Based on the distributed and collaborative DTs, these should also be integrated and run in widespread infrastructures. Moreover, the effective distribution of resources to prevent bottlenecks and ensure equitable access to technology across different regions and industries is crucial. The challenge also involves overcoming resistance to adoption, which can stem from concerns over cost, complexity, or disruption to existing workflows. Establishing clear standards and providing training and support are essential to facilitate adoption and maximize the benefits of DTs for all stakeholders.

Obj. 8 is to ensure value-added and European impact, and **increasing EU skills and capacity through the development-driven innovation community**.

Here we will also formulate a road-map for the future sustainability of the project. It is a general challenge to secure significant European impact, enhancing EU skills and capacity through innovation. A further, connected challenge is to develop and execute a strategic roadmap for the future sustainability of the **DT^{2}OPS** project. This multifaceted challenge requires aligning the project objectives with the broader goals of European economic and technological advancement to ensure that the outputs significantly contribute to the EU's competitive edge globally. Enhancing skills and building capacity within the EU implies educational initiatives and training programs that can equip the workforce with the necessary expertise in advanced digital technologies. Additionally, fostering an innovation community involves creating collaborative networks that encourage knowledge exchange and support among businesses, academia, and government entities. The development of a strategic roadmap is imperative for outlining clear, actionable steps towards long-term sustainability, requiring careful planning to address technological trends, market needs, and regulatory changes. This roadmap must also include mechanisms for monitoring progress and adapting strategies to evolving circumstances, ensuring the project's relevance and effectiveness over time.

1.1.3 Ambition of the proposed work

The following subsections detail the technological advancements envisioned by the project as a whole. Project contributions to the state of the art (SoA) are enumerated in Table 1.1 and **DT^{2}OPS** objectives related to a specific contribution are marked.

Table 1.1: **DT^{2}OPS** contributions to SoA and their relations to project objectives

Targeted technological advancements	Short name	Objectives							
		1	2	3	4	5	6	7	8
DTs and Physical counterparts	DTPH	✗					✗	✗	✗
Design, development, and operations of DTs for System of Systems	DT for SoSOps		✗				✗	✗	✗
Beyond Modeling Digital Twins – The Mod/Dev-Ops approach	DT Modeling		✗				✗	✗	✗
Collaborative Digital Twins and Translations	Collab DT		✗				✗	✗	✗
Engineering Process for CPSoS	CPSos			✗			✗	✗	✗
The Olympics Model of the **DT^{2}OPS** Lifecycle	DT Lifecycle			✗			✗	✗	✗
Digital Twin - GitOps	GitOps			✗			✗	✗	✗
AI in the Computational Twin – Bridging the simulation-to-reality Gap	Computational Twin			✗			✗	✗	✗
Digital Twins and Operations Copilot	Copilot for DT			✗	✗		✗	✗	✗
Involving Humans in Digital Transformation	Human DT				✗		✗	✗	✗
Digital Twins to Digital Quintuplets	Quintuplets				✗		✗	✗	✗
Privacy, Security and safety in the DT realm	PiSaSec					✗	✗	✗	✗

DTs and Physical counterparts

Digital Twins are digital replica of a physical entities. DTs can provide a rich representation of their corresponding physical entities and enables sophisticated control for various purposes. In practice, DTs can be applied in vastly different ways leading to different architectural designs. The **DT^{2}OPS** project aims to significantly advance the implementation of DTs in computing systems, taking into account the hardware and embedded system requirements necessary for seamless and efficient operation. DTs and their connections must accurately mirror the real-time status and conditions of their physical counterparts. This connection is essential for providing a detailed and functional representation that supports sophisticated control mechanisms for various industrial and technological purposes. Within the **DT^{2}OPS** framework, ensuring that these digital replicas can perform their functions reliably is a key requirement. To address the challenges inherent in the construction and operation of DTs, the project is dedicated to refining the design and creation processes of these systems. This includes the development of methods to effectively link physical objects with their digital equivalents, particularly in instances where no prior DT exists.

