SENSE: Semantic-based Explanation of Cyber-physical Systems

Deliverable 3.1:
Auditable SENSE Architecture

Authors	:	T. Frühwirth, G. Steindl, T. Schwarzinger, F. Ekaputra
Dissemination Level	:	Public
Due date of deliverable	:	31.12.2023
Actual submission date	:	February 2024
Work Package	:	WP3
Type	:	Report
Version	:	1.0

Abstract

Deliverable D3.1 introduces the auditable SENSE architecture as an implementation of the Generic Digital Twin Architecture (GDTA). While motivated by use cases from the Smart Grid and Smart Building domains, it provides a generic technology platform for implementing explainability in Cyber-physical Systems (CPSs). The architecture integrates features for detecting events, providing explanations, and ensuring auditability, alongside a knowledge-driven conversational interface. The design allows for deployment as a containerized application in a subsequent project phase.

History

Version	Date	Reason	Revised by
1.0	14.02.2024	Initial draft	MS

Author List

Project Partner	Name (Initial)	Contact Information
AEE INTEC	Dagmar Jähnig (DJ)	d.jaehnig@aee.at
AEE INTEC	Christoph Moser (CM)	c.moser@aee.at
TU	Thomas Frühwirth (TF)	thomas.fruehwirth@tuwien.ac.at
SIE	Juliana Kainz (JK)	juliana.kainz@siemens.com
SIE	Daniel Hauer (DH)	daniel.hauer@siemens.com
SIE	Konrad Diwold (KD)	konrad.diwold@siemens.com
SIE	Alfred Einfalt (AE)	alfred.einfalt@siemens.com
SIE	Gerhard Engelbrecht (GE)	gerhard.engelbrecht@siemens.com
SIE	Simon Steyskal (SS)	simon.steyskal@siemens.com
WU	Katrin Schreiberhuber (KS)	katrin.schreiberhuber@wu.ac.at
WU	Marta Sabou (MS)	marta.sabou@wu.ac.at

Executive Summary

The SENSE project aims to explain events occurring in technical systems regarding the area of Smart Grid and Smart Buildings. The goal is to contribute to Austria's sustainability goals by making complex systems that underlie key (and often highly polluting) infrastructures more efficient and user-friendly through explanations of (anomalous) events occurring in those systems. The SENSE system to be developed in this project aims to make complex Cyber-physical Systems (CPSs) more transparent and thereby improve the performance and user acceptance of such systems.

This report contains the description of the auditable SENSE architecture. The architecture builds upon the Generic Digital Twin Architecture (GDTA) as a conceptual framework for implementing services in CPSs. It is focused around three major concepts: *event detection*, *explainability*, and *auditability*. Within the architecture, dedicated software modules for each of these concepts interact to implement explainability in CPSs. They are supported by additional modules for data ingestion, semantic data exchange, and user interaction. Summarized, the auditable SENSE architecture comprises the following modules:

- SENSE Semantic Model: holds semantic concepts required for the SENSE system
- Time-Series Database: holds time-series data values from field devices and other equipment
- Data Ingestion: ingests data from devices and the time-series database into the SENSE system
- Semantic Data and Event Broker: acts as a hub for exchanging sensor values and events between modules
- Simple Event Detection: identifies events based on time-series data
- Complex Event Detection: identifies events based on the combination of simple (and possibly complex) events
- Semantic Event and State Log: stores a log of events as input for the explanation generation service
- Explanation Generation Service: generates explanations for certain system events by combining knowledge about current and past systems states with information from the SENSE semantic model
- Audit Box: stores an immutable log of events produced by the SENSE system
- SENSE Connector: provides a well-defined Application Programming Interface (API) to access information generated within the SENSE architecture
- Knowledge-Driven Conversational Interface: provides an intuitive user interface to interact with users of varying degrees of expertise

Table of Content

List of Figures

List of Tables

1 Introduction

1.1 Purpose and Scope of the Document

This document presents the results of WP3, Task 3.1 of the SENSE project: the auditable SENSE architecture (cf. Figure 1). The architecture provides a platform to host the necessary functionality for semantic-based event explainability (to be developed in WP4) and will be prototypically implemented in WP5. It is capable of integrating a variety of data sources, ranging from static system data such as topology information, to runtime data such as sensor values. The architecture is driven by use cases of the Smart Grid and Smart Building domains, but applicable to Cyber-physical Systems (CPSs) in general.

