in various scenarios. The same benefits can be applied also to a one-to-one synchronization approach where all the replicas are directly connected to the PT. Furthermore, the requirements and aspects that should be taken into account with respect to the replication design, implementation and management are described as follows:

- **Physical Replication:** Cloning of Physical Entities: Replication shall enable the cloning of physical entities or assets, creating identical or modified digital twins of the original physical object. This capability facilitates the creation of multiple digital counterparts within virtualization spaces. The process of replication involves translating physical attributes, properties, events, and relationships into digital representations. This translation should be accurate and efficient, preserving the essential characteristics of the physical object.

- **Digital Replication:** Beyond duplicating physical entities, the DT's digital representation (digital twin) shall also be replicable, leading to hierarchical structures of interconnected digital counterparts. This hierarchical approach allows for increased versatility and adaptability. Replicated digital twins should maintain communication and synchronization capabilities with their physical counterparts, as well as with other digital twins. This interconnectedness is vital for low-latency and live insights and responses.

- **Support for Multiple Replication Schemas:** Replication capabilities shall provide native support for multiple replication schemas, allowing DTs to choose between one-to-one, one-to-many, or other schema variations based on the specific use case and requirements. DTs should have the flexibility to customize replication schemas to align with their intended purpose and interaction patterns. This customization ensures that replication processes are tailored to the unique needs of the digital and physical entities involved. Different replication schemas may have varying resource demands. DTs should optimize resource usage based on the selected schema, minimizing overhead while ensuring effective replication and communication. This flexibility in replication schema support enhances the adaptability of DTs across diverse scenarios and ecosystems.

- **Cross-Domain Compatibility:** Replication capabilities shall enable interoperability across different domains or application scenarios. This means that a DT should be capable of replicating itself and communicating effectively with DTs in other domains, bridging the gap between them. Replication allows for the creation of specialized intermediary DTs that understand the communication patterns and requirements of each domain involved. These intermediary DTs facilitate communication and collaboration between domains.

- **Dynamic Adaptation:** Adaptive Replication: Replication capabilities should be adaptive to different deployment scenarios and dynamic changes in the operational environment. DTs shall be able to decide when and how to replicate themselves based on contextual factors and requirements. Efficient resource management is essential, ensuring that replication processes do not overburden the DT or the underlying infrastructure. This includes managing computational resources, network bandwidth, and storage.

- **Efficient Communication:** Replicated DTs shall communicate efficiently, sharing information and insights seamlessly with other DTs and digital entities. This includes the ability to prioritize and optimize communication flows for different use cases. Replicated DTs shall maintain data consistency, ensuring that all digital counterparts remain synchronized with the physical object and other replicas. Inconsistencies can lead to errors and misinterpretations.

- **Scalability and Resource Management:** Replication capabilities should be scalable to accommodate large-scale DT deployments spanning multiple domains and diverse physical assets. Scalability ensures that DT ecosystems can expand as needed. Efficient utilization of computational and networking resources is crucial to minimize replication-related overhead and support scalability.

- **Bidirectional Communication:** Replication capabilities shall support bidirectional communication between the original DT and its replicated instances. This bidirectionality ensures that data, state changes, and actions can flow seamlessly in both directions, allowing for low-latency and live synchronization and interaction.

- **Multi-Protocol Support:** The ability to support multiple communication protocols, both in the physical and digital communication layers, is crucial for effective replication. Replication shall be protocol-agnostic, allowing DTs to replicate and communicate across diverse ecosystems without being limited by specific protocols. Replication processes should adapt to the communication protocols used by the physical assets and external digital entities. This adaptability ensures that replicated DTs can effectively communicate with their counterparts, regardless of the underlying communication technologies. Multi-protocol support enhances interoperability, enabling DTs to seamlessly replicate and collaborate with entities using different protocols. It also ensures that DTs remain versatile and adaptable in heterogeneous environments. Furthermore, the replication should be supported by stringent data security and privacy standards. This includes secure data transmission and storage, access control mechanisms, and encryption to protect sensitive information.

Meeting these requirements is fundamental to harnessing the power of replication within DTs. Effective replication capabilities enable DTs to adapt, collaborate, and extend their reach across domains and environments, providing valuable insights and solutions in various application scenarios.

5.5 Composition Requirements

In a DT ecosystem, Composition capabilities play a crucial role in creating hierarchical structures, managing DT descriptions, and facilitating communication between DTs. These capabilities extend the scope of DT functionalities and have significant implications for the overall architecture.

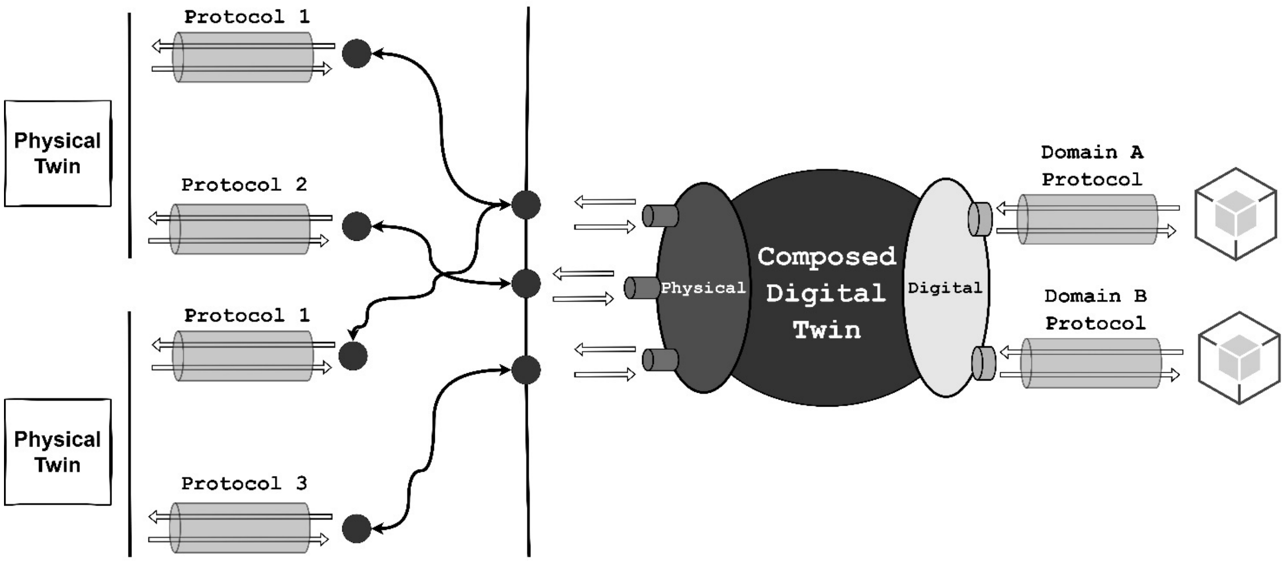

Figure 22: Composed DT communicating directly with the target PTs

Figure 22 depicts a composition configuration where the Composed Digital Twin (CDT) interacts directly with the PTs leveraging both PCC and DCC in conjunction with their respective adapters. PCC, along with its adapter PCAs, plays a pivotal role in providing a structured and standardized description of the associated PTs to the CDT. This description encompasses details about the physical assets' characteristics, properties, events, actions, and relationships. Furthermore, adapters enable the support for a variety of communication protocols to interact effectively with diverse physical assets. For instance, if one PT uses Modbus and another uses MQTT, the PCAs facilitate seamless communication with both by adapting to their respective protocols. PCC facilitates and enables bidirectional communication between the Composed DT and the PTs. The PCAs enable the Composed DT to retrieve fresh data, properties, telemetry, and other information from the PTs. Simultaneously, bidirectional communication allows the CDT to initiate and execute action requests directed towards the PTs.

On the other hand, DCC with its adapter DCAs, abstracts raw data and information from the CDT's core model, providing a simplified representation to external systems through a dedicated DTD. This abstraction shields external systems from unnecessary complexity. The DCC enables efficient communication between the CDT and external systems. The adapters ensure that the digital representation aligns with the communication protocols commonly used in the digital world, enhancing versatility. It also ensures security measures are in place, including authentication, authorization, encryption, and secure key management. This safeguards the communication between the CDT and external systems, protecting against unauthorized access and data breaches.