DT^{2}OPS breaks down this task into several key phases: selecting functional requirements, searching for and utilizing existing models from the consortium's catalogue, implementing any missing DTs, and rigorously verifying the accuracy and reliability of the models in use. **DT^{2}OPS** aims to explore the problem through several phases: from functional requirement selection, search of existing models in the consortium catalogue and implementation of missing DTs and verification of the used models. We address as well how physical components exchanging real-time information with DTs, and experimental platforms to build DTs (including protocols and standards).

Real-time data exchange between the physical components and their DTs is essential. While **DT^{2}OPS** does not directly target the advancement of the ICT infrastructure, the project demonstrators utilize and count on the related capabilities.

> Modeling and simulation of networking and connectivity for the DT is an integral part of the **DT^{2}OPS** project. The consortia will address how physical components exchange real-time information with their DTs, as well as experimental platforms to build DTs (including protocols and standards).

Depending on the specific needs of the project partners, these adaptations might involve single or multiple alterations across the system's architecture. By addressing these technical aspects, **DT^{2}OPS** aims to deliver robust, reliable, and real-time capable DT that can significantly enhance the control and operational capabilities of complex systems across various sectors. To this end, catalogues of digital twins indexing their designated use cases, architecture design patterns, usage examples, vendor and other functional and non-functional characteristics will help to utilize DTs in its application and reuse [8]. Physical Twins and Digital Twins require detailed descriptions regarding their capabilities, connections, limitations. These can be listed in catalogs so engineers and technicians can decide upon the usage of certain DTs. Such catalogues already exist. Examples include Semantic Stack by Bosch, or Change2Twin by Siemens and TTTech. Nevertheless, these DT catalogues are usually vendor-specific or available for a closed domain [9]. **DT^{2}OPS** project will create a specific DT catalogue compiled in an open and extendable way, such as described through Eclipse Vorto.

The **DT^{2}OPS** project considers the adaptability of various system tools to support the specific requirements of DTs. Modifications to compilers and operating systems are explored to ensure these tools can integrate and function effectively with the host systems. Depending on the specific needs of the project partners, these adaptations might involve single or multiple alterations across the system's architecture.

> Physical Twins and Digital Twins require detailed descriptions regarding their capabilities, connections, limitations. These can be listed in catalogs so engineers and technicians can decide upon the usage of certain DTs. Such catalogues already exist. Examples include Semantic Stack by Bosch, or Change2Twin by Siemens and TTTech. Nevertheless, these DT catalogues are usually vendor-specific or available for a closed domain [9]. **DT^{2}OPS** project will create a specific DT catalogue compiled in an open and extendable way, such as described through Eclipse Vorto.
> In this context, DT^{2}OPS exceeds the current state-of-the-art by providing a scalable, adaptable, and diverse repository of all possible open-source hardware counterparts. These features allow DT^{2}OPS to evolve and expand, especially through integration with other systems, enabling AI-assisted optimization for DT composition, semantic search and analysis, and dynamic security improvements. The framework also seamlessly integrates with third-party systems that automatically update digital twins by scanning the DT catalog for updates. Additional functionalities, such as (1) dynamic adjustment of DT operations based on real-time environmental conditions, (2) integration with technologies like AR/VR, and (3) the creation of blockchain-based DTs with built-in IP protection, exemplify its potential for third-party system and technology integration. Through these innovations, DT^{2}OPS not only surpasses current standards but also establishes an adaptive, scalable platform poised to meet the challenges of future digital twin development.