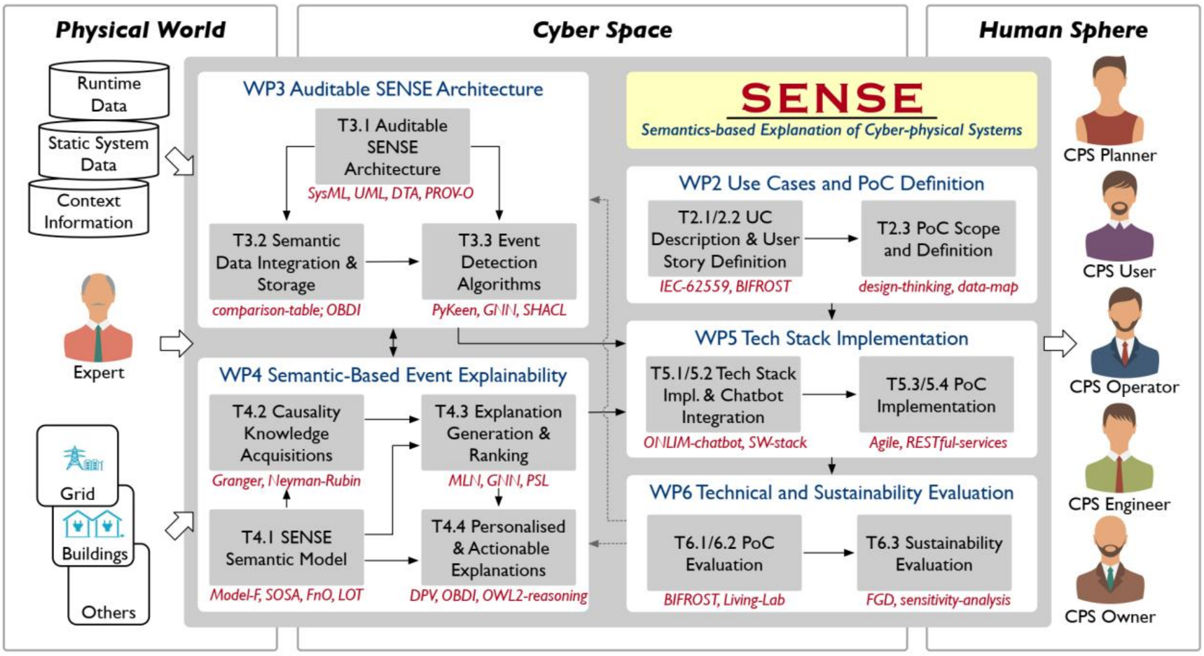

Figure 1 – SENSE conceptual components, their connections, and relevant WPs

The architecture is a direct result of a series of discussions held between all scientific and industrial project partners. Their roles and contributions are summarized in Table 1.

Table 1 Partner's involvement

Project Partner	Name (Initial)	Role/Tasks
WU	Marta Sabou (MS)	Project Coordination
WU	Katrin Schreiberhuber (KS)	Explainability
WU	Fajar Ekaputra (FE)	Explainability, Auditability
WU	Mevludin Memedi (MM)	Use case elicitation
TU Wien	Wolfgang Kastner (WK)	Project Coordination
TU Wien	Gernot Steindl (GS)	Architecture design
TU Wien	Thomas Frühwirth (TF)	Architecture design
TU Wien	Tobias Schwarzinger (TS)	Rule-based event detection
TU Wien	Mohammad Bilal (MB)	Model-based event detection
Siemens	Konrad Diwold (KD)	Supervisory

Siemens	Alfred Einfalt (AE)	Supervisory, Smart Grid use case expert
Siemens	Daniel Hauer (DH)	Smart Grid use case expert
Siemens	Juliana Kainz (JK)	Smart Grid use case expert
Siemens	Rob Poelmans (RP)	Smart Grid use case expert
Siemens	Gerhard Engelbrecht (GE)	Smart Grid use case expert
Siemens	Simon Steyskal (SS)	Smart Grid use case expert
AEE INTEC	Dagmar Jähnig (DJ)	Smart building use case expert
AEE INTEC	Christoph Moser (CM)	Smart building use case expert
MOOSMOAR Energies	Wolfgang Prüggler (WP)	Use case elicitation, economic considerations
Onlim	Ioan Toma (IT)	Knowledge-driven conversational interface
Onlim	Jürgen Umbrich	Knowledge-driven conversational interface

1.2 Structure of the Document

The remainder of this document is structured as follows: Section 2 presents digital twins (DTs) and the Generic Digital Twin Architecture (GDTA) as the underlying methodological concepts of the architecture design. The resulting auditable SENSE architecture and its software modules are then discussed in Section 3. Section 4 summarizes the main outcomes and current activities.

