Ref. Ares(2023)3700498 - 29/05/2023

Standardization Recommendations

Deliverable number: (D.4.7)
Author(s): Abdelhadi Belfadel (ISX), Sebastian Horl (ISX)
Author’(s’) affiliation (Partner short name): ISX

THE CIVITAS INITIATIVE IS CO-FINANCED BY THE EUROPEAN UNION

Deliverable No.		**4.7**	
Work Package No.	**4**	Work Package Title	**Standardization Recommendations**
Task No.	**4.5**	Task Title	**Standardization Recommendations**
Date of preparation of this version:		**11/05/2023**	
Authors:		**Abdelhadi Belfadel (ISX)** **Sebastian Horl (ISX)**	
Status (F: final; D: draft; RD: revised draft):		**F**	
File Name:		**LEAD D4.7_Standardization_Recommendations.docx**	
Version:		**1.0**	
Task start date and duration		**01/10/2021 – 31/05/2023**	

This document is issued within the frame and for the purpose of the LEAD project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 861598.

Revision History

Version No.	Date	Details
0.1	11/04/2023	1st draft version (ISX)
0.2	11/05/2023	Internal review ISX
0.3	12/05/2023	Second internal review ISX
1.0	25/05/2023	Project's Internal review

Reviewers List

Name	Company	Date	Signature
Angel Batalla	LMT	27/05/2023	
Bas Groothedde	ARG	24/05/2023	

Contents

List of figures

List of tables

List of acronyms and abbreviations

Acronym /Abbreviation	**Meaning**
ABB	Architecture Building Block
DDDAS	Dynamic Data-Driven Application System
DSS	Decision Support System
DT	Digital Twin
GDPR	General Data Protection Regulation
KPI	Key Performance Indicators
LL	Living Labs
LSP	Logistics Service Providers
MMT	Multimodal Transport
PPP	Public-Private Partnerships
SBB	Solution Building Block
TLS	Transport Layer Security
UML	Unified Modeling Language
MLib	Model Library

1. Introduction

1.1 LEAD project overview

LEAD created Digital Twins of urban logistics networks in six cities, to support experimentation and decision making with on-demand logistics operations in a public-private urban setting. Innovative solutions for city logistics are represented by a set of value case scenarios that address the requirements of the on-demand economy while aligning competing interests and creating value for all different stakeholders. Each value case combines several measures (LEAD strategies): a) innovative business models, b) agile urban freight storage and last-mile distribution schemes, c) low-emissions, automated, electric, or hybrid delivery vehicles, and d) smart logistics solutions.

Cost, environmental and operational efficiencies for value cases have been measured in 6 Living Labs (LL). Evidence-proven value cases and associated logistics solutions have been delivered in the form of exploitable Digital Twins, incorporating the models that support adaptation to different contexts and that provide incentives for PPPs.

The LEAD consortium comprises 22 partners, all of whom are involved in the Living Labs, supported by 5 international partners for knowledge transfer. This structure incentivizes the co-creation of solutions by city authorities, logistics industry leaders, start-ups, and research experts in freight modelling, complex simulation and logistics optimization.

1.2 Deliverable purpose and scope

This report is developed in the context of WP4 - LEAD Adoption and Growth Roadmap which maximises project's visibility and develop an ecosystem of stakeholders, including EU institutions, public authorities, industry actors and SMEs that will actively contribute to the project success.

The purpose of this document, D4.7 Standardization Recommendations, is to implement a set of recommendations for standardization dedicated to the LEAD Digital Twin framework focusing on urban logistics. We will provide a reference architecture building a basis for future implementations of a Dynamic Data-Driven Application System (DDDAS) for decision making in order to organize goods distribution operations in cities.

1.3 Addressing the LEAD Description of Action

The following table maps the contents of this document to the requirements of the GA.

Table 1 Deliverable's adherence to LEAD objectives and Work Plan

LEAD requirements	Section of D4.7 addressing LEAD GA	Comments
D4.7 Standardization Recommendations (ISX, M36) Recommendations report to standardisation groups and bodies		
-Implementation a set of recommendations for standardization dedicated to the LEAD Digital Twin framework focusing on urban logistics	1 3	Show how the LEAD reference architecture links to existing standards
-Reference architecture building a basis for future implementations of a Dynamic Data-Driven Application System (DDDAS) for decision making in order to organize goods distribution operations in cities	3.2 3.3 3.5.1	Description of LEAD's reference architecture
The structure of the system with its elements as well as the interactions between each element and the environment will be described.	3.3 3.4 3.5	Description of LEAD's reference architecture
- The architecture will be based on the ISO/IEC/IEEE 42010 "Systems and software engineering - Architecture description" standard, addressing "the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions".	1.4 2 3 4	Addressing the components of the standard through the TOGAF definitions in the LEAD reference architecture

1.4 Target audience

This document is public and is aimed at the project partners, individuals in those organizations, the EU, EU Reviewers, and any individual who wishes to gain insight into the digital twin architecture focusing on urban logistics.

1.5 Deliverable context

This task gathers main recommendations for providing a reference architecture for establishing future implementation of DT in urban settings. Its relationship to other tasks and deliverables ([2], [3], [4], and [5]) is as follows (see Figure 1):

- **T4.2 International Collaboration - Liaison with other EU projects and initiatives:** This task will define and prepare a clustering plan with associated activities and promoting the exchange of results, sharing knowledge, experience and best practices.
- **D4.3 Communication and Outreach Activities:** T4.3 will build a vibrant ecosystem around the project targeting EU institutions, public authorities, industry players, SMEs.
- **D4.4 Advisory Board, Transferability Platform and Policy Recommendations:** Through the Digital Twinning it is possible to visually define scenarios: the instrument is particularly effective from a communication perspective but needs the proactive participation and understanding of local stakeholders. At the same time, the approach is easily transferable to different cities and local contexts.
- **T3.2-T3.7 Living Lab Implementation:** Theses tasks prepare and implementation of the 6 LLs.
- **T3.8 Impact Assessment, Solutions Validation and Learning Conclusions:** The overall goal of this task is to assess the potential of LEAD solutions to achieve low-emission adaptive last mile logistics.

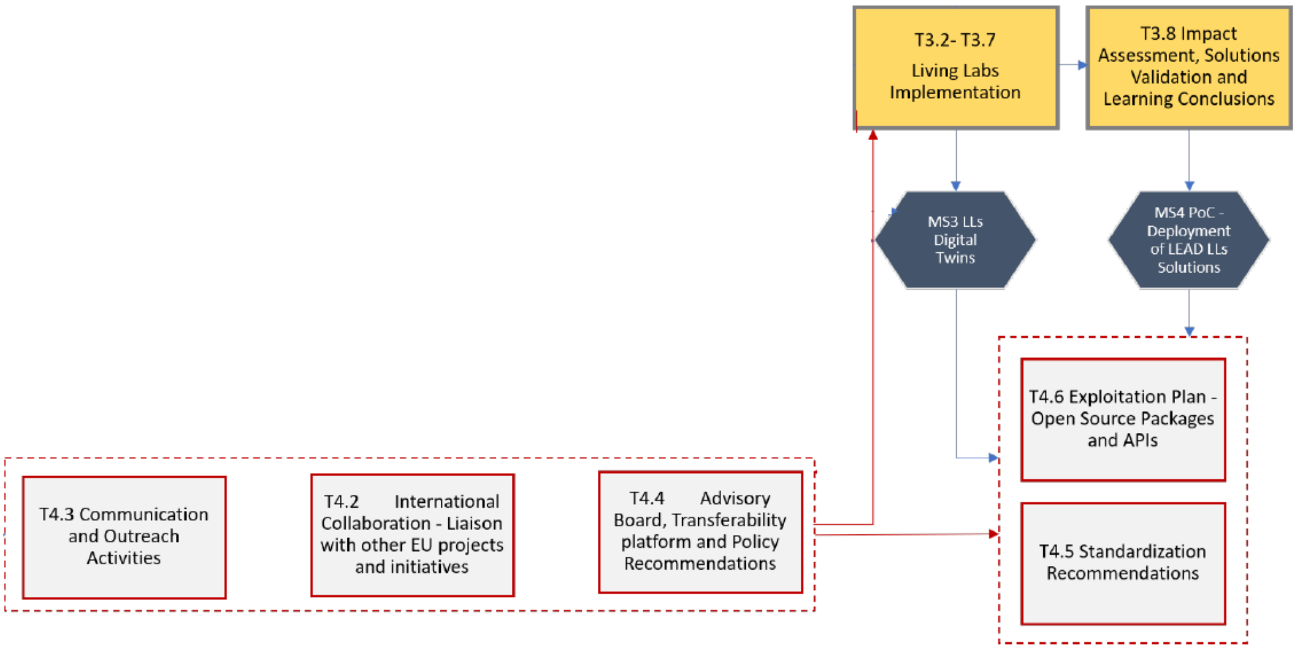

Figure 1 Deliverable (D4.7) dependencies

1.6 Document Structure

This deliverable is broken down into the following sections:

- **Architecture vision**: This first phase ensures that the architecture is considered as a whole, by creating architecture content by cycling through the business, information system, and technology architecture. This enables to provide a big-picture of the targeted LEAD Platform, its main business process, the high-level architecture, and the implementability of the architecture. This phase helps as well to converge to a target and refine each level during the next iteration of the architecture development method.

- **Business Architecture**: Based on the TOGAF recommendations [1], this phase enables to identify and analyse the Architecture Building Blocks (ABBs) that captures the architecture requirements to guide the development of the required solution building blocks. Each component in LEAD is considered as an ABB, considering, in the next architecture levels, several concepts such as the business, data, application, and technology requirements, the fundamental functionality and attributes (behaviour, interfaces provided, including security aspects), interoperability (where applicable) and dependence between building blocks.

- **Information System Architecture**: This architecture level is focusing on the internal structure of the identified components during the business architecture, along with the required data structure and component interactions. It enables also to analyse technically the identified components that should cover the targeted features. We analyze the different solutions that exist in the market referred to as Solution Building Block (SBB) to implement the required component. Each SBB will be evaluated in the form of '+++' and '---' to cover the functionality, parameters, and security. The more '+', the better a solution will be covering such aspects, while more '- ', the less likely a solution is going to be selected for reuse. In terms of reusability, the tool evaluated are ensured to have an open-source and reuse/extend of source code open enough to allow the exploitation of such tool beyond LEAD.