In the second illustrated configuration (Figure 23), the CDT instead of interacting directly with target PTs is connected to a group of child DTs each responsible for digitalizing specific PTs accordingly to the adapter physical protocols and communication paradigms and data structure.

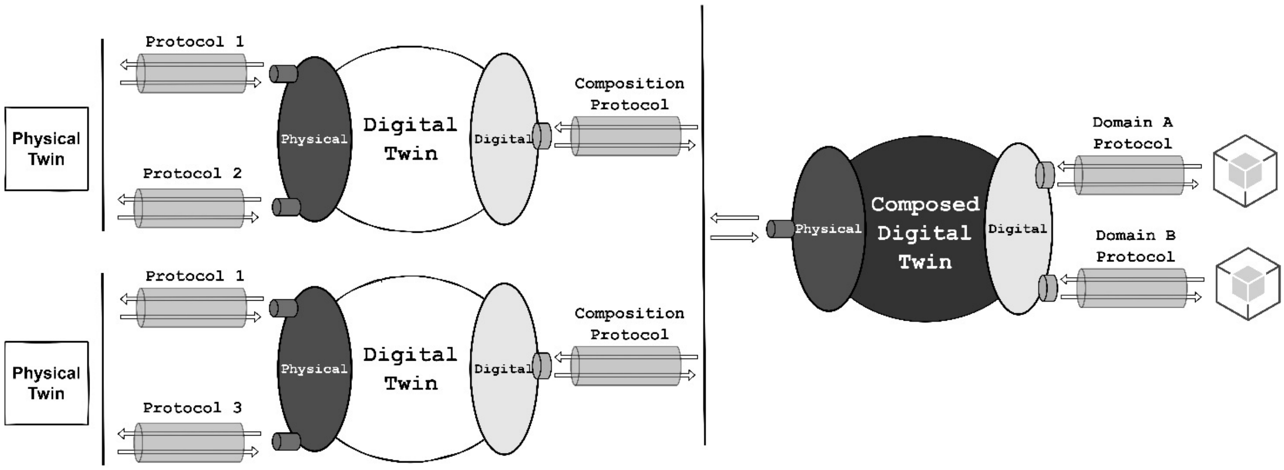

Figure 23: Composed DT aggregating and interacting with other DTs instead of directly talking with PTs

PCC, assisted by PCAs, provides a structured description of PTs to the child DTs. The PCAs ensure that each Child DT receives relevant and specific information about the associated PT for digitalization. DCC, supported by DCAs, abstracts information from each child DT's core model, providing a simplified representation to the CDT. The DCAs ensure that the digital representations from different Child DTs are standardized and can be integrated seamlessly for example using a dedicated composition protocols focused only on that specific task optimizing communication and exchanged data with a specific focus on twins' nature. Furthermore, DCAs offer flexibility for customization and configuration, allowing each child DT to align its communication patterns and protocols with specific use cases and unique requirements of individual PTs. By combining the capabilities of PCC and DCC, along with their respective adapters, in these configurations, the creation of CDTs is facilitated, and the responsibilities are decoupled and distributed among different DTs and their components and adapters. Structured communication and representation ensure effective digitalization of physical assets, fostering adaptability and efficiency in diverse scenarios.

Relying on the analysed scenarios and the importance in the design of both PCC and DCC together with their adapters, in the upcoming paragraphs outline the main requirements that should be taken into account in the design, implementation, and deployment of DT architectures and frameworks in order to support an effective and valuable composition.

Hierarchy Creation & Management: Composition shall allow for the creation of hierarchical structures, enabling a parent DT to oversee and coordinate multiple child DTs. This hierarchy facilitates the organization of complex systems, such as smart cities or industrial processes, where multiple DTs collaborate. The hierarchy should be dynamic, allowing for changes and adaptations as the operational environment evolves. This flexibility ensures that the Composition capabilities can scale and adjust to different scenarios.

Composed Digital Representation Management: Composition shall integrate with the DCC of DTs, ensuring that digital interactions with external entities extend to the entire hierarchy. This integration enables external applications to interact with the hierarchical system as a whole. The Composition capabilities should maintain consistency and synchronization between the digital representations of parent and child DTs. Changes in one DT's state should propagate appropriately to ensure a coherent view of the entire hierarchy.

Composed DT Description & Representation: Composition shall provide a unified description of the parent DT and its child DTs, including their relationships, properties, events, and actions. This unified representation allows external applications to understand the hierarchical structure and capabilities of the entire DT system. The Composition capabilities should support modeling of hierarchical relationships, ensuring that the parent DT accurately reflects the state and behavior of its child DTs. This modeling is essential for maintaining a coherent digital representation.

Hierarchical Communications: Composition shall enable seamless communication and data sharing within the hierarchical structure. This includes the ability for parent DTs to exchange information with their child DTs and vice versa. Communication should support bidirectional data flow, allowing parent DTs to monitor and control the state of child DTs while also enabling child DTs to influence the parent's behavior. This bidirectional communication is vital for coordinated decision-making and action execution.

Composition Bidirectional Communication: Composition shall support bidirectional communication flows between parent and child DTs within the hierarchy. This bidirectional interaction enables not only the monitoring and control of child DTs by the parent but also the ability for child DTs to influence the parent's behavior when necessary. This bidirectional flow of information and commands is essential for effective coordination and decision-making within the hierarchy.

Support for Multiple Protocols & Communication Patterns: Composition capabilities shall be protocol-agnostic, allowing the use of multiple communication protocols and patterns both for physical and digital communications. This flexibility accommodates diverse environments where different protocols may be prevalent. Additionally, it ensures that the Composition layer can seamlessly integrate with various systems, maximizing interoperability across domains. The support for multiple protocols should be modular and reusable to minimize development effort and promote code reusability. This modularity allows for the easy addition of new protocols and communication patterns as needed across different DT instances and deployments, enhancing adaptability and scalability.

Scalability and Adaptability: Composition shall support the scalability of DT hierarchies, allowing for the addition of new child DTs and the adjustment of existing structures to accommodate evolving requirements. The Composition capabilities should be adaptable to different application domains and scenarios, ensuring that hierarchical structures can be tailored to specific use cases.

Cross-Domain Collaboration: Composition shall facilitate collaboration between DTs from different domains or application scenarios. This includes enabling communication and data sharing between hierarchies operating in distinct domains, fostering interoperability in complex ecosystems.

Management and Monitoring: Composition shall provide management interfaces that allow administrators and operators to control and configure the behavior of hierarchical DT structures. This includes defining communication patterns, access controls, and resource allocation within the hierarchy. The Composition capabilities should support monitoring and analytics of hierarchical DTs, providing insights into the performance, behavior, and interactions of the entire system. This data is valuable for optimizing operations and decision-making.

In summary, Composition capabilities in a DT ecosystem are instrumental in structuring hierarchical relationships, managing DT descriptions, and enabling communication within the hierarchy. These capabilities extend the DT's reach, scalability, and adaptability while fostering collaboration and coordination in complex environments. The requirements outlined here serve as a foundation for designing and implementing effective Composition functionalities within DT architectures.

5.6 Cyber-Physical Awareness Requirements

Cyber-physical awareness together with structure entanglement monitoring and support central to the successful design, implementation, and deployment of DTs. This clause outlines the key requirements associated with entanglement support and awareness, as well as the capabilities for both direct and indirect communication in DTs.

This challenging aspect should involve a comprehensive understanding of the role of both PCC and DCC, coupled with their respective adapters, in ensuring the monitoring and extended awareness of the cyber-physical relationship between PTs and DTs. PCC, with its adapters PCAs, plays a central role in providing a structured and standardized description of the associated PTs. This description encompasses intricate details about physical twin, ensuring the entanglement process begins with a comprehensive understanding of the PTs' characteristics, properties, events, and relationships together with a description of connection type, speed estimation and interaction association (uni-directional or bi-directional). This information can be used both by the twin itself to decide how to digitalize the associated PT and also to expose to the digital application the nature of the cyber-physical relationship.

PCC's adaptability to various communication protocols should support and enable entanglement and cyber-physical awareness by accommodating the diverse communication needs of different PTs. Whether employing protocols such as MQTT, CoAP, Modbus, or others, the PCAs facilitate seamless interaction, contributing to the entangled state between the digital and physical counterparts. The bidirectional communication facilitated by PCC and its adapters is crucial for entanglement quality. Live and low-latency data exchange ensures that the digital representation stays synchronized with the constantly evolving state of the PTs, contributing to the high fidelity of the entangled relationship.