2 Methodology

Digital Twins (DTs) are virtual representations of their physical counterparts, such as physical objects, processes, or systems. As illustrated in Figure 2, they can be classified into Digital Models, Digital Shadows, and Digital Twins. While there is no automatic data flow between the physical and virtual entities for digital models, the digital shadow can access data from the physical entity, and only the DT employs bi-directional communication. DTs enable a conceptual model for a large variety of applications in the realm of CPSs, such as monitoring, simulation, and predictive maintenance. Traditional DTs focus on the physical and virtual entities and their connections.

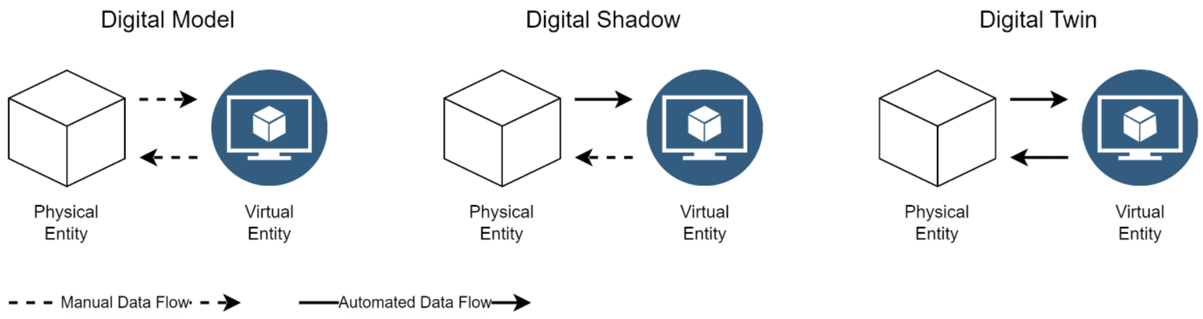

Figure 2 – Digital Model, Digital Shadow, and Digital Twin, adapted from [1]

The three dimensions of classical DTs have since been extended with data and service aspects, resulting in the Five-Dimensional Digital Twin (5D-DT) [2] illustrated in Figure 3. The physical entity comprises a range of subsystems designed to execute particular functions, equipped with diverse sensors that gather operational states and parameters. The objective of the virtual entity is to replicate the physical entity with utmost accuracy by integrating various model types, including geometric, physical, behavioral, and rule-based models. The service model provides services to both the physical entity and the virtual entity, enhancing the operational efficiency of the physical entity and maintaining the virtual entity's precision through real-time calibration of its parameters. The data model is divided into five segments: data derived from the physical entity, data generated by the virtual entity, service-related data, domain expertise, and integrating these data sources. Lastly, the connection model outlines the interactions among the DT's components.

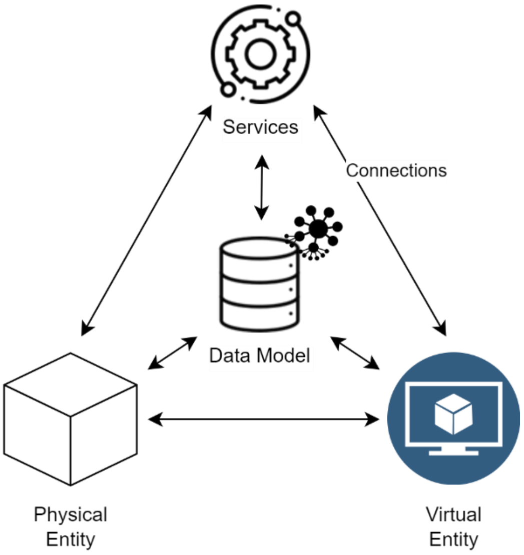

Figure 3 - 5D-DT concept, adapted from [3]

The SENSE architecture builds upon the Generic Digital Twin Architecture (GDTA) [4] concept, which refines the 5D-DT model and targets the applications of DTs during the operational state of an asset. It is aligned with the layers of the Reference Architectural Model Industrie 4.0 (RAMI 4.0) (cf. Figure 4): asset, integration, communication, information, functional, and business. The GDTA allows the reuse of data integration methods, existing data models, and the general service infrastructure for a variety of use cases in the CPS domain, enabling the developers to focus on service development. Furthermore, using this approach as a basis for the SENSE architecture ensures reusability of developed services, such as event detection and explanation, in other applications, eventually creating a framework of reusable services.