Requirements and overall ambition

In order to provide a framework for designing, developing, connecting, integrating the DTs, and address the issues of Collaborative and Distributed Digital Twins (as defined by Obj. 2), the following requirements are identified to be covered:

Req. DTSoS.1 DT integration into a System of Systems (SoS) environment

Req. DTSoS.2 DT reflection accuracy and timeliness with the physical space

Req. DTSoS.3 DT modeling descriptors, languages, and tools

Req. DTSoS.4 DT systems collaboration for services composition

Req. DTSoS.5 DT system evolvability as individual DTs are changed, added and removed

Req. DTSoS.6 DT mobility and migration in the cloud-to-edge compute continuum

Req. DTSoS.7 DT replication for resilience, fault-tolerance, security, scalability, & availability

In support of the structured evolvement of the concepts of DTs, efforts have been made to identify some key properties that are of high importance in the context of DTs. In addition, by recognizing DTs as part of a System of Systems (SoS), some of these properties can be further refined by recognizing each instance of a DT as a constituent system in an SoS. According to [10] and based on the original definition of SoS by Mayer in 1998 [11], an SoS must satisfy five characteristics: (i) the operational and (ii) managerial independence of constituent systems; (iii) geographical distribution; (iv) emergent behavior; and (v) evolutionary development processes. This means that (Req. DTSoS.1) a DT framework must support the integration into an SoS, including mechanisms for the discovery and use of services, the sharing of data and resources, the support for extra-functional properties and the late binding of service consumers, providers and prosumers.

Challenges associated with DTs relate to the fidelity of the two-way synchronization between the physical and virtual spaces, the interoperability with existing software being used in production, cybersecurity concerns, and the add-on costs related to implementing profitable solutions based on DTs [2]. Built on these observations and other works on the desired properties for DT systems [12, 13], we identify that (Req. DTSoS.2) DTs should assure the accuracy and timeliness properties between the DTs and the physical objects they represent. They should also be aware of entanglements that challenge the synchronization between the physical and virtual spaces, as well as the synchronization between DTs in case they collaborate to compose more advanced services.

The key enabling technology to design and operate complex, composable DTs is advanced modelling (Req. DTSoS.3). Recent studies [14] show that the complexity of IoT-based projects rose to the level that time to market for connected products increased 80 % since 2020. Managing complexity with model-based systems engineering in the industry has a long-standing history with many proven approaches. In the domain of DTs, mainly component models were used to design and represent knowledge virtually. Various open and closed-source projects target data or domain-specific modelling. However, as the need for SoS DTs appeared, these approaches turned out to be insufficient. We must introduce modelling technologies to design and integrate sub-system level DTs to build and operate SoS DTs. Novel language support and design tools are needed for the development: from the specification phase until the operation, providing support also for the adaptability of the SyS DT to the evolving environment. Physical layer, hardware, software and the service level have to be integrated into a SoS DT, and system-level requirements have to be ensured with the help of the SoS DT, such as security, safety or reliability.

Furthermore, (Req. DTSoS.4) DT systems need to be capable of collaboration to compose services based on different types of elementary DTs that represent physical entities, objects, and systems, with potentially multiple levels of aggregated DTs that bring together sets of elementary DTs to create the composed services. This need connects closely to the observation of that a DT should be seen as a constituent system in an SoS, which means that (Req DTSoS.5) a DT system must be capable and prepared to evolve as individual DTs adapt to remain in synch with the physical space.DTs systems are expected to operate in the IoT-edge-cloud (IEC) continuum. Thus, (Req. DTSoS.6) DTs must be capable of mobility to migrate to compute domains that best fit current application requirements, e.g., on real-time synchronization. In addition, (Req. DTSoS.7) DTs should support replication for purposes such as resilience, fault-tolerance, security, scalability, and availability.

Ambition beyond SoA – We aim to advance beyond the state of the art related to the here-identified challenges related to Req. DTSoS.1-.7. These challenges will be addressed by Work Package 2 and in relation to the specific objective 2.