On the other hand, DCC, along with its adapters DCAs abstracts and represents the entangled and the established cyber-physical relationship between the PT and the DT in the digital realm. The DCAs ensure that the digital representation aligns with the entangled state of the PTs, allowing for an accurate and synchronized reflection in the digital domain. DCC's efficiency in communication is crucial for maintaining synchronization. The adapters optimize communication flows, prioritizing entangled data exchange based on the live synchronization needs of both the PTs and the digital ecosystem. This efficiency is vital for the timely propagation of changes between the physical and digital layers.

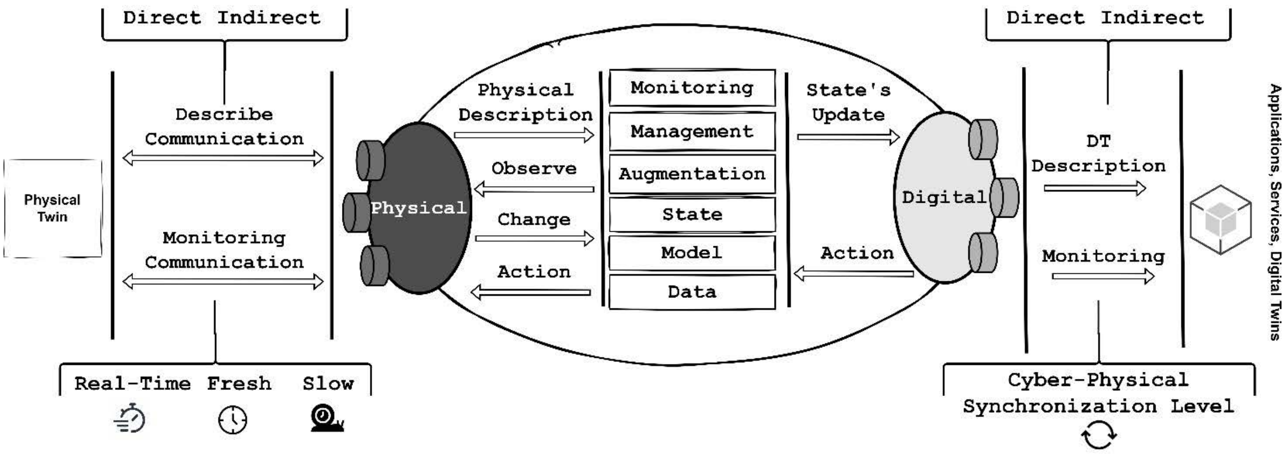

Figure 24: Schematic representation of the DT entanglement across cyber-physical digitalization process

Both PCC and DCC (as depicted in Figure 24), through their respective adapters, should incorporate mechanisms for monitoring of cyber-physical communication and entanglement quality over time during both DT and PT life cycle evolution. Monitoring tools gauge parameters such as latency, data consistency, and synchronization rates, providing insights into the health of the entangled relationship. The entanglement awareness in the digital layer involves understanding the quality and dynamics of the communication, its delay, or the overall status (e.g. to detect disconnection). PCC and DCC should introduce comprehensive approaches for monitoring the health and performance of entanglement, ensuring that the digital layer remains aware of the nuances of the physical entities it reflects.

By emphasizing entanglement quality, monitoring, and synchronization through PCC and DCC, DTs can achieve a harmonious integration of the physical and digital realms. This not only ensures a faithful representation of the physical world but also enables the digital layer to proactively respond to changes in the cyber-physical entanglement, fostering adaptability and efficiency across diverse deployment scenarios.

Relying on this analysis, some of the core requirements related to the mapping, monitoring and representation of the level and quality of the cyber-physical relationship within the digitalization process between PTs and DTs are the following:

- **Cyber-Physical Relationship Modeling:** DTs shall be capable of modeling the ever-changing cyber-physical relationship between the DT and its PT known as entanglement. This modeling should encompass various aspects, including connectivity, speed, and association. The DT should dynamically adapt to these entanglement factors, reflecting how tightly it is connected to its physical counterpart together with to the variation of physical communications over time and how they can impact replication quality and the associated entanglement with respect the target deployment and the associated requirements.

- **Dynamic Awareness:** Entanglement in DT communication is highly dynamic and contingent upon various factors, such as the nature of the PT, available infrastructure, communications, protocols and domain-specific requirements. DTs shall possess dynamic awareness of these entanglement factors and adjust their behavior and communication accordingly. This awareness ensures that the DT remains aligned with specific use case requirements. Furthermore, this awareness shall be presented to the digital space in order to enable external applications and services to be aware of the quality of the DT avoiding malfunctioning or quickly detecting degradation in performance or in the quality of the digitalization process.

- **Fidelity of Digital Representation:** The ability of DTs to faithfully mirror the state of their PT is closely tied to entanglement. In direct and live communication, the DT's representation remains synchronized with the PT, ensuring a reliable reflection of the physical world. In indirect communication, where updates may be delayed, DTs should maintain a reasonable level of fidelity in their digital representations.

- **Adaptability to Connectivity:** The level of connectivity between the DT and its PT can vary significantly. DTs shall be adaptable to scenarios ranging from disconnected DTs with no direct connection to strong, direct connections between the DT and its PT. This adaptability allows DTs to function effectively in diverse environments.

- **Timeliness of Communication:** Promptness in communication is a critical requirement. DTs shall support a spectrum of communication timeliness, from non-real-time scenarios with delayed updates to hard real-time communication requiring instantaneous data transmission. This flexibility ensures that DTs can meet the low-latency demands of different applications.

- **Bi-Directional Association:** The nature of association between the DT and its PT can be uni-directional or bi-directional. DTs shall be capable of both monitoring and controlling the PT in bi-directional associations, allowing for versatile interaction patterns. This flexibility is essential for applications with varying control requirements.

- **Impact on Physical Communications:** Entanglement factors significantly influence the design of the Physical Communication Layer and its internal modules and implemented behaviors. DTs shall be able to implement physical communication that are aware of their performance over time through monitoring metrics in order to support a high level of entanglement awareness in particular in scenarios where the optimization and support for live and fresh data exchange and reliability are required.

- **Application Suitability:** DTs shall enable external digital applications to be aware and evaluate their suitability for specific applications based on the level of entanglement. In direct, live and low-latency communication scenarios, DTs are ideal for control and monitoring systems. In situations with indirect communication and lower promptness, DTs are better suited for historical data analysis or predictive modeling.

- **Impact on DT Implementations:** Recognizing the dynamic nature of entanglement is vital for making informed decisions when deploying DTs. It influences the design of the Physical Communication layer, the suitability of DTs for low-latency applications, and the reliability of their digital representations. Understanding and managing entanglement empowers organizations to align their DT implementations with specific use case requirements.

Entanglement support and awareness, along with capabilities for direct and indirect communication, are fundamental for the effective deployment of twins. DTs should dynamically adapt to changing cyber-physical relationships, ensuring connectivity, promptness, and association align with application requirements. These considerations influence the design of the PCC, the suitability of DTs for different applications, and the reliability of their digital representations, ultimately leading to successful DT implementations.

Furthermore, in the context of analysing the requirements and the characteristics associated with the possibility to have direct and indirect DTs communication patterns the following structural points and requirements can be highlighted:

- **Direct Communication Requirements:**
  - **Live & Quick Interaction:** Direct communication in DTs demands quick synchronization and interaction capabilities. DTs shall support instantaneous data exchange with their Physical Twins (PTs) to facilitate immediate control and monitoring, making them suitable for applications requiring rapid responses and decision-making.

- **Low Latency:** Low latency is critical for direct communication scenarios. DTs should minimize communication delays between themselves and their PTs, ensuring that updates and actions are processed without perceptible delays. This requirement is essential for also to envision real-time control and monitoring applications.
- **High Bandwidth:** Direct communication may involve high data transfer rates, especially in applications where extensive sensory data is generated. DTs should accommodate high bandwidth requirements, allowing them to handle large volumes of data seamlessly and without bottlenecks.
- **Reliability:** Reliability is paramount in direct communication. DTs should ensure that data transmitted between the digital and physical realms is highly dependable. This reliability is essential for maintaining the trustworthiness of the digital representation and supporting critical decision-making processes.
- **Security:** Security is a top priority in direct communication scenarios, particularly when control commands are involved. DTs shall implement robust security measures to protect against unauthorized access, data breaches, and cyber-physical attacks. Security protocols, encryption, and access controls are vital components of direct communication requirements.
- **Scalability:** DTs should be scalable to accommodate direct communication with multiple PTs simultaneously. This scalability ensures that DTs can effectively manage and control various physical entities within their operational domain, making them suitable for large-scale applications.