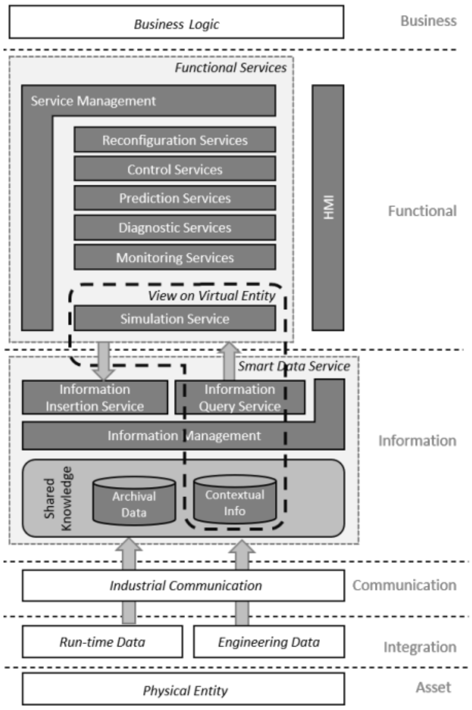

Figure 4 – Generic Digital Twin Architecture aligned with the layers of RAMI 4.0 [4]

3 Auditable SENSE Architecture

The auditable SENSE architecture instantiates the GDTA, as illustrated in Figure 5. It combines numerous services derived from the SENSE use cases and user stories analysis derived from WP2 (described in D2.1 [5]) and aligns them according to the layers of RAMI 4.0/GDTA mentioned above. While inspired by use cases of the Smart Grid and Smart Building domains, the architecture is designed to apply to CPSs in general. Its primary tasks are processing time-series data streams, detecting events on these data streams, generating explanations for relevant events, and presenting them to the user via a conversational interface. In the following, the core services of the architecture are discussed, structured by the corresponding layers from bottom to top.