- Related to Req. 2.1 and 2.2. Existing container orchestration mechanisms are not prepared to address cyber-physical applications, e.g., no clear support for status updates from the physical space [12]. We will progress beyond SoA by defining a framework for designing, developing, connecting, and integrating DTs into a SoS. We aim to provide guidance on how to ensure the needed fidelity of the two-way synchronization between DTs and surrounding cyber-physical systems.
- Related to Req. 2.3. New generation modelling languages, such as the SysML v2 language, are not yet applied for the design and support of the operation of SoS DTs. We will develop new functional verification methods and extra-functional analysis techniques to provide a full-fledged model-based development platform for SoS DTs.
- Related to Req. 2.3. Generative modelling techniques – that extend generative AI techniques with graph-model-based reasoning – have not yet been explored for the support of DT design. We will investigate the generation of SoS DT models from textual specifications to automatize the design of SoS DTs, supporting engineers in the construction of complex SoS DTs. The development time of SoS DTs will be significantly reduced.
- Related to Req. 2.4 and 2.5. A layered architecture for DTs as proxies to cyber-physical devices and for composing SoS with collaborative DTs has been proposed in [4]. These concepts and design principles have been validated in lab environments. We aim to refine the work and elevate the solutions for demonstration in industrially relevant environments of the use case, targeting prototype demonstrations in operational environments.
- Related to Req. 2.6 and 2.7. Resources in the IEC continuum need to be (cost) effectively and efficiently orchestrated, preferably seamlessly to reduce the effort needed to operate a SoS in this continuum. Colony OS is an open-source meta-operating system designed to improve the integration and utilization of diverse computing platforms, including IoT, edge, cloud, and HPC [15]. We will progress beyond SotA by integrating Colony OS with the layered architecture for DTs as proxies, aiming at prototype demonstrations in operational environments of the use cases.

Beyond Modeling Digital Twins – The Mod/Dev-Ops approach

The Software Engineering Institute at Carnegie Mellon University introduced the concept of TwinOps, where Digital Twins meet DevOps. Figure 1.4 depicts the main ingredients of the TwinOps approach. The Digital Twin engineering approach is supported by the model-based systems engineering workflow. The Mod/Dev part (left side of the Figure 1.4) represents the model-based development, while the Ops part (right side of the figure) shows the steps and tasks during operation. The whole process uses models as main artefacts: SysML v1.6 and AADL models support the design procedure, Modelica is used to represent the physical word and simulation technologies enable the analysis at the system level. Model-Based System Engineering (MBSE) became the backbone of the Digital Twins. The novel approach introduced automatic code generation to support the development.

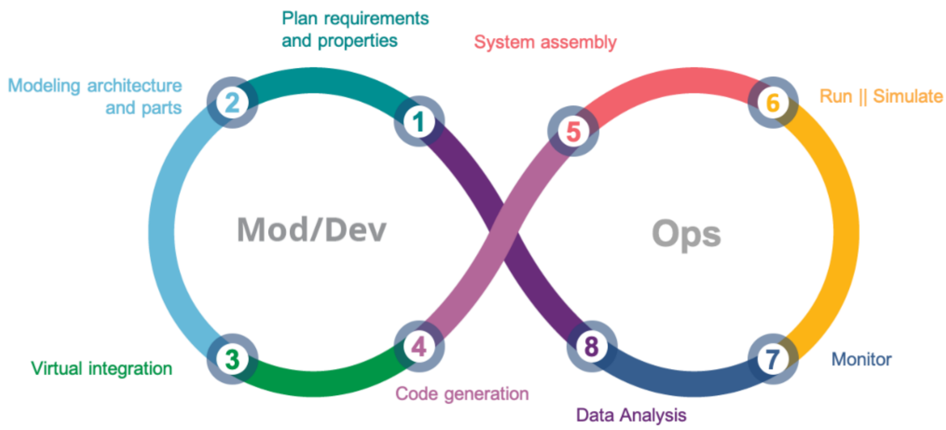

Figure 1.4: ModDevOps is a systems/software co-engineering culture and practice that aims at unifying systems engineering (Mod), software development (Dev) and software operation (Ops). The main characteristic of the ModDevOps is to strongly advocate abstraction, automation, and monitoring at all steps of system construction.