- **Indirect Communication Requirements:**
  - **Data Buffering and Storage:** Indirect communication often involves delayed data transmission. DTs shall be equipped with data buffering and storage capabilities to capture and store data when communication with PTs is temporarily interrupted. This ensures that no information is lost during intermittent connections.
  - **Data Synchronization:** Data synchronization is crucial in indirect communication scenarios. DTs shall synchronize their digital representation with the PT once communication is reestablished, ensuring that the digital model accurately reflects the physical state. This synchronization maintains the integrity of historical data and the reliability of the DT.
  - **Graceful Degradation:** DTs should gracefully degrade their functionality during indirect communication. When live and low-latency communication is not possible, DTs shall transition into a mode that focuses on data storage, analysis, and preparation for future actions. This graceful degradation ensures that DTs remain valuable even in challenging communication conditions.

Direct and indirect communication requirements play a significant role in shaping the design and capabilities of Digital Twins (DTs). Direct communication demands quick and live interaction, low latency, high bandwidth, reliability, security, and scalability, making DTs suitable for applications that require immediate control and monitoring. In contrast, indirect communication necessitates data buffering, synchronization, predictive analytics, efficient resource utilization, and graceful degradation, enabling DTs to operate effectively in scenarios with intermittent connectivity. Balancing these requirements allows DTs to excel in diverse application domains, from live and synchronized low-latency control systems to historical data analysis and predictive modeling.

5.7 Cross-Domain Communication Requirements

Cross-domain DTs bridge the boundaries between different application domains and enable interoperability and collaboration across diverse contexts. In this context, the DCC and its associated DCAs play a critical role in facilitating seamless communication, data exchange, and interaction among DTs operating in disparate domains. DCC, modularity allows DTs to communicate across diverse ecosystems, each potentially utilizing different communication protocols. For instance, a manufacturing DT employing Modbus can seamlessly communicate through different scenarios, ensuring interoperability across varied application domains. DCAs within the DCC ensure that communication adheres to standardized interfaces, shielding the core DT from intricacies associated with diverse protocols. This standardization promotes a consistent method of interaction, simplifying the process of integrating DTs from different domains. Nevertheless, DCAs abstract the raw data and information from the core DT, providing a simplified and standardized representation for external DTs in different domains. This abstraction shields external systems from unnecessary complexity, promoting efficient and straightforward interactions.

DCC, through the utilization of DCAs, supports hierarchical structures of interconnected digital counterparts. For example, an energy management DT in the industrial domain can be hierarchically linked to a smart city DT, enabling collaborative decision-making and resource optimization across domains. As previously introduced, DCAs offer the flexibility for customization and configuration of communication patterns, allowing DTs to align with the specific use cases and unique requirements of individual domains. For instance, a transportation DT might customize its communication patterns when interacting with a smart infrastructure DT. Since the DCC should be adaptive, capable of reconfiguring communication parameters and protocols based on the specific requirements of each domain. DCAs enable DTs to dynamically adapt their communication strategies, ensuring seamless integration and interoperability in evolving operational scenarios.

Consider a scenario where an agriculture DT, responsible for monitoring crop health and irrigation in the agricultural domain, needs to collaborate with a weather forecasting DT in the meteorological domain. The DCC, with its DCAs, facilitates the seamless integration of these DTs. The agriculture DT's DCA translates specific crop data into a standardized format, allowing the weather forecasting DT to understand and incorporate this information for more accurate predictions. The hierarchical structure established through DCC enables live collaboration, enhancing decision-making processes in both domains.

In essence, the DCC, with its associated DCAs, as schematically illustrated in Figure 25 acts as a unifying force in the cross-domain integration of DTs. By providing adaptable, standardized, and hierarchical communication, DCC ensures that DTs from different application domains can collaborate efficiently, creating a synergistic digital ecosystem that transcends boundaries and addresses complex challenges through collective intelligence.

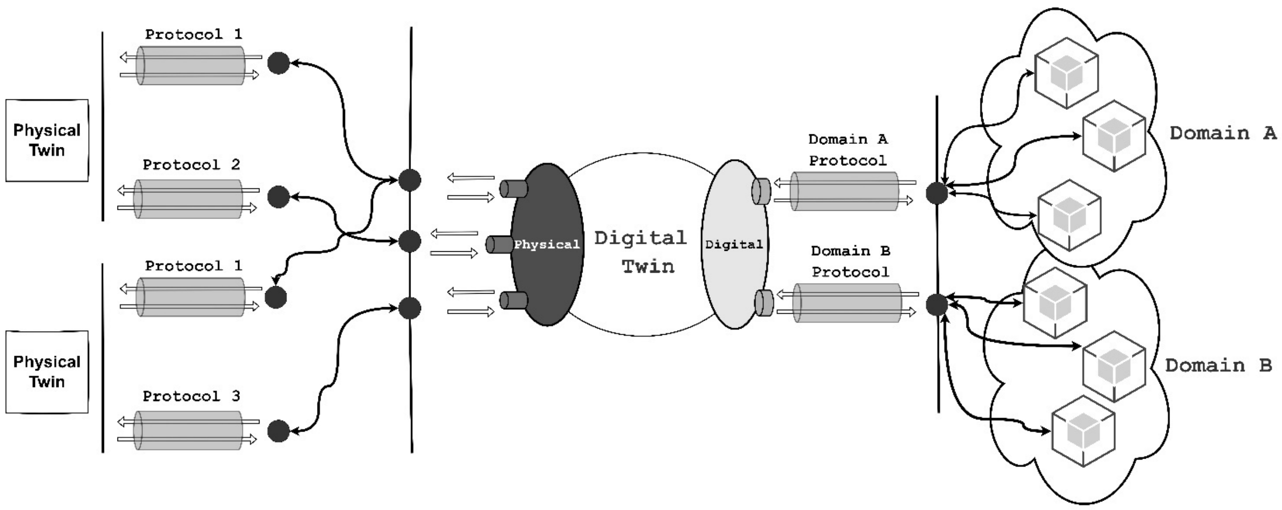

Figure 25: Representation of Cross-Domain DTs scenario exploiting replication, composition and entanglement awareness

- To effectively support these capabilities, Cross-Domain DTs should meet specific requirements:
- **Replication Compatibility:** Cross-Domain DTs shall be compatible with various replication strategies, such as one-to-one and one-to-many, to facilitate the creation of digital replicas that can operate seamlessly across different domains. This compatibility allows DTs to replicate themselves and communicate with counterparts in other domains, bridging the gap between them.
- **Composition Flexibility:** Cross-Domain DTs shall offer flexible composition capabilities to create hierarchical structures where a parent DT oversees the coordination of child DTs from different domains. This hierarchical structure streamlines cross-domain communication, allowing the parent DT to act as a mediator and translator between the child DTs and external entities. The flexibility of composition enables DTs to adapt to different collaboration scenarios and effectively manage diverse entities.
- **Entanglement Awareness:** Cross-Domain DTs require a deep understanding of entanglement with physical counterparts in various domains. They shall be aware of the dynamic and ever-changing nature of the relationship between digital and physical entities, considering factors like connectivity, promptness, and association. This awareness allows Cross-Domain DTs to adapt to the unique entanglement characteristics of each domain, ensuring effective communication and synchronization.

- **Interoperability:** Interoperability is a fundamental requirement for Cross-Domain DTs. They should support communication protocols, data formats, and interaction patterns from multiple domains, allowing them to seamlessly integrate with different ecosystems. This interoperability extends to both physical and digital communication, ensuring that Cross-Domain DTs can collaborate with entities from various domains.

- **Security & Access Control:** Security is paramount in cross-domain scenarios. Cross-Domain DTs shall implement robust security measures to protect data and interactions when bridging different domains. Access control mechanisms shall be in place to ensure that only authorized entities can interact with the DTs, preventing unauthorized access or malicious actions.