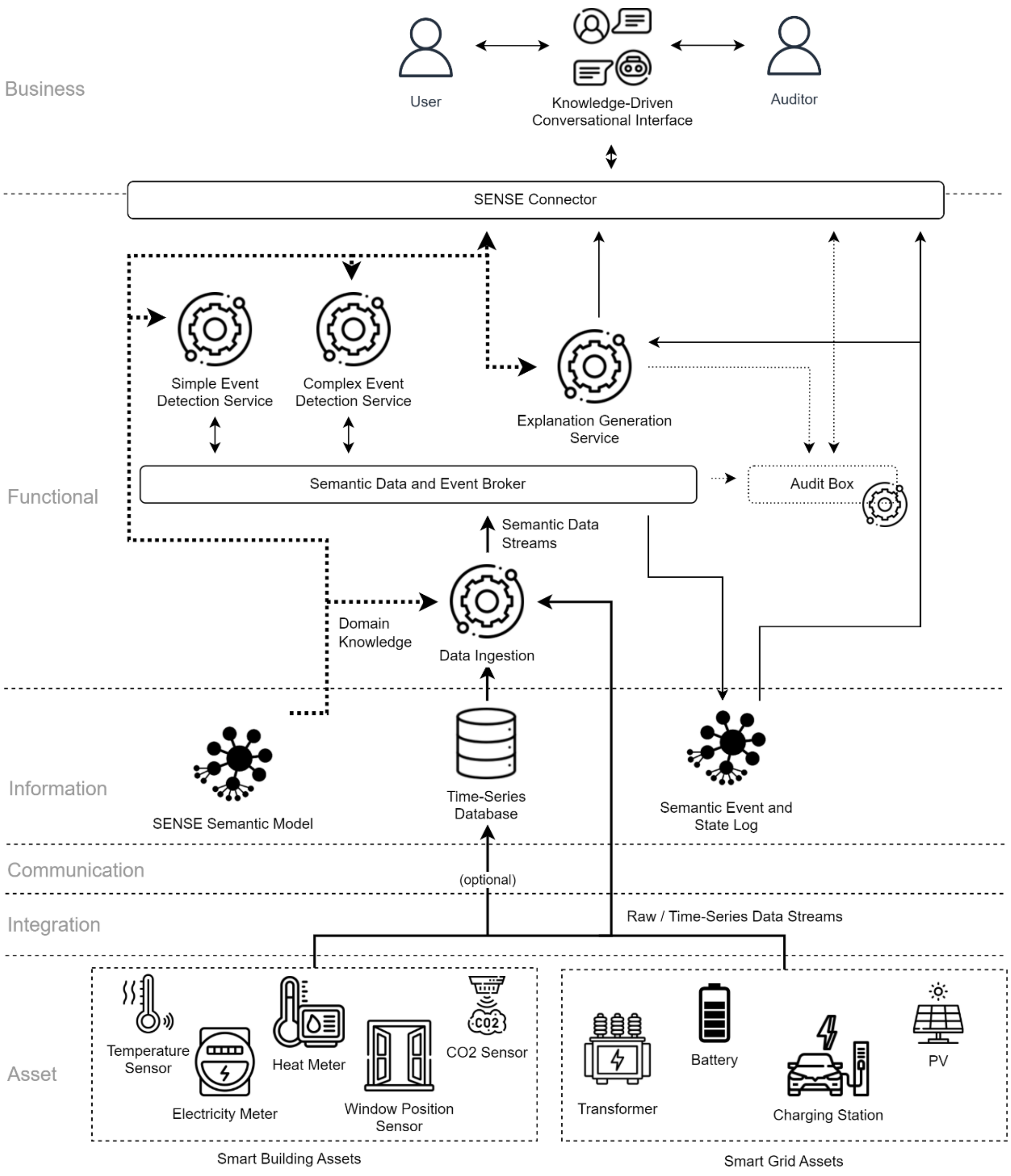

Figure 5 - Auditable SENSE Architecture

3.1 Asset Layer

All data sources for the SENSE system are situated in the asset layer. Data sources primarily include sensors in different equipment, providing their data via domain-specific or proprietary interfaces. However, from the perspective of the SENSE architecture, more sophisticated data sources that provide information possibly relevant for event detection and explanation are also assigned to the asset layer. An example of such an asset would be a weather service providing access to past and current weather information, which influences properties such as temperatures and Photovoltaics (PV) power generation and, thus, might be relevant.

3.2 Integration Layer

Typical data in CPSs are usually provided through an Industrial Internet of Things (IIoT) infrastructure because of their distributed nature. The available communication systems are often domain-specific. Some examples are OPC Unified Architecture (OPC UA) and Message Queuing Telemetry Transport (MQTT) in the industrial domain or Building Automation and Control network (BACnet) in the building sector. Also, more recent concepts, such as the Asset Administration Shell (AAS), can occur as a provider for data and events. In some practical use cases, the sensor data is stored in a special time-series database that also provides direct access to the current (last) and historical values.

3.3 Communication Layer

The fact that there are no modules assigned explicitly to the integration and communication layer can be contributed to the focus of the SENSE project on event detection and explanation generation. Existing libraries and tools, e.g., PLC4X, support a variety of the above-mentioned protocols/interfaces and may be instantiated for integrating domain-specific data sources but are not the focus of the SENSE architecture. They are thus not explicitly illustrated and discussed.

3.4 Information Layer

To provide context to data and events as well as to integrate data from a range of data sources, the **SENSE semantic model** is needed. This model describes certain aspects of the domain, such as concept hierarchies and their relations, to model the physical asset of the DT. This forms an ontology (or knowledge graph), which can be used to retrieve information about assets, including available entities, event sources, and mappings for the integration process. The semantic model is used by several components of the SENSE architecture. Various technologies can be used to create such a semantic model. Depending on semantic expressiveness and tooling support, standards such as NGSI-LD, domain-specific solutions like the DTDL, or a more general knowledge modeling language such as the Web Ontology Language (OWL) could be appropriate.