- **Technology Architecture**: This last phase enables to define the communication endpoints of the identified components in the Information System Architecture to be used during the implementation. The focus is on API definitions and common data models that are provided for LEAD inter-communication, as well as the security aspects and the infrastructure that supports the LEAD platform.

1.7 Architecture Framework standards and methodologies

In this section we present the methods and tools used to support this architecture development effort. The goal is to design and consolidate a reference architecture for city-oriented digital twin systems.

1. ISO/IEC/IEEE 42010

ISO/IEC/IEEE 42010 is an International Standard titled: Systems and software engineering — Architecture description. The Standard, published in 2011, is the result of a joint ISO and IEEE revision of the earlier IEEE Std 1471:2000: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems [6]. IEEE 1471 was developed by the IEEE Architecture Working Group under the sponsorship of the IEEE Software Engineering Standards Committee. In September 2000, the IEEE Standards Board approved IEEE 1471 for use. In March 2006, IEEE 1471 was adopted as an ISO standard. It was published in July 2007 as ISO/IEC 42010:2007. Its text was identical to IEEE 1471:2000. ISO/IEC/IEEE 42010:2011 replaces both ISO/IEC 42010:2007 and IEEE Std 1471:2000.

2. What does the Standard standardize?

The original IEEE 1471 specified requirements on the contents of architecture descriptions of systems. An architecture description (AD) expresses the architecture of a system of interest. An AD could be a document, a repository or a collection of artefacts used to define and document an architecture [6]. ISO/IEC/IEEE 42010 adds definitions and requirements on architecture frameworks and architecture description languages (ADLs) as conformance cases.

The Standard defines architecture as fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.

3. Conceptual Model of the ISO/IEC 42010

The Standard has several parts. The first part is a conceptual model of architecture descriptions (ADs). The conceptual model, sometimes called the “metamodel”, defines a set of terms and their relations. This metamodel has been widely used and discussed in industry and in the academic literature. The conceptual model establishes terms, their definitions and relations which are then used in expressing the requirements.

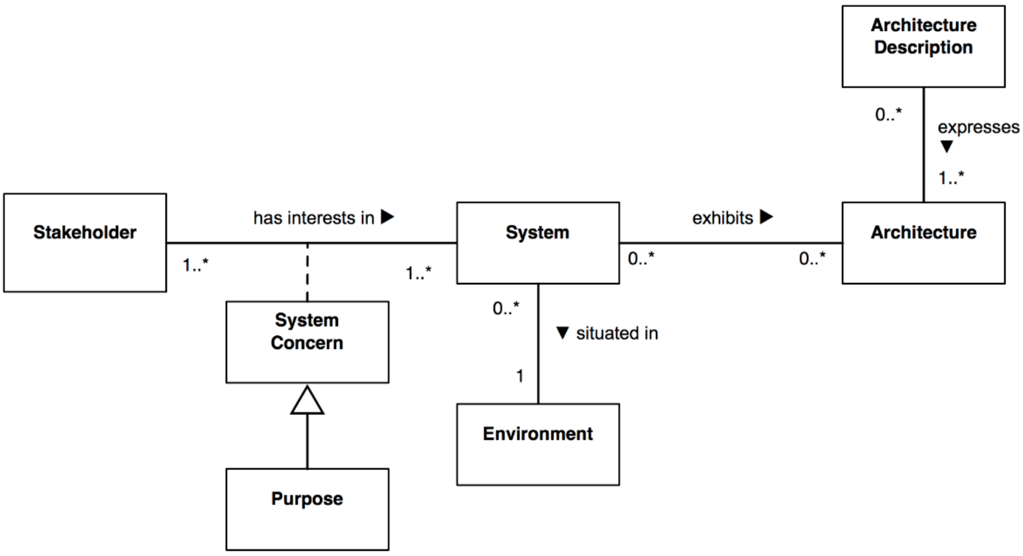

Figure 2 Meta-Model of the ISO/IEC/IEEE 42010 [7]

As described by [6], the elements of the diagrams refer to:

- **System:** A System is situated in its Environment. That environment could include other Systems [6].

- **Stakeholders** have interests in a System; those interests are called Concerns. A system’s Purpose is one very common Concern. (Note: In the original edition of the Standard, systems were said to have missions; in the revision, Purpose was chosen as a more general replacement to this term.) [6]

- Systems have **Architectures**. An **Architecture Description** is used to express an Architecture of a System [6].

- **System**: The Standard takes no position on the question, what is a system?
  In the Standard, the term system is used as a placeholder – e.g., it could refer to an enterprise, a system of systems, a product line, a service, a subsystem, or software. Systems can be man-made or natural. Nothing in the Standard depends upon a particular definition of system. Users of the Standard are free to employ whatever system theory they choose [6].

- **Environment:** Every System inhabits its Environment. A System acts upon that Environment and vice versa. A system's Environment determines the range of influences upon the system. In the Standard, Environment is intended in the widest possible sense to include developmental, operational, technical, political, regulatory, and all other influences which can affect the architecture. These influences are categorized as Concerns [6].

- **Architecture:** Systems have architectures. In the Standard, the architecture of a system is defined as: "fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution". The definition was chosen (1) to accommodate the broad range of things listed above under System: the architecture of X is what is fundamental to X (whether X is an enterprise, system, system of systems, or some other entity); and (2) to emphasize (via the phrase "concepts or properties") that a system can have an architecture even if that architecture is not written down [6].

- **Architecture Description:** An Architecture Description (AD) is an artefact that expresses an Architecture. Architects and other system stakeholders use Architecture Descriptions to understand, analyse and compare Architectures, and often as "blueprints" for planning and construction. ADs are the primary subject of ISO/IEC/IEEE 42010 [6].

4. Architecture Development Methodology and Architecting Tool

The architecture work in LEAD previous to this deliverable is constructed and consolidated following the TOGAF (The Open Group Architecture Framework) standard and its guidelines, which is an open, industry consensus framework for Enterprise Architecture [1].

TOGAF provides an architecture development method and tools for assisting the acceptance and the production of architecture assets. It is based on an iterative process model supported by best practices and re-usable set of existing assets.

TOGAF embraces the ISO/IEC 42010:2007 terminology and is consistent with the standard. In TOGAF, "architecture" has two meanings depending upon the context:

- A formal description of a system, or a detailed plan of the system at component level to guide its implementation
- The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time

There are four architecture domains that are commonly accepted as subsets of an overall architecture work, all of which the TOGAF standard is designed to support.
The Business Architecture defines the business strategy, governance, organization, and key business processes. **The Data Architecture** describes the structure of an organization's logical and physical data assets and data management resources. **The Application Architecture** provides a blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes. And finally, the **Technology Architecture** describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services; this includes IT infrastructure, middleware, networks, communications, processing, standards, etc.

1.8 Reference standardisation initiatives

Since its introduction, different authors have defined DTs in different terms. The original definition was proposed by [8] who defines a DT as a digital replication of a physical system that is depicted as a distinct digital entity but related to the physical system upon demand. The core elements of a DT are: (1) real world, (2) virtual world and (3) the exchange of data/information flow between real and virtual worlds.

The ultimate objective of introducing Digital Twins in last mile logistics is to improve the operation and efficiency of parcel delivery, reduce costs and externalities through forecasting and predictions of future states and support advanced decision making through the entire logistics lifecycle, while also fostering stakeholder participation via reliable real-life information. Technology enablers for building Digital Twins include modelling, predictive analytics and decision-making methods, and the use of lifecycle-oriented knowledge with historical and real-time operational and city data. The LEAD architecture design therefore considered key standardisation initiatives in logistics, transport data and digital twins to eventually enable the smooth flow of goods and the collaboration among the stakeholders across end-to-end logistics and supply networks.

Many applications of digital twin frameworks are found mostly in the context of the product, shop floor, and production management [9] [10]. The first definition of the DT was forged by the NASA to mirror the life of its flying twin [11]. From that moment on, several initiatives ([12], [13], [14], [15], [16]) tend to standardise and construct conceptual and DT platforms in manufacturing-oriented digital twin systems. One of the recent published standards is the ISO 23247 [12]. It divides the conceptual digital twinning system into four layers. The bottom layer depicts the elements of manufacturing that are observable. This layer depicts the elements of the production floor that need to be modelled. Yet, the difficulty we face today is that existing architectures are either too generic or too specific [10] to be used in logistics due to the complexity of the twinning as mentioned in section 1.

City logistics is one of the essential elements of urban mobility planning. The growth of e-commerce activities is a trend that has been reinforced by the COVID-19 pandemic. The ultimate objectives of introducing DT in city logistics are to improve the operation and efficiency of parcel delivery; to reduce costs and externalities through forecasting and predictions of future states; and to support advanced decision-making through the entire logistics lifecycle, while also fostering stakeholder participation via reliable real-life information. Thus, the adoption of DT in city logistics aims to help stakeholders to take decisions about the future implications of a given policy, aiming at supporting their information and enforcing the decision process [17].

Forecasting both behaviour and reactions to structural changes and policy measure implementations in urban freight transport requires two elements. [17] emphasize the usage of behavioural and simulation models to be used jointly when modelling such future behaviour, especially when considering the public-private urban context.

DTs for urban logistics are evolved systems relying on three essential technical building blocks. First, relevant models for the physical counterpart under study are required. The models must be sufficiently detailed to meet the intended purpose. However, they should not be more detailed than necessary to avoid overfitting and support ease of use. It is worth mentioning that digital twins are not identical twins. As highlighted by [18], the notion of an exact mirror is “an idealization and aspiration that may never be achieved.”

Second, the object of the study must be described by evolving data as physical conditions tend to vary over time, whether it be traffic, air quality, or noise levels. Thus, informed policy decisions must be based on evolving datasets rather than point-in-time data collection that simply provides a snapshot of urban conditions at a given time. In this way, systemic changes over much larger time scales and patterns can be better understood.

Third, like datasets, digital twins also require dynamic updates related to simulation models that update and evolve based on physical conditions [19]. However, [20] and [21] identify the difference among a digital twin, a digital model and a digital shadow depending on the integration level.