- **Data Transformation & Normalization:** Cross-Domain DTs shall be capable of transforming and normalizing data from various domains. This capability ensures that data from different sources can be standardized and understood within the context of the DT. It allows for consistent data representation and meaningful insights across domains.

- **Scalability:** Cross-Domain DTs should be scalable to accommodate the complexities of multiple domains. They should handle interactions with numerous digital and physical entities, supporting large-scale applications that span various domains. Scalability ensures that Cross-Domain DTs can meet the demands of diverse use cases.

- **Management & Monitoring:** Cross-Domain DTs require robust management and monitoring capabilities. Administrators should be able to oversee and control interactions across domains, ensuring that the DTs operate effectively. Monitoring tools should provide insights into the behavior and performance of Cross-Domain DTs in different contexts, supporting manageability and accountability.

- **Adaptability & Customization:** Cross-Domain DTs shall be adaptable and customizable to meet the specific requirements of each domain they interact with. This adaptability includes the ability to adjust communication protocols, data representations, and interaction patterns according to the unique characteristics of each domain. Customization ensures that Cross-Domain DTs can seamlessly fit into different ecosystems.

Cross-Domain DTs play a pivotal role in enabling interoperability, collaboration, and communication between diverse application domains. To fulfill their potential, these DTs should adhere to a set of requirements that encompass replication compatibility, composition flexibility, entanglement awareness, interoperability, security, data transformation, scalability, management, monitoring, adaptability, and customization. By meeting these requirements, Cross-Domain DTs can operate effectively in a wide range of contexts, fostering innovation and cooperation between different domains.

5.8 Edge-to-Cloud Communication Requirements

Edge and Cloud DTs serve as key components of the digital ecosystem, offering distinct advantages and capabilities. For a seamless integration and effective management, synchronization, and communication between Edge and Cloud DTs, careful consideration should be applied with respect to both PCC and DCC together with their adapters PCAs and DCAs as schematically illustrated in Figure 26.

The DCC, along with DCAs, should support scalable communication that caters to the dynamic nature of Cloud DTs. Cloud DTs often handle vast amounts of data and require efficient communication protocols. DCAs enable the translation of data from Edge to Cloud-compatible formats, ensuring seamless communication between the diverse environments. On the other hand, on the Edge side, the PCC, facilitated by PCAs, should support adaptive communication. Edge DTs operate in resource-constrained environments, and PCAs play a pivotal role in optimizing communication for these constraints. For instance, using lightweight protocols like MQTT for communication in bandwidth-limited Edge scenarios.

The DCC should abstract raw data from Edge DTs, providing a hierarchical representation that aligns with the Cloud's processing capabilities. DCAs play a crucial role in transforming and packaging Edge-generated data into standardized formats for effective processing and analysis in the Cloud. On the Edge, the PCC with PCAs should ensure that data representations are efficient and tailored to the specific requirements of Edge applications. PCAs can perform localized processing and abstraction before transmitting relevant information to the Cloud and also hierarchical management and organization of twin can be applied on the edge by exploiting adapters configurations and adoption.

The DCC and DCAs facilitate bidirectional communication, enabling Edge DTs to transmit low-latency data, events, and updates to the Cloud.

EXAMPLE: A smart city Edge DT monitoring traffic patterns can continuously synchronize its data with a Cloud-based urban planning DT.

Conversely, the DCC ensures that updates and insights from Cloud analytics reach the Edge DTs efficiently. DCAs play a role in translating high-level insights into actionable data for Edge devices. This bidirectional synchronization enhances the Edge and Cloud DTs' collective intelligence.

With respect to security, the DCC should support secure communication protocols for transmitting sensitive data to the Cloud. DCAs implement security measures to encrypt and authenticate data, ensuring that Edge-generated information is protected during transmission and storage in the Cloud. On the Edge, the PCC and PCAs should implement security measures suitable for resource-constrained environments. PCAs ensure that data transmitted to and from the Edge is secure, establishing a robust and trusted communication channel.

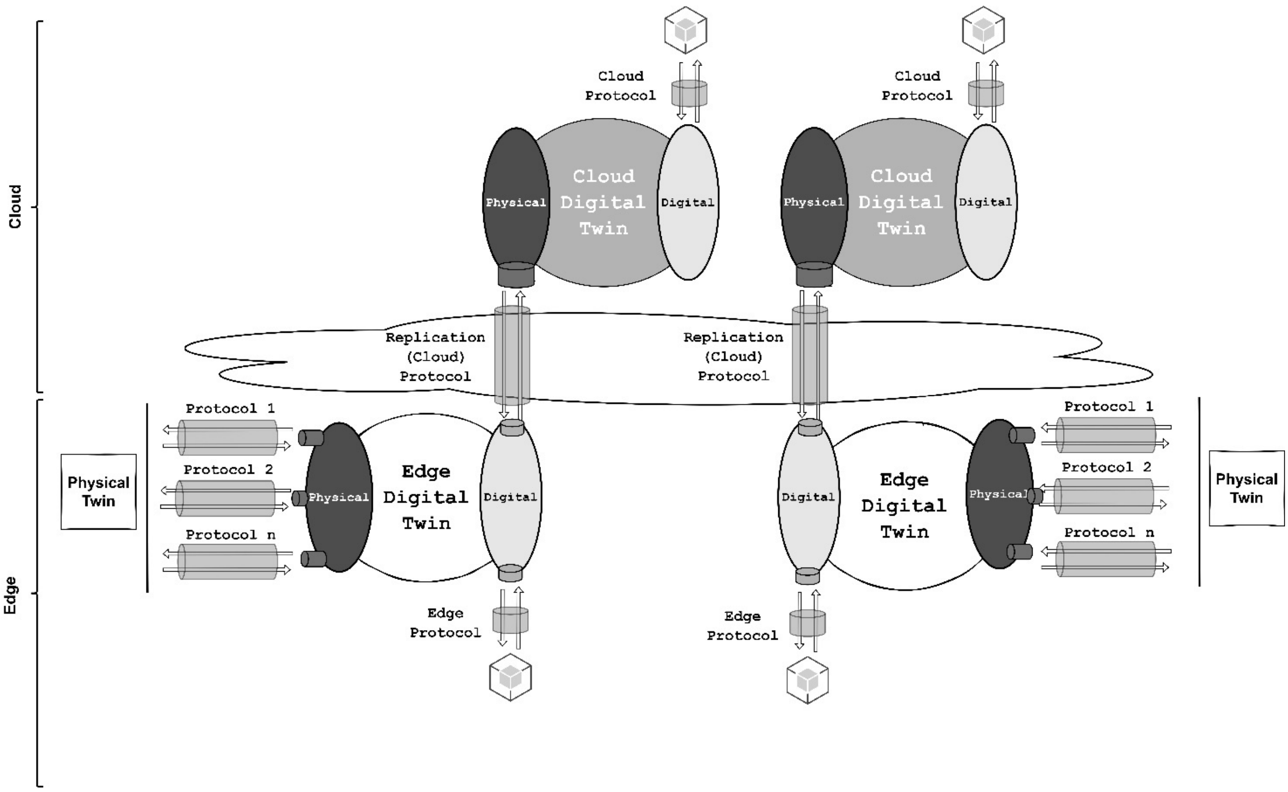

Figure 26: High-level representation of edge-to-cloud DT continuum and deployment opportunities in terms both of replication and composition

PCAs enable Edge DTs to initiate communication with the Cloud when needed, allowing for adaptive deployment models. For instance, an Edge DT monitoring manufacturing equipment can trigger Cloud analytics for predictive maintenance based on live observations. Conversely, DCAs support Cloud-initiated communication to Edge DTs, allowing for updates, patches, or reconfiguration. This bidirectional adaptability ensures that both Edge and Cloud DTs can respond to changing requirements and scenarios effectively.

Consider an IIoT scenario where Edge DTs monitor machinery health on a factory floor. The PCC and PCAs handle low-latency or even real-time communication, ensuring rapid response to equipment anomalies. Simultaneously, the DCC and DCAs enable the Edge DTs to synchronize their data with a Cloud-based DT responsible for predictive maintenance, leveraging historical data and advanced analytics to optimize equipment performance. Summarizing, the effective management and synchronization of Edge and Cloud DTs rely on the harmonious integration of DCC, PCC, and their associated adapters. This approach ensures that data flows seamlessly between Edge and Cloud environments, optimizing the collective intelligence and capabilities of the digital ecosystem.