However, by relying on the traditional systems engineering languages and traditional modelling techniques, there is a huge potential to extend the TwinOps approach.

> In the **DT²OPS** project, the state-of-the-art will be improved by offering new generation systems and data modelling languages for the engineers to support the design phase. Generative modelling technologies will provide automatic construction of design models from specifications, and the project will provide design space exploration technologies to explore the space of the design decisions and find the best candidate according to the extra-functional requirements. Critical applications of DTs also necessitates that the DT representation of the SoS have to be verified both at design time and runtime. Novel formal verification methods will be tailored to the new generation system and data description languages to provide verification for the SoS DTs.

Beside extending system level DTs, the project aims the development of new technologies at the component level. The emerging of RISC-V-based chips in the industry necessitates the development of functional DTs to support the verification of RISC-V-based systems.

> The project will provide formal models for certain aspects of RISC-V-based embedded systems to support the co-engineering of hardware and software. In addition, the DT of an industry grade embedded real-time operating system will also be developed to help the developers engineering functionally correct services on top of the infrastructure.
> Another direction of component level DTs aims the representation of AI systems. As the AI techniques are gaining more and more application in the DT domain, the project aims the development of model-based DT representations for critical AI applications. Quantitative graph-based models represent the requirements for the AI system and serves as a runtime DT to ensure correct reasoning even in deep-learning/neural network-based AI systems. This novel application of the concepts of DTs have never been explored before.

Collaborative Digital Twins and Translations

Digital Twin solutions are offered mostly as Software as a Service (SaaS), while some solutions (mostly open source) can be self-hosted and/or self-managed. State of the art products and projects are considered and summarized in Table 1.2.

These solutions are not interoperable from a data model point of view since each have their own Domain Specific Language (DSL). These custom DSLs create significant additional effort to integrate the services into other production systems or engineering tools. Each tool having its own data model requires one-to-one translation to target data formats which is not scalable, nor maintainable. In the **DT²OPS** project, an interoperability layer is to be considered and developed, as depicted in Figure 1.5.

> In the realm of AI/ML, Large Language Models have shown initial promise with current state of the art to be able to understand, generate and potentially translate DSLs. There are several techniques that can be researched for generic cases (e.g. also in AIMS 5.0 project), but there is no specific work done on Digital Twins and DT model translations.
> These LLM-based DSL translation techniques can be considered, from fine-tuning of models with examples, through Low-Rank Adaptation of models up to supplying fundamental models with Retrieval Augmented Generation (RAG) for effective translation.

Moreover, some LLMs are better tailored to structured output (e.g. JSON or XML), then others. In the **DT²OPS** project, DT interoperability needs to be researched to mitigate costs around a lot of DT languages and solutions, so European Industry can better understand and utilize these products and open source solutions.

Table 1.2: Comparison of state of the art Digital Twin solutions

Digital Twin solution	Vendor	Open/closed source	Deployment	3D modeling	DT Domain Specific Language	Source
TwinMaker	AWS	Closed	SaaS	Yes	Custom JSON	AWS IoT TwinMaker
Azure Digital Twin	Microsoft	Closed	SaaS	No	DTDL v3 (JSON)	Azure Digital Twins
Vorto	Eclipse	Open-source	Self-hosted	No	Vortolang (YML)	Vortolang
Ditto	Eclipse	Open-source	Self-hosted	No	WoT DSL	
Hono	Eclipse	Open-source	Self-hosted		WoT DSL	
Omniverse DT	Nvidia	–	SaaS or Self-hosted	Yes	OpenUSD	Nvidia Omniverse
Things	Bosch	–	SaaS	No	Based on Eclipse Ditto	Bosch Things Intro