In addition to standards adopted in Digital Twins, the Joint Technical Committee (ISO/IEC JTC 1) covers many areas including AI, automatic identification and data capture techniques, cloud computing, data usage, IoT. DTs is also a focus area of IEC Technical Committee 65, which develops international standards for industrial process measurement, control and automation. As mentioned earlier, the Digital Twin - Reference Architecture [12] (PWI JTC1-SC41-5) provides in manufacturing-oriented digital twin systems a standardized generic DT Reference Architecture using a common vocabulary, reusable designs and industry best practices. The document uses a top-down approach, beginning with collecting the most important characteristics of DT along its life cycle, abstracting those into a generic DT Conceptual Model, deriving a high-level system-based reference with subsequent dissection of that model into five architecture views from different perspectives [22]. Modelling work carried out in this task T4.5 is built on the aforementioned initiatives.

2. Architecture Vision

This phase ensures that the architecture is considered as a whole, by creating architecture content by cycling through the business, information system, and technology architecture. This enables to provide a big-picture of the standardized architecture, its main business process, the high-level architecture, and the implementability of the architecture. This phase helps as well to converge to a target and refine each level during the next iteration of the architecture development method.

As a strong point to mention, the following reference architecture for building a basis for future implementations of a Dynamic Data-Driven Application System (DDDAS) for decision making in order to organize goods distribution operations in cities has been accepted for publication in the journal of Computers, Environment and Urban Systems[1]. This journal is an interdisciplinary journal publishing cutting-edge and innovative computer-based research on urban systems, systems of cities, and built and natural environments, that privileges the geospatial perspective.

So far, there is no conceptual framework for a digital twin in urban logistics in literature. While there are conceptual frameworks for other domains as depicted in section 1.7, the ongoing work on LEAD look at the digital twinning processes from a dedicated urban logistics perspective, uncovering, for instance, the need for covering a spectrum from reactive (short-term, in-the-loop) to predictive (long-term, one-shot) models at the same time with their

[1] https://www.sciencedirect.com/journal/computers-environment-and-urban-systems

respective requirements in data acquisition and execution times. A key component is thereby the ability of this reference architecture to adaptively react to changing inputs to update the system state and re-run models on demand. These insights are derived from the requirements of real-world use cases. Finally, we provide practical value by proposing a list of open-source tools with which the targeted business process and the conceptual framework can be brought to implementation.

3. Reference Architecture for Urban-oriented Digital Twin Systems

1. Reference Business Architecture for urban-oriented DT lifecycle instantiation

This section presents and analyses a generic business process enabling the exploitation of DTs for City Logistics. First, as provided in deliverable D2.1 [2], we realized workshops with relevant city logistics stakeholders and defined relevant value cases and real-world scenarios (description of the context, together with the actors, territories and companies involved, the main problematics they are facing and the targeted goals). Then, we identified the common future workflow that is shared between all the value case scenarios, and that is generic enough to be used by any other city along with their specific simulation models. This is to support experimentation and decision-making and create value for all different stakeholders.

The generic business process is depicted in Figure 3. The first pool (Information Process, *PI*) manages the contextual information from the physical twins, and keeps the data updated during the whole lifecycle of the entire process. In addition, it enables to push updated contextual data to the linked physical twins, in case a stakeholder needs to update the state of the physical twin. The second pool (Execution Process, *PE*) describes the workflow that manages the user tasks and model configuration. These tasks enable stakeholders to construct simulation scenarios, with a possibility to choose and configure simulation model properties linked to it. The specific task *Monitoring* initializes the data monitoring (if applicable in the modelled scenario) to check the received data from physical twins and restart the simulation when a certain configured threshold is reached. The third and last pool (Decision process, *PD*) describes the workflow managed by a decision system. It receives the outcomes of the simulation as a set of successful scenarios. Then the decision system selects the best strategies among produced scenarios. Finally, the stakeholder that has launched the simulation scenario is notified when a generation of a detailed evaluation report is produced.

This to compare the produced KPIs (from simulation) with the ones calculated based on a refreshed contextual data of physical twins. It is important to mention the bidirectional communication between the digital twin and its physical counterpart that is materialized through the link between the outcome of the best scenario and the PD task that allows to publish the updates to the physical counterparts in the real world.

As depicted in this generic business process, there is a need to manage and integrate several kinds of simulation models in a given scenario. Consequently, a model library with a flexible approach is needed to categorize the urban freight system models, describe them, and most importantly, provide a means to handle all the transformations needed to transform the outputs of a model into another model.

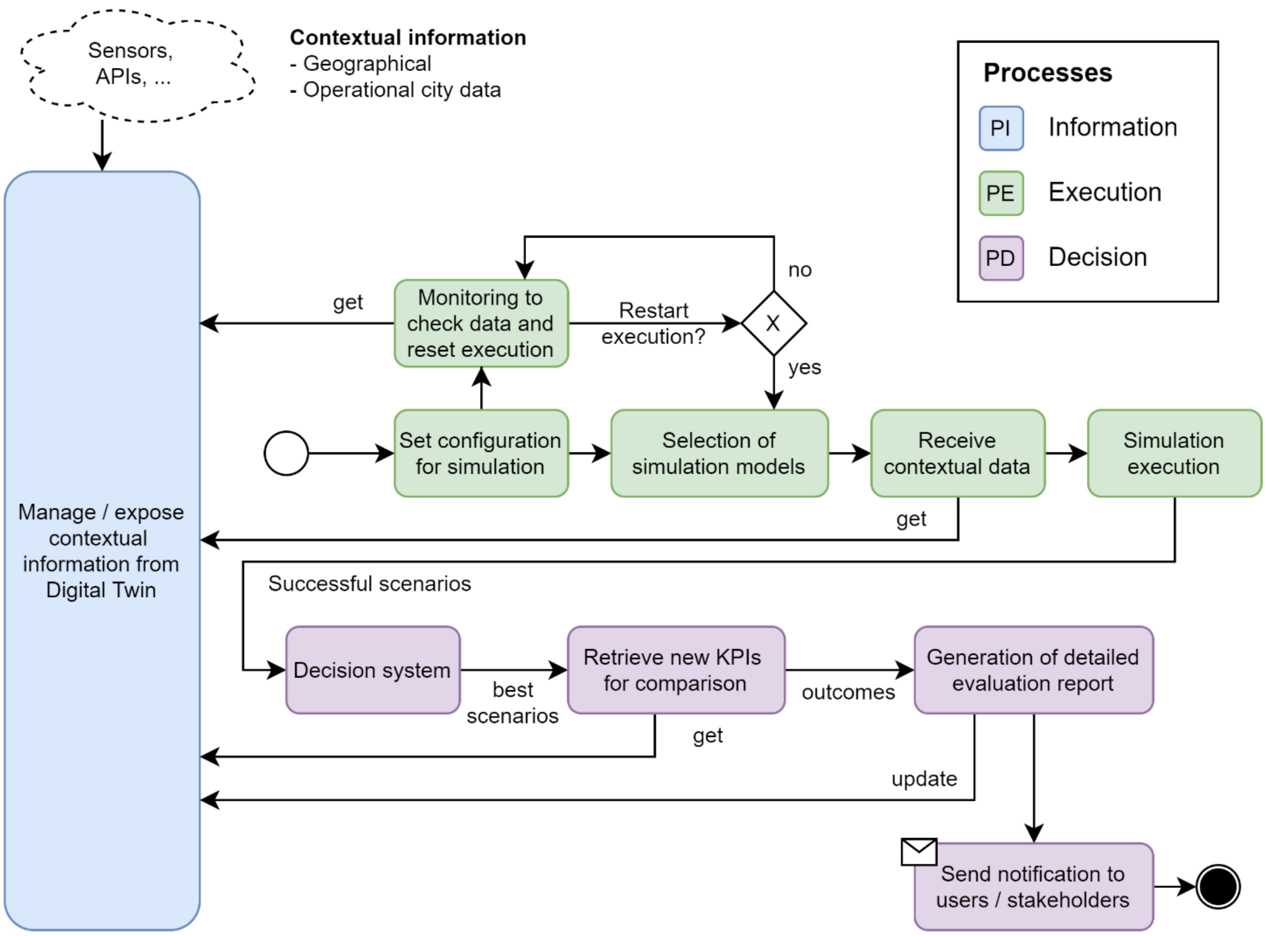

Figure 3 Generic Business Process for urban-oriented DT lifecycle instantiation

2. Reference Architecture for Urban-oriented Digital Twin Systems

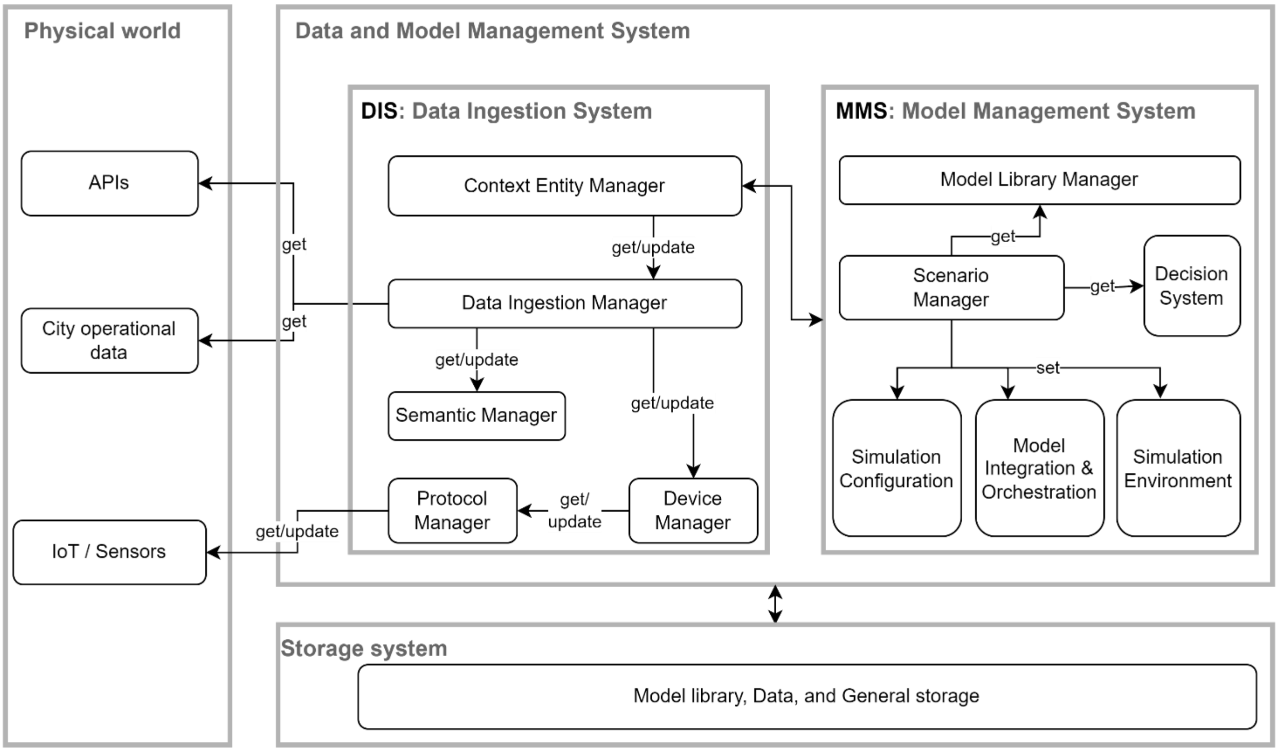

Figure 4 Reference Architecture for Urban-oriented Digital Twin Systems

Figure 4 presents the DT reference framework for urban logistics, which is divided into three main layers. The left layer *Physical World* represents the sensors and external entities such as *Sensors and IoT Entities*, *City Operational Data* and exposed *APIs*. The middle layer *Data/Model Management System* consists of two main elements: the DIS *Data Ingestion System* and the MMS *Model Management System*.