To effectively design, implement, and deploy Edge and Cloud DTs, specific requirements should be addressed:

- **Replication & Distribution:** DTs shall support replication strategies that enable them to create digital replicas and distribute them across edge nodes allowing replicas to effectively communicate without limitations associated to their architectural deployment on the edge or in the cloud. This capability for example is crucial for maintaining low latency and supporting real-time applications at the edge and on the other hand to deploy cloud replicas to efficiently replicate and distribute DT instances across cloud infrastructure, ensuring scalability and redundancy. Edge and Cloud DTs should seamlessly operate within the edge-to-cloud continuum. They should support dynamic migration of DT instances between edge nodes and the cloud, ensuring flexibility and adaptability across the continuum.

- **Customization & Configuration:** Edge and Cloud DTs shall allow for customization and configuration to meet the specific requirements of various edge environments and cloud infrastructures. This flexibility ensures that DTs can adapt to diverse ecosystems and use cases.

- **Physical & Digital Communication:** Both Edge and Cloud DTs shall address unique communication requirements. Edge DTs should excel in handling low-latency, high-throughput communication with edge devices and sensors. They should support edge-specific protocols and formats to optimize data exchange. Cloud DTs, on the other hand, should focus on efficient communication with diverse cloud services and data centers, considering scalability and reliability in a cloud-based environment. Both Edge and Cloud DTs should ensure that their communication capabilities align with the specific demands of their deployment environments. Furthermore, Edge DTs should enforce strong security measures at the edge to protect against local threats and ensure data privacy. Cloud DTs should implement robust security mechanisms to safeguard data and interactions within the cloud environment. Both Edge and Cloud DTs should support access control and encryption.

- **Composition & Orchestration:** Both Edge and Cloud DTs shall provide robust support for composition and orchestration management. Edge DTs should support hierarchical compositions that allow them to coordinate with other Edge DTs and Cloud DTs, optimizing resource utilization and decision-making. Cloud DTs should offer powerful orchestration tools to manage Edge DTs and coordinate their actions within the cloud infrastructure.

- **Entanglement Management:** Edge DTs shall be well-equipped to manage entanglement with physical entities at the edge. This entails a deep understanding of connectivity, promptness, and association with edge devices and sensors. Cloud DTs, on the other hand, should focus on managing entanglement with diverse Edge DTs and ensuring seamless communication across the cloud environment. Edge DTs should prioritize low latency to support live, real-time and near-real-time applications. They should be designed to minimize communication delays between edge devices and DT instances. In contrast, Cloud DTs should be able to handle or more in general to be aware of higher latency inherent in cloud-based communication, ensuring efficient operation even in scenarios with delayed responses and providing to digital application and service an updated and dynamic description of the status of the DT and its relationships with its replicas on the edge or the relationship with the physical counterpart.

- **Interoperability:** Interoperability is a fundamental requirement for both Edge and Cloud DTs. They shall support a wide range of communication protocols, data formats, and interaction patterns to facilitate seamless integration with edge devices, sensors, and cloud services. This interoperability extends to both physical and digital communication following principles and requirements introduced and illustrated in previous clauses.

- **Scalability:** Scalability is essential for both Edge and Cloud DTs. Edge DTs should scale horizontally by adding more edge nodes to accommodate growing workloads. Cloud DTs should provide vertical scalability to handle increased demands within cloud infrastructure. Scalability ensures that DTs can meet the requirements of diverse use cases.

- **Data Processing & Analytics:** Edge DTs shall possess data processing and analytics capabilities to extract meaningful insights from local data sources. They should support edge analytics to enable low-latency decision-making. Cloud DTs, on the other hand, shall offer extensive data processing and storage resources to perform advanced analytics and support data-intensive applications.

- **Management & Monitoring:** Robust management and monitoring tools are essential for both Edge and Cloud DTs. Administrators should be able to oversee and control DT instances, monitor their performance, and ensure efficient resource utilization. Edge DTs should offer edge-specific management tools, while Cloud DTs should provide cloud-centric management capabilities.

Edge and Cloud DTs play critical roles in the digital landscape, offering unique capabilities to support edge computing, cloud services, and the continuum between them. To fulfill their potential, these DTs should adhere to a set of requirements that encompass replication and distribution, composition and orchestration, entanglement management, interoperability, latency control, scalability, security, data processing, management and monitoring, edge-to-cloud continuum support, and customization. By meeting these requirements, Edge and Cloud DTs can effectively serve diverse use cases and contribute to the advancement of digital ecosystems.

6 Digital Twin Description & Interoperability

6.0 Introduction

As previously anticipated and introduced in previous clauses 4 and 5 and taking into account the ever-evolving and challenging landscape of DTs, a fundamental aspect that should be taken into account in the design and implementation of DT oriented architectures and framework is the ability to precisely define, describe, and discover twins within one or multiple deployments. To harness the full potential of DTs, it is essential to have clear guidelines and robust mechanisms in place for Digital Twin Descriptions (DTDs) and their discoverability across multiple cross-domain use cases [i.11].

This clause analyses the requirements and design principles associated to the definition and adoption of a description for DTs and the essential synergy between this description capability and its discoverability through DTs deployments where multiple twin instances can be discovered in an interoperable way by digital applications and services. These facets collectively form the backbone of how DTs are defined, communicated, and made accessible in the digital realm.

The first step to explore is related to the analysis of the specific prerequisites and criteria that govern the formulation of DTDs. These requirements encompass various aspects that should be present in DT instances and infrastructures, from identity and metadata to state descriptions, interfaces, security measures, and more. A comprehensive understanding of these prerequisites is vital for constructing effective and standardized DTDs.

Moving forward, this clause delves into the intricate process of discovering DTDs. Effective discoverability is paramount for leveraging DTs efficiently across diverse industries and domains and examines the main requirements in terms of methodologies, protocols, and techniques that can guide the design and implementation of DT's architectures to facilitate the seamless exploration and retrieval of DTDs, promoting interoperability and integration.

6.1 Digital Twin Description Requirements

A DTD is a vital component in the world of DTs, serving as the blueprint structure that defines and represents the essence of a twin. It shall encapsulate various characteristics and attributes that are instrumental in comprehensively describing a DT, enabling seamless interaction, monitoring, and interoperability across diverse application domains.

The DTD's primary role shall be to encapsulate the essential attributes and characteristics of a DT, effectively serving as a digital representation of the physical entity. Through a DTD, stakeholders can model a DT's state by leveraging its various components:

- **Global Unique Identifier:** Each DT shall be globally uniquely identified and its identifier should be reported in the DTD in order to provide an immediate reference to the target twin.
- **PT References:** DTDs shall include identifiers of the associated PTs, type, description, and the potential associated metadata. This context aids in deployment, discovery, and understanding the DT's capabilities and relationships and in particular the digitalized physical assets that it is managing. In this case PTs' identifiers can be associated to the specific custom domain associated with the physical counterpart and domain-driven information needed to understand and identify the PT can be integrated in the metadata.
- **Properties:** These represent readable and observable attributes of the DT and can include variables that change dynamically with the PT's evolution. Examples of properties include temperature readings from a sensor or the on/off state of a switch.

- **Relationships:** Relationships denote connections between the DT and other PTs or DTs. They can be observable, dynamically created, and subject to change over time. Relationships may convey different semantic meanings, such as containment (e.g. a floor containing rooms), cooling (e.g. an HVAC system cooling a room), or billing (e.g. associating a compressor with a specific user).

- **Events:** Events represent significant occurrences at the PT or domain level. These events are observable and can encompass a wide range of scenarios, from detecting anomalies to capturing critical domain-specific incidents.

- **Actions:** Actions refer to the capabilities exposed by the PT, which can be performed by or on the device to alter its status or interact with the environment. For instance, a DT representing a light fixture may include a toggle action to turn the light on or off.