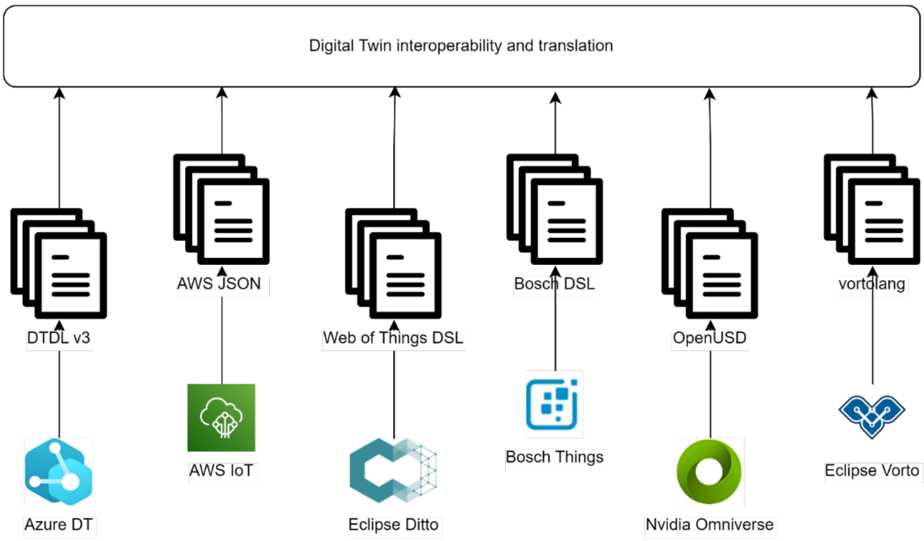

Figure 1.5: Proposed interoperability layer in the **DT²OPS** project to address challenges with integrating services using different Domain Specific Languages (DSLs).

Engineering Process for Cyber-Physical System of Systems

The engineering process for complex Cyber-Physical Systems of Systems (CPSoS) significantly benefits from standardized process models, particularly in industrial engineering. One such model is defined by the ISO/IEC 81346 standard [16], which provides a structured approach to automation engineering. Building upon this, the Arrowhead Tools Engineering Process (AHT-EP), introduced in previous work [17], extends the principles of ISO 81346 to be more dynamic and service-oriented, making it highly applicable for engineering CPSoS solutions.

At the core of AHT-EP is a design philosophy that balances flexibility and adaptability with the robustness of traditional engineering process models. This balance ensures that AHT-EP can be applied across a wide range of industrial domains, from manufacturing and energy grids to logistics and production systems.

The evolution of engineering tools is essential in this process. While traditional tools remain valuable, the future of engineering will require incorporating AI/ML-supported technologies. Examples of these tools include:

LLM-based copilots Supporting engineers during the Functional Design phase, these AI-powered assistants can enhance creativity, accuracy, and speed by providing context-aware suggestions or automating repetitive tasks.

Deep Neural Networks For Operations and Management, DNNs can perform advanced object detection and real-time monitoring, ensuring more reliable and intelligent control systems.

Anomaly Detection Algorithms During the Maintenance phase, these algorithms can identify potential failures or performance drops, enabling proactive maintenance and minimizing downtime.

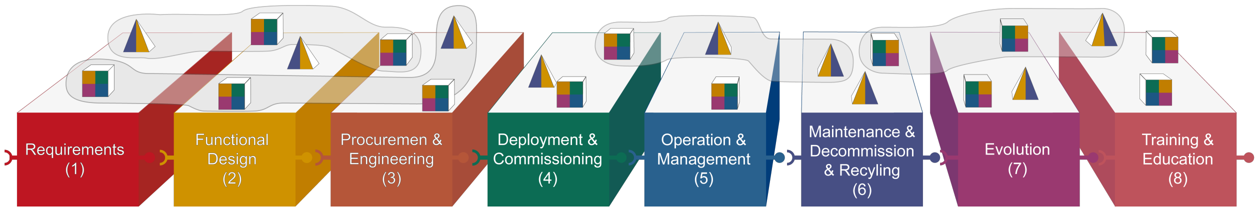

Figure 1.6: Automated output-input connections of several engineering process steps to create toolchains of the Engineering Process for Cyber-Physical System of Systems