The DIS provides a solution to the need depicted by the *PI* task of the generic business process (see Figure 4 Reference Architecture for Urban-oriented Digital Twin Systems) that manages the contextual information from the physical twins and exposes the twinned data through web APIs to the other framework's internal components. It is intended to ingest the contextual entities (through the *Data Ingestion Manager* and maintain the digital twins updated, managing bi-directional connections with the associated devices that are exposing the contextual data via the *Device* and *Protocol Manager* components. This allows the management of the communication protocols to the physical counterpart and the exposition of this contextual data to the *MMS*. Due to the heterogeneity of the possible physical twins (cameras, sensors, open data, ...), several protocols (HTTP, MQTT or AMQP managed through an IoT platform or an IoT agent[2]) and data format (and the storage by extension) might be involved at this level.

Since a digital twin needs a detailed model of its environment and its physical counterpart, DIS provides the ability to enhance the DT framework and related artefacts with descriptions that are based on semantic annotations or standard ontologies by means of the *Semantic Manager* component. See [23] for an urban logistics ontology. This enables the use of business-oriented specific semantic models to provide a detailed and integrated view of all relevant artefacts, thanks to the use of a unified urban freight ontology that might be used by any model in a simulation scenario.

The MMS brings a response to the need illustrated by the *PE* and *PD* tasks (see Figure 3 Generic Business Process for urban-oriented DT lifecycle instantiation) to integrate multiple types of simulation models in a particular scenario. The MMS consists of several components, including the *Model Library Manager*, which provides an outline of all of the available models (as elaborated in D2.2), as well as their specifications and the connections between the models, and the key performance indicators (KPIs) that are useful for decision making. Please note that all data provided to MMS components during the configuration or simulation phase comes from the *Context Entity Manager* which exposes all physical counterpart's data to MMS components via web APIs.

The *Scenario Manager* allows the registration of what-if scenarios and their related models and configuration (model parameters, contextual data and expected KPIs). The *Model Integration and Orchestration* aims to orchestrate the models by connecting the datasets and the outputs from one model to another based on the what-if scenario configuration. The *Simulation Configuration* component is intended to manage the setup of the simulation environment in terms of technical aspects (parameterization of a virtual machine, technical libraries, etc.) As for the *Simulation Environment* component, it is a virtual execution environment generated on the basis of the simulation configuration. This component uses containerization and virtualization to create dynamically an execution environment for every scenario in order to execute the simulation and to collect the outputs of the scenario.

Digital twins offer real-time communication among assets and systems. The challenge is to know how the data is then processed to provide true added value. In this context, the value added is provided by the KPIs that are tailored precisely to the intended application. For this purpose, the final component of the MMS is the *Decision System* which aims to decide what results of the performed what-if scenario are most likely to be achieved. It considers the key performance indicators of a defined scenario and then recommends the required interventions in the physical world via the DIS to reach the predicted outcome. This decision

[2] *An IoT agent is a component that allows a range of devices to feed their data to be handled from a context broker by using their own native protocols*

system constructs its knowledge using different inputs such as model outputs, scenario parameters, model parameters, contextual data and predicted outcomes.

The remaining component is the *Storage system* for handling general storage such as what-if scenarios, scenario configuration, models and related data, or an ontology store [24] related to the use of the semantic manager component.

3. Model Library and Meta-model

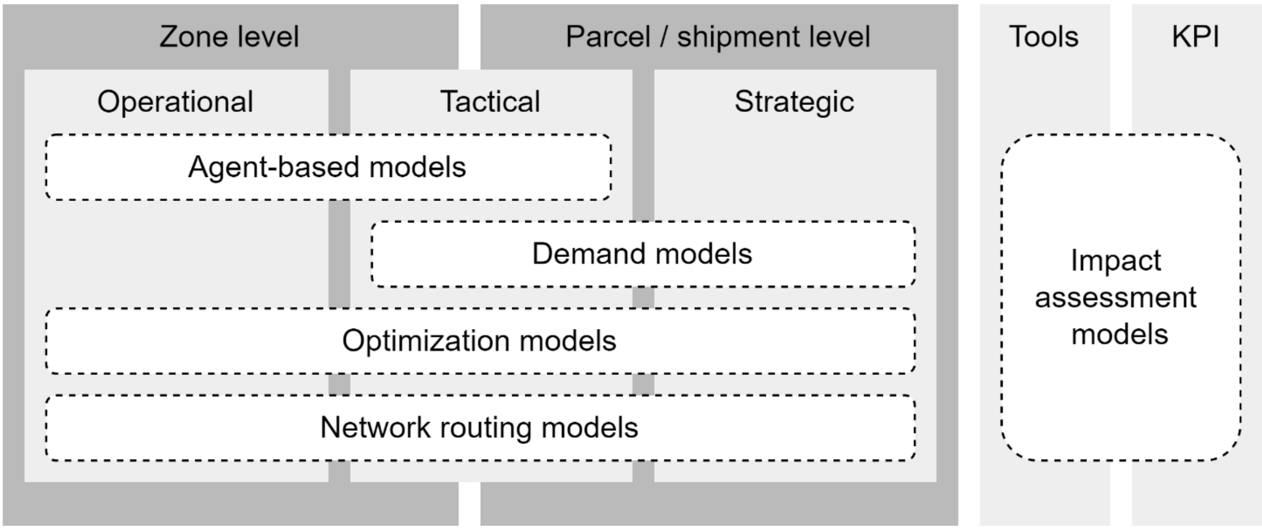

Figure 5 Structure of the Meta-model

The multiple value case scenarios that are considered in the generic business process model needs a flexible approach to be able to accommodate them. For this, a generic model library (MLib) that contains a set of models that can represent different aspects of the urban freight system (see D2.2). The objective of the MLib is to store multiple models that, when combined, allow the user to model different value cases, according to their needs.

In this work, The MLib is conceived as collection of models, where different model types, languages and frameworks can be included. This means that the models included can be theory driven, data driven, AI based or any other type of model. The models in the MLib are connected with connector functions that handle all the transformations needed to transform the outputs of a model into another model. These connectors work in a 1-to-1 fashion, making explicit connections. This has the advantage of increased modularity of the models (the model implementation does not need to change much from their original form to be included), makes data transformation invisible to users and avoids defining a unified urban freight ontology that affects every model.

To assist the user in the model selection, the MLib is organized in the Meta-Model (M2) according to its role in a broader city logistics framework, where an instance of the M2 is a set of models that represents a value case. As highlighted by [25] and [18], the time horizon of the model should match the time horizon of the city activity it wants to replicate. Thus, the M2 is organized by the time scope of logistic decisions: Strategic, Tactical and Operational and by the resolution of the model: Zone or Parcel/Shipment level. Strategic level models are models that deal with long term decisions, such as demand trends and depot locations. Tactical level models define how the fulfilment processes are done. Finally, operational models deal with routing and delivery. This classification allows the user to choose the model that better represents their value case. Additionally, an extra category for models specifically designed to evaluate the outputs of the other models and convert them into KPIs that are useful for decision making.

The current version of the M2 was developed according to the needs of living labs. The models in the MLib were further categorized according to their method, such as optimization, demand models, network models, Agent based models, impact assessment, and routing models. By being aware of the modelling methods used, the user can have more detailed knowledge and control of how the DT is replicating the targeted freight system. Figure 5 shows how the models are categorized.

4. Application and Data Architecture for DDDAS and DSS

To have a self-contained deliverable to facilitate the reading, this section repeats the relevant points on the application and data architecture levels of the internal components of DDDAS and DSS from deliverable D2.1 TR- Solution Architecture [2]:

Each component is described in detail with its internal structure and the interactions between internal or external components. First, an internal detailed description of sub-components and their interactions is realized based on the legend presented in Figure 6. Then, a presentation of the main data entities that is needed on those components is presented depending on the level of details of each component (meta-model, or specific class diagram). Finally, an analysis of each component is realized to define solution building blocks (SBBs) that we are considering at this level of detail for the implementation of each component and its sub-components.

Each SBB will be evaluated in the form of '+++' and '---' to cover the functionality, parameters, and security. The more '+', the better a solution will be covering such aspects, while more '-', the less likely a solution is going to be selected for reuse. In terms of reusability, the tool evaluated are ensured to have an open-source and reuse/extend of source code open enough to allow the exploitation of such tool beyond LEAD.

From this, an extra column has been added showing the final decision of the LEAD consortium, as to whether the solution evaluated is re-used or not for developing the component. The values of this column should be interpreted as follows:

- **Reuse as is**: the SBB is taken as an initial
- **Partial**: some of the functionality from the SBB is taken as a basis, but the development of the component needs further manipulation
- **Research**: the SBB is not known enough and, thus, more research is needed on this SBB to make a final decision
- **Buy**: the SBB, or the functionality of the SBB, is bought into the project in the form of re-use or minimal development on top of it
- **Don't use**: the SBB is disregarded for the development of the component

The listed SBB solutions are collected with the objective to maximize the overlap with components' functionalities. For the mapping between the components and the SBBs, the following architecture principles have been followed:

- Whenever possible, each SBB should cover ALL the targeted component functionalities
- Whenever possible, each SBB should present reliable non-functional properties to be implemented within the technical architecture
- Whenever possible, each SBB should offer interoperability enablers to get access to all its functionalities
- Other principles that could be targeted could be: reuse, coherence, plug and play, etc.