- **State:** The DTD defines and expose not only the description of the available characteristics and capabilities of the twin (in terms of properties, events, relationships, and actions) but also the current context or conditions of the target twin, providing access to read the values of Properties, Relationships, Events, and Actions. This functionality enables monitoring and control of the DT's behavior. The DTD should include the characteristics and the specification related on how to interact with DT's state and capabilities according to the current configuration of the DCC and the associated DCAs. Multiple serialization and representation of the DTD together with different communication technologies and protocols can be adopter through different DCAs to interact with the DT's state and potentially also directly with a single capability such as property, events, relationships, or actions.

- **Monitoring:** DTDs shall introduce information about how an external application can monitor the DT discovering and observing available and exposed metrics and events generated by the DT, enhancing overall accountability and the awareness on both the cyber-physical relationship between DT and PT and also their evolution during their lifecycle. Digital applications can proactively detect and handle specific occurrences, anomalies, or changes in the DT's environment or internal state.

- **Management:** DTDs shall include management-related specifications, allowing external application to discover the capabilities and functionalities of the DT's software instance that can be managed and how it can be done. Management functionalities can change with respect to the nature of the DT and the adopted implementation and might include for example lifecycle control (start, stop, restart), versioning, and the dynamic configurations of both parameters and potentially also of active PCAs and DCAs. This addition ensures that stakeholders can oversee and control the DT's evolution and adaptability over time.

The DTD's ability to capture, describe and represent these attributes and capabilities of an active DT allows for a faithful digital representation of the DT's state together with its current context. It enables stakeholders to monitor, control, and interact with the DT in ways that mirror its physical counterpart, fostering seamless integration and interoperability across various application domains and use cases.

To ensure the effectiveness and interoperability of DTs, a set of fundamental requirements should be met in the creation, deployment, and utilization of DTDs:

- **Identity and Metadata:** At its core, a DTD provides information about the global unique identity of the associated DT. This identity serves as a digital fingerprint, allowing the DT to be addressable within the software space. In addition to identity, DTDs include metadata, offering valuable insights such as the DT's type, description, and annotations. Metadata enriches the context surrounding the DT, making it more understandable and discoverable.

- **State Description:** Perhaps the most crucial aspect of a DTD is its ability to describe the state of the DT. This encompasses a comprehensive representation of the DT's properties, relationships, events, and behaviors/actions. Properties are observable attributes that can change dynamically, events capture relevant occurrences, relationships define connections to other DTs, and behaviors/actions encompass the actions and capabilities that the DT can perform. The state description allows stakeholders to monitor and control the DT's behavior and functionality effectively.

- **Physical & Digital Description:** DTDs provide a detailed account of the available interfaces for interacting with the DT, both in the physical and digital realms. For physical interfaces, this means describing how the DT integrates with its digitalized PTs. In the digital domain, the DTD specifies supported interaction protocols, data formats, and operations. These interfaces are critical for seamless integration, ensuring that the DT can communicate and interact with various platforms and digital services.

- **DT's Description Communication Adoption:** DTD serves as a common reference for both the Digital Communication layer and the Physical Communication layer within a DT instance. While the DTD primarily aids the Digital Communication layer in exposing the DT's structure to external digital services and applications, it should also support scenarios involving DT replication and composition. In these cases, the Physical Communication layer can leverage the DTD to understand the essential characteristics of a DT that need to be replicated or composed. This dual usage ensures consistency and coherence in DT interactions across various scenarios and communication layers, enhancing the overall interoperability and versatility of the DT ecosystem.

- **Monitoring:** To enhance the DT's accountability and support proactive decision-making, DTDs can include information about monitoring metrics. These metrics capture key aspects of the DT's lifecycle and any errors or anomalies that may occur. By incorporating monitoring data into the DTD, digital applications and services gain the capability to detect and respond to specific events, changes, or irregularities in the DT's environment or internal state.

- **Security & Privacy:** Security and privacy are paramount in DT interactions. DTDs offer a means to specify security measures associated with communication between the DT, PTs, and external digital applications. These measures include authentication, access control, encryption, and data protection mechanisms. By outlining security and privacy requirements within the DTD, it ensures that interactions with the DT remain secure and confidential.

- **Semantics and Contextual Information:** To facilitate meaningful integration and reasoning about the DT, DTDs can incorporate semantic annotations and contextual information. These additions provide deeper insights into the DT's capabilities and interactions. Semantic annotations enhance interoperability by aligning semantics, enabling better understanding, interpretation, and meaningful utilization of the DT.

- **Interoperability:** Ensure that the DTD can be mapped or translated to other established standards and description formats, such as oneM2M [i.13] or Web of Things (WoT) [i.8], to promote cross-ecosystem compatibility and enable the seamless integration of DTs into broader IoT and cyber-physical systems. This capability allows for greater flexibility and collaboration across different IoT ecosystems and ensures that DTDs remain adaptable to evolving industry standards and best practices.

These requirements collectively form a robust foundation for creating DTDs that enable seamless discovery, integration, and interoperability within the DT ecosystem. By adhering to these prerequisites, stakeholders across diverse domains and industries can effectively harness the power of DTs, fostering innovation and knowledge sharing while promoting standardized practices for digital representations of physical entities.

6.2 Digital Twin Description Discoverability & Requirements

In the context of DTs, effective discoverability of DTDs is crucial for enabling seamless interactions, promoting interoperability, and supporting the deployment of a wide range of applications and digital services across diverse domains. Discoverability, in this context, involves the ability for applications and potentially other twins to identify and access the DTDs associated with various DT instances without prior knowledge, allowing applications and digital services to explore and interact with them efficiently.

Discoverability in the realm of DTs opens numerous possibilities and benefits and should be one of the foundational capabilities of DT architectures and frameworks in order to enable the effective creation of a DT ecosystem. It allows applications and digital services to autonomously find and engage with DTs, even in large-scale and dynamically changing environments.

Figure 27 delineates two primary approaches to DTD discovery, each tailored to specific deployment scenarios and operational requirements. In the first configuration, DTDs are discovered within a local network where PTs, DTs, digital applications, or other twins are active and interconnected. The responsibility for implementing local discovery protocols, such as ZeroConf, mDNS, or UPnP, can be incorporated into an active DCAs or a dedicated adapter solely focused on supporting discoverability. Consider for example a smart home environment where various devices, twins and applications are interconnected. A dedicated adapter within this environment employs ZeroConf to allow an external service to discover the presence of a specific DT. Once discovered, the DTD associated with the DT is accessed through a well-known endpoint, leveraging an HTTP RESTful approach.

In the second configuration, DTDs are centralized, indexed, stored, and managed within a directory entity. This directory entity serves the dual purpose of enabling DTs to create, update, and delete their DTDs over time while also providing external applications with the ability to search and access available DTDs. Imagine for example a smart city scenario where numerous DTs are deployed across various sectors. A centralized directory entity manages and indexes all DTDs associated with the deployed DTs. An external application searching for specific DT information can directly access the directory, discover relevant twins, and subsequently communicate with them using the information provided in the DTD descriptor and associated Digital Communication Adapters (DCAs).

Advanced directory configurations interaction patterns may include for example digital applications that can register DTDs as intermediate twins, enabling indirect access to twins that may not have direct visibility of the directory. This facilitates a hierarchical structure where certain twins act as intermediaries for others or digital application in charge of managing multiple twins through different deployments. Furthermore, multiple instances of DTD directories may be interconnected to support synchronized and distributed management. This can be achieved through deployment in various locations, edge and cloud environments, ensuring redundancy, scalability, and efficient DTD management across diverse scenarios.

EXAMPLE: In a manufacturing setting with edge computing capabilities, DTD directory instances are deployed across different factory locations.

Each instance manages DTDs locally, and they are interconnected to ensure a synchronized and distributed approach. This enhances fault tolerance and enables efficient DTD management in a distributed environment.