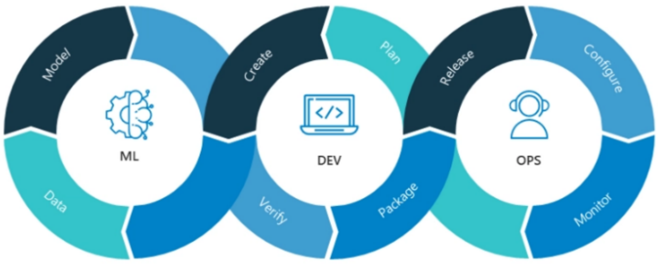

Figure 1.7: The generic model for MLOps

We can categorize the tools in the depicted toolchain, for example the ones that are useful in Functional Design or those can be of major help in Deployment & Commissioning .

The practical benefit of using the AHT-EP is that tools can be associated to each of the eight steps. If we tailor the output of the step$_n$ tool to the input of the step$_{n+1}$ tool, humans can focus on more meaningful tasks than format conversions. Use-case examples for automatic execution of such toolchain steps are provided by [17].

These waterfall-like process models pose, in general, serious requirements on the underlying solution architecture involved: namely that (i) the technology toolchain and platform is relatively stable throughout the cycle, (ii) the infrastructure and deployment requirements are established at project start, (iii) the steady-state operational model can be clearly established at an early stage.

To stay relevant in modern industrial environments, the AHT-EP must evolve to fit the current AI/ML-supported productization lifecycles. This requires integrating the AHT-EP with both the MLOps toolchain and Digital Twin (DT) lifecycles.

For instance, the creation, operation, synchronization, and validation of Digital Twins must be part of the engineering process. DT models require continuous production data to remain accurate, and their outputs can be reused to train and validate other machine learning applications. This feedback loop between DT and ML systems ensures the ongoing improvement of both, enhancing operational performance and adaptability.

> In summary, by aligning the AHT-EP with modern AI/ML methodologies like MLOps and Digital Twins, we enable a more adaptive and intelligent approach to CPSoS engineering, where continuous learning, automation, and validation drive improvements across the entire system lifecycle.

The Olympics Model of the DT^{2}OPS Lifecycle

As the proposed process goes, the whole MLOps lifecycle needs to be augmented with the requirements driving the Digital Twin and Systems Engineering design flow. As described earlier, the challenge is bringing the AHT-EP and other CPSoS engineering standard development processes together with the ML-Dev-Ops cycles of the organization.

The MLOps lifecycle, while highly effective in managing machine learning and operations, must be extended to meet the unique requirements of Digital Twin development and Systems Engineering processes. One of the core challenges in modern industrial environments is aligning the Advanced Human Technology-Enhanced Process (AHT-EP) and other Cyber-Physical Systems of Systems (CPSoS) engineering standards with the iterative Machine Learning (ML) development and operations cycles, commonly referred to as ML-Dev-Ops.

This expanded approach is visually represented by the Olympics Model of the **DT^{2}OPS** lifecycle (Figure 1.7), symbolizing the intertwined nature of these distinct but interdependent processes.

The Olympics model assumes a collaborative, agile delivery process where various IT and OT teams work together. This unified framework allows for the continuous deployment and orchestration of different components of CPSoS, ranging from embedded systems to control systems and enterprise applications. While the MLOps cycle phases are well-established, we propose several important extensions that address the specific needs of Digital Twins and Systems Engineering as depicted in Figure 1.8.