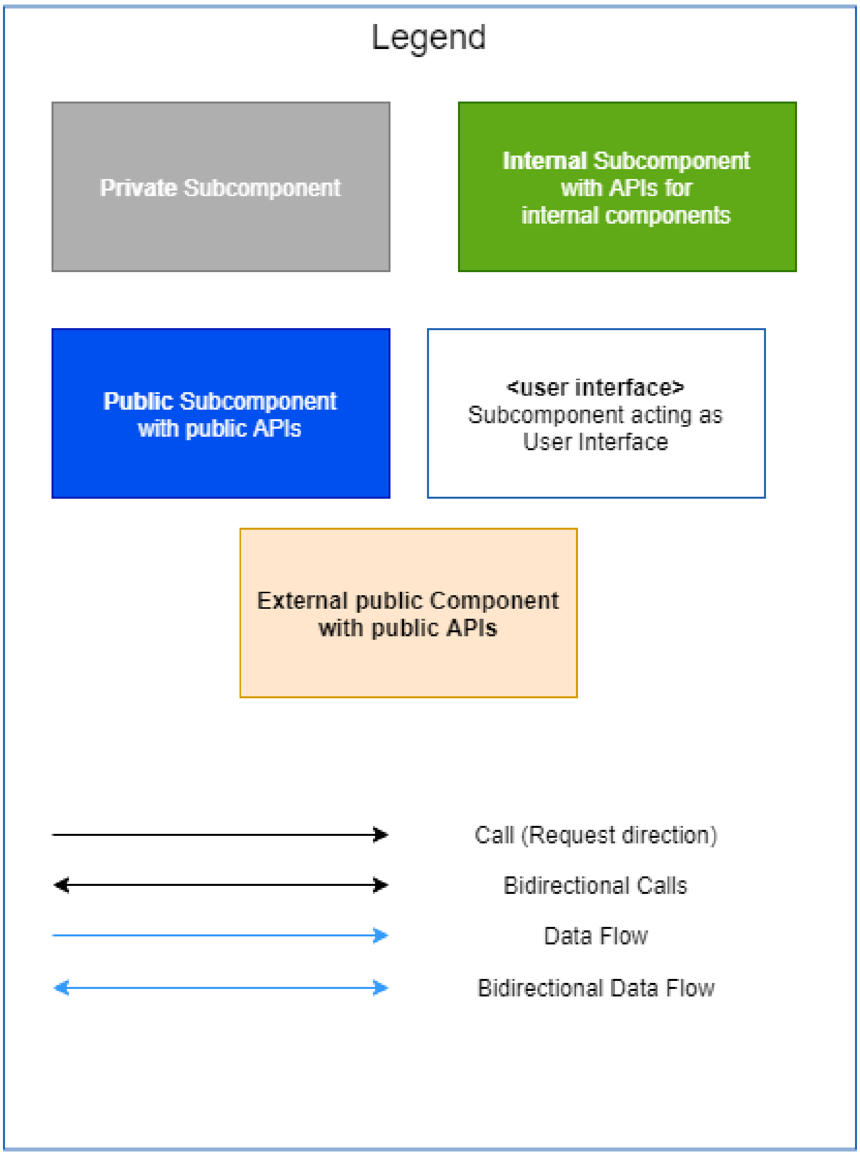

Figure 6 Legend of the LEAD Components' Architecture Diagrams

1. Dynamic Data-Driven Application System (DDDAS) Component

I. Component Interactions description diagram

The following diagram presents the main interactions of the DDDAS [5] (see also full view in deliverable [2] D2.1 - Annex B)

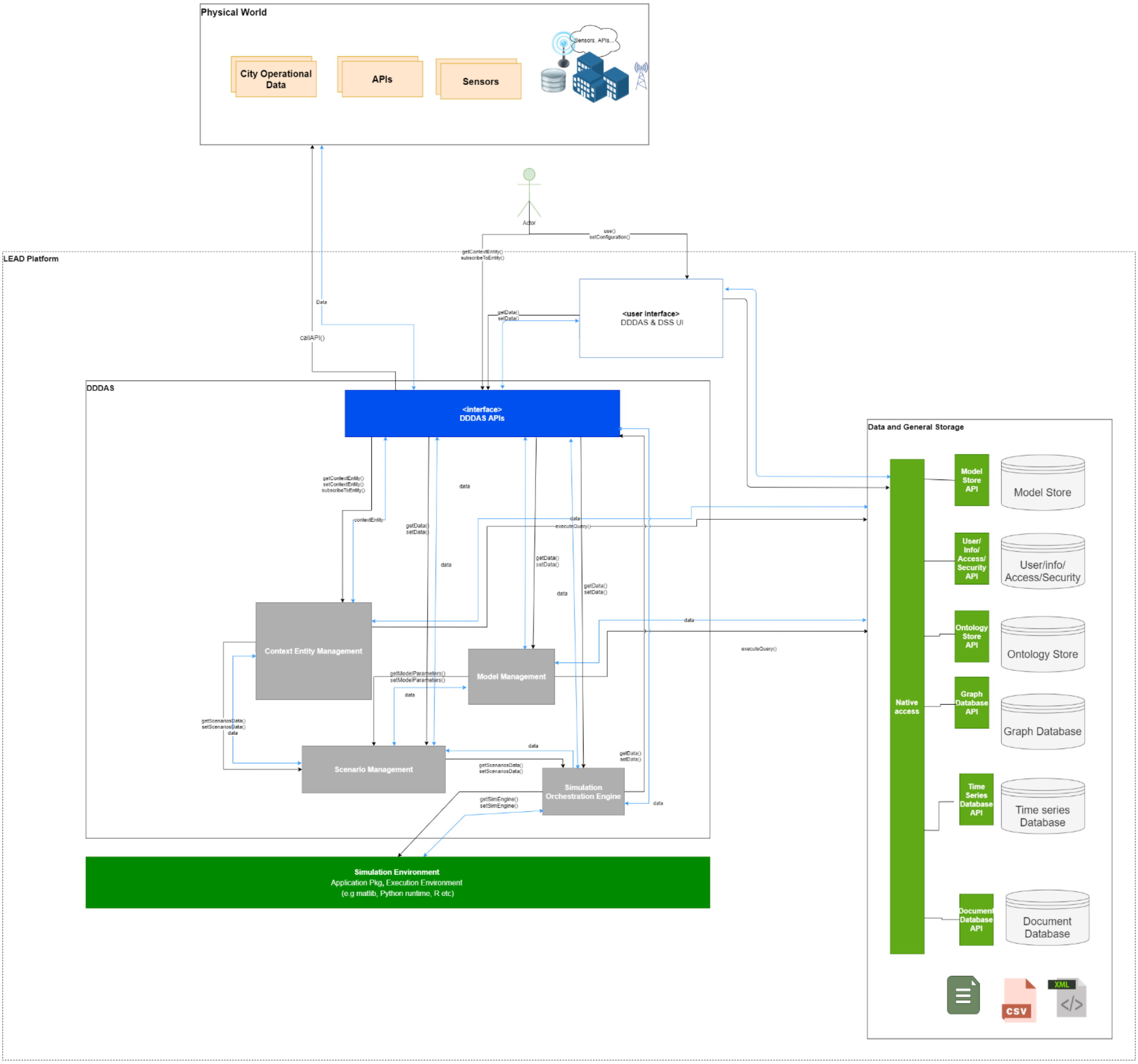

Figure 7 DDDAS Interaction Diagram – full view in deliverable D2.1 - Annex B [2]

The DDDAS component consists of several sub-components, which are described in this sub-section. The connections or communications between sub-components will be performed through direct class calls. And the communications between internal sub-components (in grey colour) and the other components (green for internal or yellow colour for the external) is realized through the exposed Restful APIs.

As depicted in Figure 7, the DDDAS is composed of the following sub-components:

- **Context entity management**: Manages the context entity registration, updates and data retrieval.

- **Model Management**: Manages the models from the model library

- **Scenario Management**: Manages the registration of scenarios, their linked models and configuration of models

- **Simulation Orchestration Engine Management**: Manages the configuration of simulation and the orchestration of the application packages

- **DDDAS APIs:** Exposes the APIs to get access to the functionalities of the sub-components described earlier.

- **Simulation Environment:** This component is outside of the DDDAS; however, it is important to include it in this description since it has a strong link to the simulation orchestration engine. This component enables to execute the prepared simulation in the DDDAS and initialize a complete environment (such as virtual machine, libraries like python, R…) to execute the targeted simulation and retrieve the outputs.

II. Component Interactions definition

DDDAS sub-component have several interactions. These interactions are reflected in the form of a table with the following structure (columns):

- **Sub-Component:** This column defines the sub-component or the sub-function
- **Needs/Gives**: This column represents the following type of interactions:
  - **Gives**: The sub-component provides the sub-component defined in column “With” the data described in column “What”
  - **Needs**: The sub-component needs from the sub-component defined in column “With” the data described in column “What”
  - **Needs/Gives**: The “What” is exchanged on both directions

- **What**: This column describes the interaction or the exchanged data
- **With**: This column points at the sub-component(s) that interacts with the sub-component defined in column “Sub-Component”

Table 2 DDDAS Interactions Description

Sub-Component	Needs/Gives	What	With
API	Needs	Sensor/DB data	Real World
API	Gives	Sensor/DB data	Context Entity Management
API	Gives	Sensor/DB data	Orchestration Engine
Context Entity Management	Needs	Sensor/DB data	API
Context Entity Management	Needs	Scenario Thresholds	Scenario Management
Context Entity Management	Gives	Sensor/DB data	Storage
Context Entity Management	Gives	Sensor/DB data	Scenario Management
Model Management	Needs/Gives	Data parameters/Model definitions	Simulation Management /Model Library
Model Management	Needs/Gives	Model definitions/Data parameters	Model Library/ Simulation Management
Scenarios Management	Gives	Scenario Data	Simulation Orchestration Engine
Scenarios Management	Needs	Sensor/DB data	Context Entity Management
Scenarios Management	Needs	Model Information	Model Library
Scenarios Management	Needs	User parameters	API (Dashboard)
Scenarios Management	Needs	Simulation status	Simulation Orchestration Engine
Scenarios Management	Gives	Simulation bundle	Simulation Orchestration Engine
Scenarios Management	Gives	Scenario thresholds	Context Entity Management

Scenarios Management	Gives	Simulation results	DSS

III. Component Classes and Information Exchanged

Figure 8 shows the associated data model for the described interactions in section ●4.1.II. We defined a meta-model for information exchanged with the context entity manager, the model manager and the scenario manager. In case of the context entity manager, the main class is the “Entity” class, that reflects the contextual data entities that are ingested from external APIs (e.g. sensor data). This data entity can have several attributes, augmented with more metadata (if needed), enabling to have a dynamic structure to fit different kind of contextual data entities. Regarding the model or scenario manager, the same concept is applied to fit to different kinds of models and scenarios and let the DDDAS sub-components generic enough to gather all the required data for different contexts.