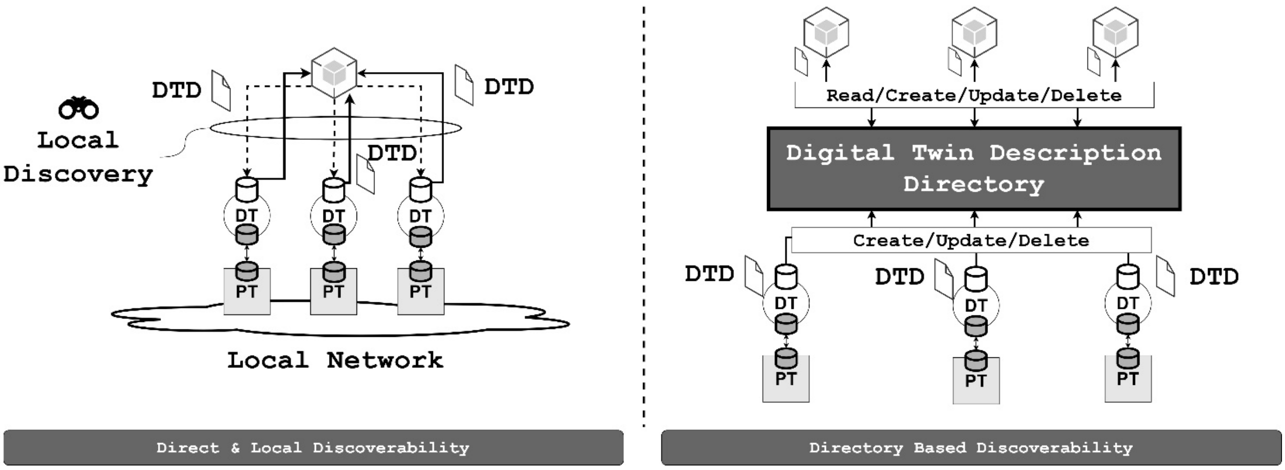

Figure 27: Schematic representation of DT Description Discovery approaches taking into account both local and directory-base solutions

This capability is particularly valuable in scenarios where a priori knowledge about existing DTs is limited or non-existent. Scenarios, requirements and aspects that might be taken into account in the design and development of DT architecture and framework are the following:

- **Dynamic Cyber-Physical Environments:** In challenging cyber-physical environment such as IoT and IIoT deployments, where sensors, devices, and assets are continuously added or removed, discoverability ensures that new DTs can be seamlessly integrated into existing systems, enhancing low-latency decision-making and automation in procedures such as also data collection and the application of dynamic interaction policies without being affected by the fragmentation of the physical layer.

- **Cross-Domain Integration:** When deploying DTs across different domains and industries, discoverability enables applications to bridge the gap between disparate DT ecosystems, fostering collaboration and information exchange. For instance, in a smart city context, DTs related to transportation, energy management, and public services can be discovered and integrated to optimize urban operations.

- **Scalable Digital Services:** For digital services that rely on DTs, discoverability simplifies the onboarding process. Services can automatically find and access relevant DTs, reducing manual configuration efforts and accelerating deployment.

- Achieving an effective discoverability of DTDs relies on a combination of distributed and centralized techniques. These techniques enable both Service Discovery, which involves finding the existence of a DT, and Resource or Capability Discovery, which allows for the exploration of a DT's characteristics and properties before initiating interactions.

- **Service Discovery:** This initial phase involves identifying whether a DT exists within the network or environment. It focuses on the discovery of DT instances and their availability for interaction. Techniques such as service broadcasting, directory services, or decentralized discovery protocols can be employed to determine the presence of DTs.

- **Resource or Capability Discovery:** Once a DT is discovered, applications and digital services often need to understand its capabilities, interfaces, and data structures before meaningful interactions can take place. Resource or Capability Discovery mechanisms enable the exploration of a DT's DTD, providing insights into its properties, relationships, events, behaviors, actions, interfaces, and communication protocols. These mechanisms facilitate informed decisions about how to engage with the DT effectively and can rely on a structured and interoperable design of the Digital Communication layer of a twin allowing a seamless interaction to retrieve the DTD and interact with the DT's instance.

- **Enabling Seamless Exploration:** Seamless discovery and exploration of DTDs empower applications and digital services to autonomously adapt to changing environments and evolving DT ecosystems. It reduces the barriers to entry for developers and service providers, fostering innovation and collaboration across industries. By abstracting the complexity of interacting with DTs and promoting interoperability, discoverability plays a pivotal role in harnessing the full potential of DT technologies.

In the world of DTs, where the integration of physical and digital components is paramount, discoverability of DTDs emerges as a fundamental enabler. It empowers applications and digital services to explore, find, and interact with DTs in a seamless and autonomous manner, driving innovation, interoperability, and scalability across various domains and use cases. In this new and challenging context, some of the additional requirements that are important and might be integrated and taken into account in the design of discoverability solutions are:

- **Unique Identifier:** Each DT shall have a unique and standardized identifier, making it addressable in a software space. This identifier should be associated with the corresponding PTs or physical entities associated to the digital replica.

- **Metadata Inclusion:** DTDs shall include comprehensive metadata about DTs, including their type, description, associated PTs, and relevant contextual information. Metadata enhances the understanding and context of DTs.

- **Resource & State Description:** DTs shall provide resource descriptors that outline their properties, relationships, events, and behaviors/actions. These descriptors should be structured, standardized, and machine-readable.

- **Standardized Representation:** DTDs shall be represented in a standardized format or data model to facilitate interoperability across different platforms, systems, and implementations.

- **Semantic Annotations:** DTDs shall incorporate semantic annotations to enhance the interpretation of DT capabilities and interactions. This promotes semantic interoperability and supports meaningful integration and reasoning.

- **Distributed Discovery Mechanisms:** Implement distributed discovery mechanisms that enable the autonomous discovery of DTs within a network or environment. These mechanisms should consider scalability and decentralized approaches.

- **Centralized Directory Services:** For scenarios where a centralized directory service is appropriate, DTDs shall provide information necessary for directory services to maintain up-to-date listings of available DTs.

- **Cyber-Physical Awareness:** DTs shall offer detailed descriptions of their resources, capabilities and entanglement monitoring allowing applications and digital services to explore and understand their functionalities and being aware of their performance and cyber-physical quality over their life cycle.

- **Supported Communication Protocols:** DTDs shall specify the communication protocols and data formats supported by DTs, both for the digital and physical communication layers. This information aids in selecting appropriate interaction patterns.

- **Security & Access Control Information:** Include security-related information in DTDs, such as authentication mechanisms, access control policies, encryption methods, and data protection measures. This ensures secure interactions with DTs.
- **Versioning and Change Tracking:** DT shall enable versioning and change tracking for DTDs to manage updates and modifications over time. This is essential for tracking changes to a DT's properties, behaviors, and interfaces and allows to compare different versions of the same twin over time.
- **Integration with Existing Standards:** DT shall allow for integration with other existing standards and description formats, such as WoT or oneM2M, to promote compatibility and alignment with established IoT and cyber-physical system ecosystems.
- **Resource and Service Indexing:** DT shall implement resource and service indexing to facilitate efficient querying and discovery of specific DT capabilities or services within a DT ecosystem.
- **Query and Search Mechanisms:** DT shall enable query and search mechanisms for DTDs, supporting advanced search criteria based on properties, relationships, and other contextual information. These approaches can be supported both a single twin to support the discovery of a target property, events, relationships or actions or at an inventory level where multiple DTDs can be queried.
- **DTD Updates:** Ensure that DTDs and DT discovery mechanisms support live synchronization and updates to reflect changes in the availability, state, or properties of DTs. Enable DTDs to dynamically adapt to changes in the DT ecosystem, accommodating the addition or removal of DT instances.
- **Cross-Domain Compatibility:** Design DTDs and their discoverability to be cross-domain compatible, allowing them to function seamlessly across different application domains and industries.
- **Machine-Readable Documentation:** Provide machine-readable documentation for DTDs to aid developers and systems in understanding and interpreting the DT's capabilities.
- **Interoperable Ontologies:** Support interoperable ontologies or semantic models for describing DTs to enhance semantic understanding and reasoning about DT interactions.

These requirements collectively contribute to the development of effective DTD discoverability mechanisms, enabling applications and digital services to explore, interact with, and leverage DTs seamlessly and autonomously in various contexts and use cases.

Annex A (informative): Change History

Date	Version	Information about changes
2023-05	0.0.1	Early Draft with main section and subsections
2023-09	0.0.2	Stable Draft with revised sections and improved Section 4
2023-09	0.0.3	Stable Draft
2023-10	0.0.4	Stable Draft R1
2023-12	0.0.5	Final Draft
2024-01	0.0.6	TBapproved draft, Technical Officer review before EditHelp publication pre-processing, STF 628 Leader review
2024-02	1.1.1	First published version

History

Document history		
V1.1.1	February 2024	Publication