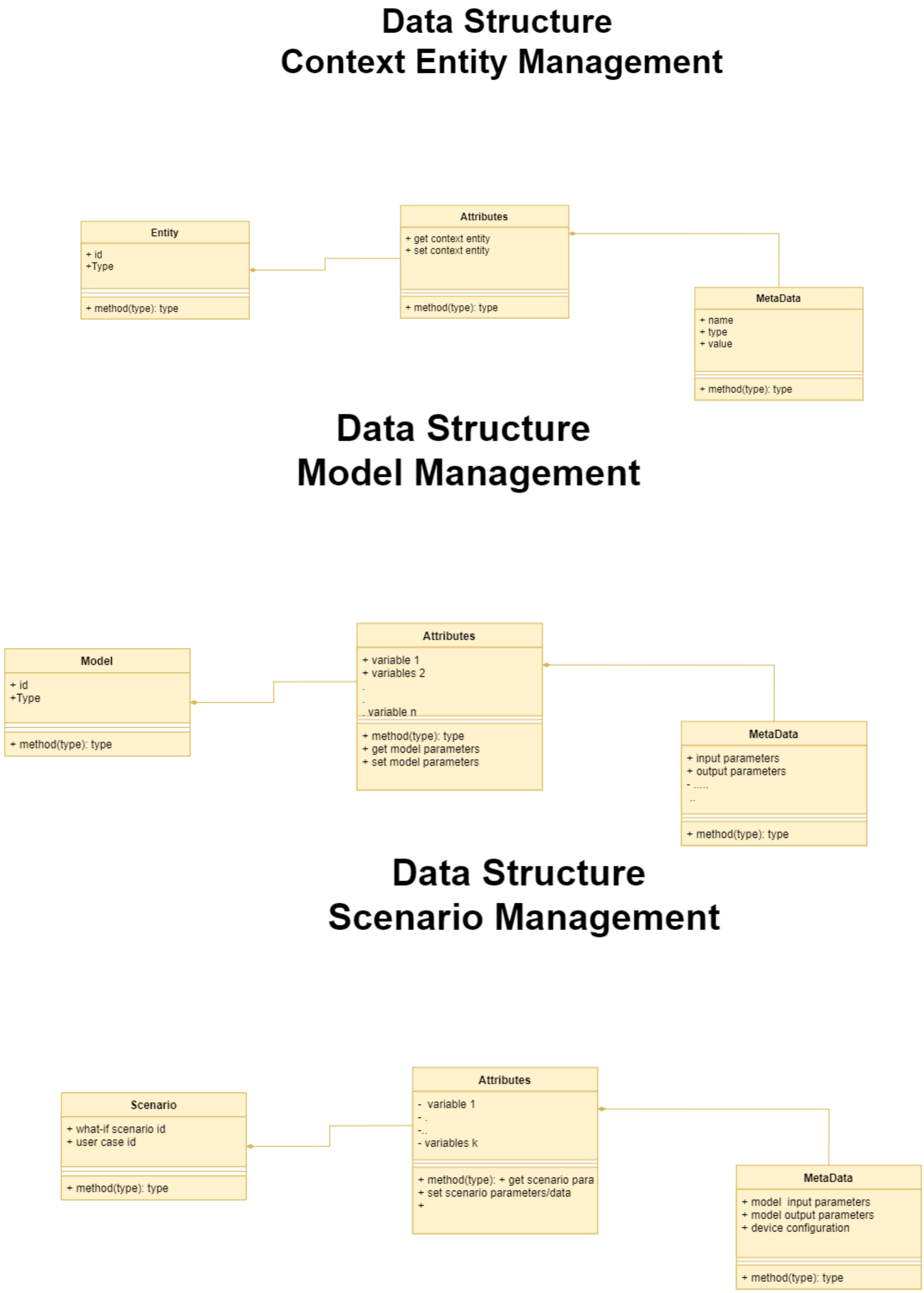

Figure 8 DDDAS Components – Generic Data Structures

IV. Definition of Solution Building Blocks: Reuse vs Make vs Buy

Table 3 below analysis the DDDAS to identify solution building blocks (SBBs) that might be considered for the implementation of each DDDAS component and its sub-components. This evaluation is made based on the methodology described in the introduction of section 4.

Some final decision about the choice of the solution building blocks are presented and justified in D2.4 deliverable for the development of the DDDAS in the LEAD project context.

Table 3 DDDAS Solution Building Blocks Analysis

Component	Solution	Level (Generic ->Specific)	Functionality	Evaluation	Level
Context Entity Manager	Orion Context Broker	Generic	All	+++	Reuse as is
	Eclipse DITTO	Generic	All	+++	Partial
	Kafka	Specific	Data pipelines	++	Partial
DDDAS Dashboard	Angular JS	Generic	All	+++	Reuse as is
Model Manager	Python	Generic	All	++	Partial
Scenarios Management	Python	Generic	Programming Logic	++	Partial
Simulation Orchestration Engine	SLRUM	Specific	Workflow Orchestration	+++	Reuse as is
	Drop & Compute	Specific	Workflow Management	++	Partial
	Python	Generic	Programming Logic	++	Partial
	Docker, Kubernetes	Generic	Virtual Environment, Environment Configuration	+++	Reuse as is
	KVM	Specific	Virtual Environment	+++	Reuse as is
	Ansible	Specific	Environment Configuration	+++	Reuse as is

2. Decision Support System (DSS) Component

I. Component Interactions description diagram

The following diagram presents the main interactions of the DSS (see also full view in deliverable [2] D2.1 - Annex B).

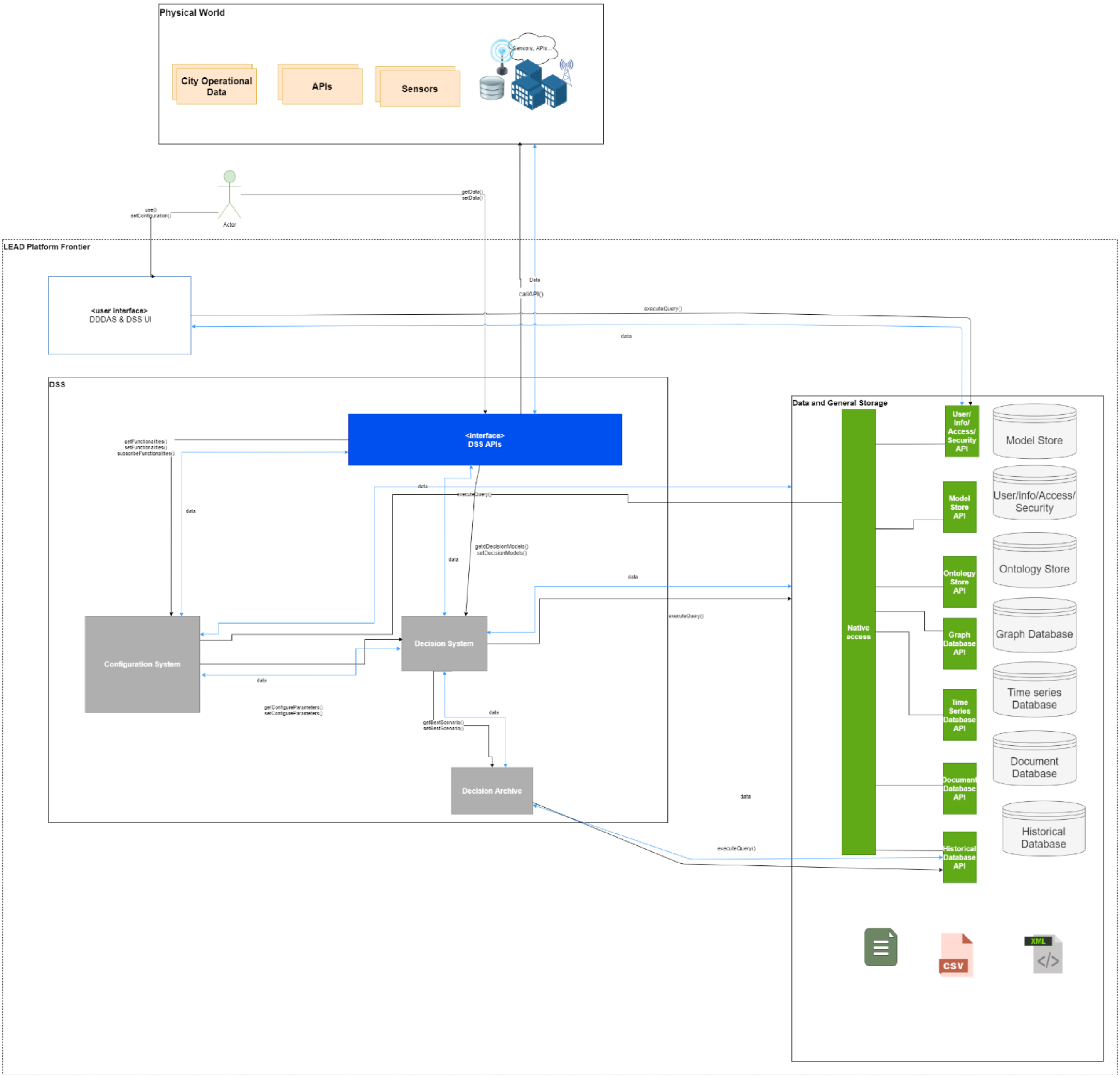

Figure 9 DSS interaction diagram - full view in deliverable [2] D2.1 - Annex B

The DSS component consists of several functions, which are described in Section 4.2.2. The connections or communications between functions will be performed through direct class calls. And the communications between internal functions (in grey colour) and the other components (green for internal or yellow colour for the external) is realized through the exposed Restful APIs.

As depicted in Figure 13, the DSS is composed of the following sub-components:

- **Configuration System**: It configures model, scenarios and devices parameters.
- **Decision System**: Select predictive models to choose the best scenarios among the successful ones that have been provided by DDDAS.
- **Decision Archive**: It is essentially a data base that keeps all the relative information regarding with the best scenario that has been chosen by Decision System.
- **DSS APIs:** Exposes the APIs to get access to the functionalities of the sub-functions described earlier.

II. Component Classes and Information Exchanged

The figure below is indicative of the LEAD platform underlying data structure (entities, attributes and metadata definition). The data structure will be developed in the context of Task T2.4 DDDAS (Looping Control) - Sensing - Data ingestion.

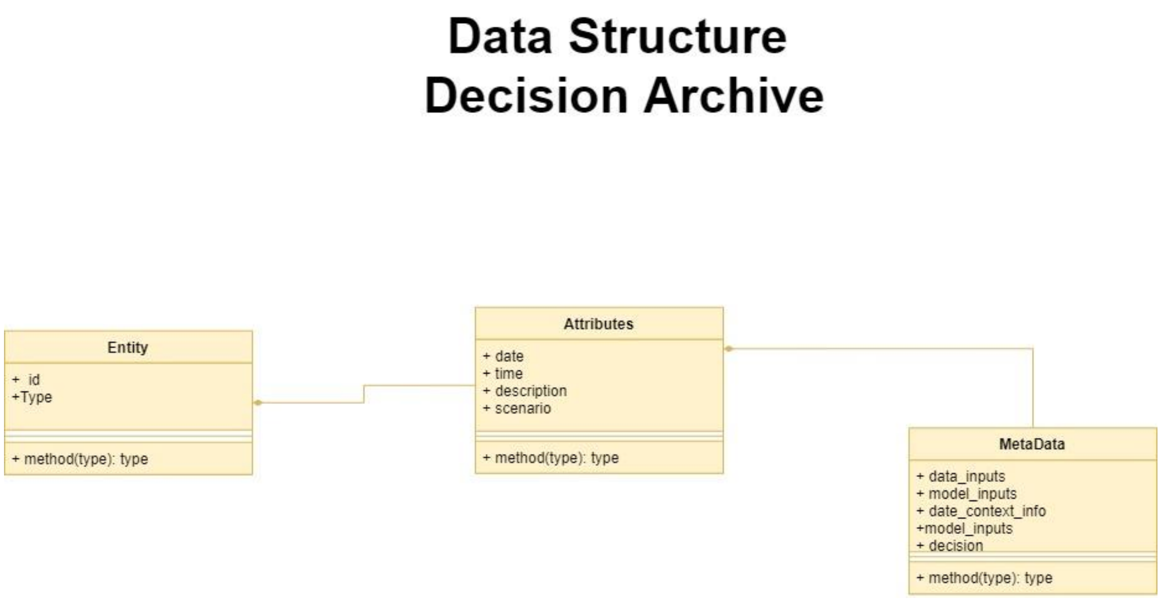

Figure 10 DSS Components Data Structures

4. Technical Insights

The DT Reference Architecture Framework depicted in Figure 4 might be built on top of existing open-source cloud computing technology as suggested in section 4. This section gives additional insights that is refreshed based on some new identified technical tools in the literature and presented in Figure 11.

The technical architecture can apply containerization techniques like Docker[3] and Kubernetes[4] for use case model integration. Regarding the cloud environment, several cloud solutions exist in the market. For example, the hybrid stack could be provided by Cloudify[5], can integrate cloud and local environments. Other existing platforms are OpenStack[6], Cloud Foundry[7], and Apache CloudStack[8]. Classical approaches adopt the idea of deploying full virtual machines (KVM [9]).

For the *Data and Model Management System*, a job scheduler can be employed over Docker/Kubernetes APIs, with a solution to handle data ingestion such as Kafka backbone, along with an open source framework for creating and managing digital twins in the IoT environment such as Eclipse DITTO[10] or Fiware Orion Context Broker[11] for the *Context Entity Manager*. For the *Decision System* (component of the *Model Management System*) one can employ Bayesian inference techniques coded in Python. Regarding the *Simulation Configuration* and *Simulation Environment* components, solutions for IT automation to configure systems, deploy software and orchestrate IT tasks might be considered such as Ansible[12], SLRUM[13] and KVM.

For the connection of the physical world with the data and model management system, there are several front-end with open-source solution for creating customizable dashboards (APIs) such as Grafana[14] technology tools, that is widely used to compose observability dashboards with metrics, logs, and application data.

---

[3] https://www.docker.com
[4] https://kubernetes.io
[5] https://cloudify.co
[6] https://www.openstack.org
[7] https://www.cloudfoundry.org
[8] https://cloudstack.apache.org
[9] https://www.linux-kvm.org
[10] https://www.eclipse.org/ditto
[11] https://fiware-orion.readthedocs.io/en/master/
[12] https://www.ansible.com
[13] https://slurm.schedmd.com/documentation.html
[14] https://grafana.com

For the *Storage System*, different solutions may be considered. For example, because sensor data can be produced very quickly, a traditional database may reach its limits in terms of processing speed and size rather quickly, requiring a different approach to managing this type of data. In this situation, big data solutions are required to achieve the necessary speed and scalability. Time series database such as InfluxDB[15] or document-oriented database like MongoDB[16] may be used for the querying and storage.

Component Type	Implementation Tools
Data & Model Management System	Docker, Kubernetes, Apache Kafka, Apache hadoop, Apache Spark, Apache Flink, Apache Mahout, Eclipse DITTO
Context Entity Manager	Fiware Orion Context Broker, Scorpio Broker, Orion-LD Context Broker, Stellio Context Broker
Decision System	Bayesian Inference Techniques (Python)
Simulation Configuration & Environment	Ansible, SLRUM, KVM
Dashboard & APIs	Grafana, Postman APIs
Storage System	RDBMS, InfluxDB, MongoDB, Hadoop HDFS, HABSE, hive,
Cloud environment	Cloudify, OpenStack, Cloud Foundry, Apache CloudStack, Docker

Figure 11 Additional implementation tools

5. Conclusion

This deliverable has defined the reference architecture and its main generic process that is shared between all LLs. The global architecture defines all reference architecture components and their interconnection. It is based on the main components defined in the description of action but at a more detailed level including their main interactions. Then a specific and detailed view of each component is described, along with its behaviour and functionalities, technical foundations and infrastructure details.

This deliverable is based on the work carried out in Tasks T2.2 (Digital Twin Models Library), T2.3 (Decision Support System and APIs), and T2.4 (DDDAS – Sensing – Data ingestion). Indeed, for T2.2, a detailed definition of its main consumer client in LEAD platform is defined (DDDAS) with a detailed specification about its targeted functionalities, data structure, and interactions with the model library. This helped to better design and consolidate the library of open source models that will be used in the LLs digital twins. Regarding T2.3 and T2.4, a

[15] https://www.influxdata.com
[16] https://www.mongodb.com

detailed specification is provided, along with the UIs, APIs, components interactions, and a development plan that will enable the understanding, development and continuous improvement of the components.

To conclude, the following reference architecture has been accepted for publication in the journal of Computers, Environment and Urban Systems[17], an interdisciplinary journal publishing cutting-edge and innovative computer-based research on urban systems and systems of cities. The authors who contributed to this paper are: Abdelhadi Belfadel; Sebastian Hörl; Rodrigo Javier Tapia; Dimitra Politaki; Ibad Kureshi; Lorant Tavasszy, and Jakob Puchinger.

6. References

[1] The Open Group, TOGAF® Version 9.1. A Pocket Guide. The Open Group. December 2011.

[2] LEAD2.1, "D2.1: Technical Requirements - Solution Architecture".

[3] LEAD2.2, "D2.2: Digital Twin models library".

[4] LEAD2.4, "D2.4: Decision Support System and APIs".

[5] LEAD2.6, "D2.6: LEAD DDDAS System".

[6] "ISO/IEC/IEEE 42010 Website," [Online]. Available: http://www.iso-architecture.org/ieee-1471/cm/. [Accessed 12 05 2023].

[7] ISO42010, "ISO/IEC/IEEE 42010:2011 Systems and software engineering — Architecture description," ISO, 2022.

[8] M. Grieves, "Origins of the digital twin concept," *Florida Institute of Technology,* vol. 8, 2016.

[9] C. Zhuang, J. Liu and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *The International Journal of Advanced Manufacturing Technology,* vol. 96, p. 1149–1163, 2018.

[17] https://www.sciencedirect.com/journal/computers-environment-and-urban-systems

[10] H. Haße, B. Li, N. Weißenberg, J. Cirullies and B. Otto, “Digital twin for real-time data processing in logistics,” in *Artificial Intelligence and Digital Transformation in Supply Chain Management: Innovative Approaches for Supply Chains. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 27*, pp. 4–28, 2019.

[11] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne and L. Wang, “Modeling, simulation, information technology & processing roadmap,” *National Aeronautics and Space Administration,* vol. 32, p. 1–38, 2012.

[12] ISO, "ISO 23247-1:2021(en) Automation systems and integration — Digital twin framework for manufacturing — Part 1: Overview and general principles," Geneva, 2021.

[13] V. Lopez and A. Akundi, “A conceptual model-based systems engineering (mbse) approach to develop digital twins,” in *2022 ieee international systems conference (syscon)*, 2022.

[14] J. Bickford, D. L. Van Bossuyt, P. Beery and A. Pollman, “Operationalizing digital twins through model-based systems engineering methods,” *Systems Engineering,* vol. 23, p. 724–750, 2020.

[15] J. Hatakeyama, D. Seal, D. Farr and S. Haase, “Systems engineering “V” in a model-based engineering environment: Is it still relevant?,” in *2018 AIAA SPACE and Astronautics Forum and Exposition*, 2018.

[16] Y. Qamsane, J. Moyne, M. Toothman, I. Kovalenko, E. C. Balta, J. Faris, D. M. Tilbury and K. Barton, “A methodology to develop and implement digital twin solutions for manufacturing systems,” *IEEE Access,* vol. 9, p. 44247–44265, 2021.

[17] E. Marcucci, V. Gatta, M. Le Pira, L. Hansson and S. Bråthen, “Digital Twins: A Critical Discussion on Their Potential for Supporting Policy-Making and Planning in Urban Logistics,” *Sustainability,* vol. 12, p. 10623, 2020.

[18] M. Batty, "Digital twins," *Environment and Planning B: Urban Analytics and City Science,* vol. 45, p. 817–820, September 2018.

[19] M. J. Kaur, V. P. Mishra and P. Maheshwari, “The convergence of digital twin, IoT, and machine learning: transforming data into action,” in *Digital twin technologies and smart cities*, Springer, 2020, p. 3–17.

[20] W. Kritzinger, M. Karner, G. Traar, J. Henjes and W. Sihn, “Digital twin in manufacturing: a categorical literature review and classification. IFAC-PapersOnLine

51 (11), 1016–1022 (2018)," in *16th IFAC Symposium on Information Control Problems in Manufacturing INCOM*, 2018.

[21] S. M. E. Sepasgozar, "Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment," *Buildings,* vol. 11, p. 151, 2021.

[22] [Online]. Available: http://www.transmodel-cen.eu/.

[23] N. Anand, M. Yang, J. R. van Duin and L. Tavasszy, "GenCLOn: An ontology for city logistics," *Expert Systems with Applications,* vol. 39, 2012.

[24] B. DuCharme, Learning SPARQL: querying and updating with SPARQL 1.1, " O'Reilly Media, Inc.", 2013.

[25] C. Wildfire, *How can we spearhead city-scale digital twins?,* 2018.

[26] A. 3. Specification, "Application Usage Viewpoint," [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/apdxc.html. [Accessed 10 02 2021].

[27] T. O. Group, "Implementation and Deployment Viewpoint".

[28] UN/CEFACT, "Multi Modal Transport Reference Data Model (UN/CEFACT SHIP/MMT-RDM)".

[29] UN/CEFACT, "UN/CEFACT Buy-Ship-Pay Reference Data Model BSP-RDM Version 1.0.," 2019.

[30] [Online]. Available: https://developers.google.com/transit/gtfs.

[31] S. Hörl and M. Balac, "Reproducible scenarios for agent-based transport simulation," *Arbeitsberichte Verkehrs Raumplan,* vol. 1499, 2020.

[32] Y. Xiao and M. Watson, "Guidance on Conducting a Systematic Literature Review," *Journal of Planning Education and Research,* vol. 39, pp. 93-112, 2019.

[33] M. Woschank, E. Rauch and H. Zsifkovits, "A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics," *Sustainability,* vol. 12, p. 3760, 2020.

[34] D. Wohlfeld, "Digitaler Zwilling für die Produktion von Übermorgen," *Zeitschrift für wirtschaftlichen Fabrikbetrieb,* vol. 114, p. 65–67, 2019.

[35] L. Wanganoo and A. Patil, “Preparing for the smart cities: IoT enabled last-mile delivery,” in *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, 2020.

[36] S. Verlinde, C. Macharis, L. Milan and B. Kin, “Does a mobile depot make urban deliveries faster, more sustainable and more economically viable: results of a pilot test in Brussels,” *Transportation Research Procedia,* vol. 4, p. 361–373, 2014.

[37] TOGAF, The Open Group Architecture Framework TOGAF™ Version 9, Basharat Hussain, 2009.

[38] L. Tavasszy and G. de Jong, Modelling Freight Transport, " Elsevier", 2014.

[39] F. Tao, H. Zhang, A. Liu and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on Industrial Informatics,* vol. 15, p. 2405–2415, 2018.

[40] H. Snyder, “Literature review as a research methodology: An overview and guidelines,” *Journal of Business Research,* vol. 104, pp. 333-339, 2019.

[41] B. N. Silva, M. Khan, C. Jung, J. Seo, D. Muhammad, J. Han, Y. Yoon and K. Han, “Urban planning and smart city decision management empowered by real-time data processing using big data analytics,” *Sensors,* vol. 18, p. 2994, 2018.

[42] G. Schrotter and C. Hürzeler, “The digital twin of the city of Zurich for urban planning,” *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science,* vol. 88, p. 99–112, 2020.

[43] A. Schroeder, A. Beltagui, V. G. Shi, C. Kandemir, O. Omidvar, M. Yang, R. Hughes and R. Wasserbauer, “Digital twin for Advanced Service delivery systems: Opportunities and challenges,” in *EurOMA Conference 2020*, 2020.

[44] T. Sakai, A. Romano Alho, B. K. Bhavathrathan, G. D. Chiara, R. Gopalakrishnan, P. Jing, T. Hyodo, L. Cheah and M. Ben-Akiva, "SimMobility Freight: An agent-based urban freight simulator for evaluating logistics solutions," *Transportation Research Part E: Logistics and Transportation Review,* vol. 141, p. 102017, September 2020.

[45] L. Raes, P. Michiels, T. Adolphi, C. Tampere, T. Dalianis, S. Mcaleer and P. Kogut, “DUET: A Framework for Building Secure and Trusted Digital Twins of Smart Cities,” *IEEE Internet Computing,* 2021.

[46] D. Puiu, S. Bischof, B. Serbanescu, S. Nechifor, J. Parreira and H. Schreiner, “A public transportation journey planner enabled by IoT data analytics,” in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017.

[47] Y. H. Pan, N. Q. Wu, T. Qu, P. Z. Li, K. Zhang and H. F. Guo, “Digital-twin-driven production logistics synchronization system for vehicle routing problems with pick-up and delivery in industrial park,” *International Journal of Computer Integrated Manufacturing,* vol. 34, p. 814–828, 2021.

[48] L. Ntziachristos, D. Gkatzoflias, C. Kouridis and Z. Samaras, "COPERT: A European Road Transport Emission Inventory Model," in *Information Technologies in Environmental Engineering*, I. N. Athanasiadis, A. E. Rizzoli, P. A. Mitkas and J. M. Gómez, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, p. 491–504.

[49] Y. Liu, P. Folz, S. Pan, F. Ramparany, S. Bolle, E. Ballot and T. Coupaye, “Digital Twin-Driven Approach for Smart City Logistics: The Case of Freight Parking Management,” in *IFIP International Conference on Advances in Production Management Systems*, 2021.

[50] D. Lee and S. Lee, “Digital Twin for Supply Chain Coordination in Modular Construction,” *Applied Sciences,* vol. 11, p. 5909, 2021.

[51] A. Langevin and D. Riopel, Logistics Systems: Design and Optimization, " Springer", 2005.

[52] B. Korth, C. Schwede and M. Zajac, “Simulation-ready digital twin for realtime management of logistics systems,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018.

[53] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” vol. 2, January 2007.

[54] M. Joerss, J. Schröder, F. Neuhaus, C. Klink and F. Mann, *Parcel delivery. The future of last mile: McKinsey & Company report,* 2016.

[55] M. Jeske, M. Grüner and F. Weiß, “White Paper: Big Data in der Logistik.[pdf] T-Systems,” *Retrieved from,* 2014.

[56] A. Jena, *Fuseki: serving RDF data over HTTP,* 2014.

[57] A. Horni, K. Nagel and K. W. Axhausen, Eds., The Multi-Agent Transport Simulation MATSim, Ubiquity Press, 2016.

[58] S. Hörl and M. Balac, "Synthetic population and travel demand for Paris and Île-de-France based on open and publicly available data," *Transportation Research Part C: Emerging Technologies,* vol. 130, p. 103291, September 2021.

[59] S. Hörl and J. Puchinger, “From synthetic population to parcel demand: A modeling pipeline and case study for last-mile deliveries in Lyon,” in *Transport Research Arena 2022*, 2022.

[60] J. Hopkins and P. Hawking, “Big Data Analytics and IoT in logistics: a case study,” *The International Journal of Logistics Management,* 2018.

[61] M. Hökkä, P. Kaakinen and T. Pölkki, “A systematic review: non-pharmacological interventions in treating pain in patients with advanced cancer,” *Journal of advanced nursing,* vol. 70, p. 1954–1969, 2014.

[62] J. P. T. Higgins, J. Thomas, J. Chandler, M. Cumpston, T. Li, M. J. Page and V. A. Welch, Cochrane handbook for systematic reviews of interventions, John Wiley & Sons, 2019.

[63] L. F. Herrera-Quintero, K. Banse, J. Vega-Alfonso and A. Venegas-Sanchez, “Smart ITS sensor for the transportation planning using the IoT and Bigdata approaches to produce ITS cloud services,” in *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, pp. 1–7, 2016.

[64] M. Hearn and S. Rix, “Cybersecurity considerations for digital twin implementations,” *IIC J. Innov,* p. 107–113, 2019.

[65] V. Gatta, E. Marcucci, P. Delle Site, M. Le Pira and C. S. Carrocci, “Planning with stakeholders: Analysing alternative off-hour delivery solutions via an interactive multi-criteria approach,” *Research in Transportation Economics,* vol. 73, p. 53–62, 2019.

[66] D. Denyer and D. Tranfield, “Producing a systematic review.,” 2009.

[67] M. de Bok, L. Tavasszy and S. Thoen, "Application of an empirical multi-agent model for urban goods transport to analyze impacts of zero emission zones in The Netherlands," *Transport Policy,* July 2020.

[68] M. de Bok and L. Tavasszy, “An empirical agent-based simulation system for urban goods transport (MASS-GT),” *Procedia Computer Science,* p. 126–133, 2018.

[69] F. Coelho, S. Relvas and A. P. Barbosa-Póvoa, “Simulation-based decision support tool for in-house logistics: the basis for a digital twin,” *Computers & Industrial Engineering,* vol. 153, 2021.

[70] S. Boschert and R. Rosen, “Digital twin—the simulation aspect,” in *Mechatronic futures*, Springer, pp. 59–74, 2016.

[71] A. Booth, A. Sutton and D. Papaioannou, “Systematic approaches to a successful literature review,” 2016.

[72] A. Belfadel, S. Hörl, R. J. Tapia and J. Puchinger, “Towards a Digital Twin Framework for Adaptive Last Mile City Logistics,” in *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2021.

[73] J. Allen, M. Piecyk, M. Piotrowska, F. McLeod, T. Cherrett, K. Ghali, T. Nguyen, T. Bektas, O. Bates, A. Friday, S. Wise and M. Austwick, “Understanding the impact of e-commerce on last-mile light goods vehicle activity in urban areas: The case of London,” *Transportation Research Part D: Transport and Environment,* vol. 61, pp. 325-338, 2018.

[74] ALICE-ETP, “Roadmap to The Physical Internet,” 2020.

[75] N. B. Ali and M. Usman, “Reliability of search in systematic reviews: Towards a quality assessment framework for the automated-search strategy,” *Information and Software Technology,* vol. 99, p. 133–147, 2018.