Furthermore, sensor data may either be ingested directly into the system or stored in a **time-series database** as an intermediary. Using a time-series database has two main benefits. Firstly, the time-series database may serve as a standardized interface: for many applications, interfaces to time-series databases are already available, as this is commonly a first step in the digitalization efforts of industries and companies (cf. Industry 4.0 Maturity Index[1]). Most

---

[1] https://www.fogwing.io/industry-4-0/industry-40-maturity-model/

time-series databases offer subscription services, informing their subscribers about any new measurements. Secondly, establishing connectivity via a time-series database has the obvious benefit that the data can be stored and reused for later analysis and other services. On the downside, the time-series database introduces additional latency and additional complexity regarding storage and processing capabilities that the platform needs to provide.

Methods for interlinking semantic and time-series data, as well as the benefits and drawbacks of each method, is currently investigated in detail in Task T3.2.

The **semantic event and state log** is another semantic data store in the information layer. It serves as a temporary datastore for events, and as events typically cause a state change, state information is also captured here. The semantic event and state log establishes as an interface between the purely event-focused analysis of data streams by the simple and complex event detection modules of the functional layer and the explanation generation module that builds upon current and historical state information. The information stored within the semantic event and state log module is interlinked with concepts, usually described in the semantic model of the digital twin. The storage system of the semantic event and state log may be incorporated into the SENSE semantic model, or a dedicated knowledge base may be instantiated.

3.5 Functional Layer

The **data ingestion** module has to perform two main tasks. First, depending on the data/event source, the module has to poll for new incoming data or register if a publish/subscribe communication mechanism is supported. Second, the incoming data and events are provided by their sources in various formats. Thus, the module has to normalize this information and perform mappings into the semantic model. This allows a seamless integration of semantic data for further processing. The interlinking of time-series data streams and their semantics, as well as alternative semantic data integration methods are subject of Task T3.2 and will be discussed in more detail in the upcoming Deliverable D3.2.

The **semantic data and event broker** provides a common interface for other SENSE modules to connect to. Specifically, it abstracts the various data sources that can provide data to the system but also serves as a message bus for events. This approach enables decoupling the various modules within the architecture. Thus, modules can easily be exchanged, and additional functionalities can be added, even during runtime. Data exchanged via the semantic data and event broker is encoded in Resource Description Framework (RDF), thus incorporating all necessary contextual information instead of raw data values.

Two different modules are responsible for detecting events (as investigated in T3.3 of SENSE). The first one is the so-called **simple event detection** module. It is responsible for detecting events within the data provided by the assets. The architecture supports two general approaches to simple event detection. The first one is the **rule-based approach**, using logical rules that are defined externally by a domain expert or, in some cases, could be semi-automatically generated based on the information from the semantic model. Such simple rules could be, for example, a threshold violation or one signal exceeding another. The complexity of the rules that can be defined depends on the expressiveness of the language used for their definition. A typical example is Signal Temporal Logic (STL).

The second approach for event detection is **model-based,** in which a mathematical model describes the dynamic behavior of the physical part. This approach is needed for use cases where simple rules are not sufficient because of the complexity of the problem. The incorporated models can be divided into white-box, grey-box, or black-box models. White-box models are physics-based and represented mainly by partial differential equations. Creating such models is often rather complex and time-consuming, as many parameters are not known precisely. Black-box models are purely data-driven. Machine learning is a common technique to create such models. Enough run-time data is crucial to capture the dynamic of the physical part, and problems can occur if they are applied under conditions that have not been in the training data. Gray-box models are based on physical principles, but specific parameters are identified based on historical run-time data. Thus, they need a higher modeling effort but are usually more accurate in situations not seen in the training data. Generally, every model can be deployed within the simple event detection module as long as an appropriate run-time environment is provided.

Complex events are defined as a combination of simple and/or complex events. An example of a complex event is that a simple event A occurs within 5 minutes after a simple event B, triggering the complex event C. The **complex event detection** module incorporates a complex event detection engine to detect such events. Such engines are designed to process and analyze complex patterns of events to detect patterns. Usually, they incorporate temporal logic for understanding event sequences and causality. The necessity of such complex event processing is very use-case-specific. Because of the decoupling of the modules through the semantic data and event broker, the complex event detection module is optional. It can be enabled if needed without further changes within the service.

As a core element of the SENSE architecture, the **explanation generation service** is responsible for generating explanations either upon user request via the conversational interface of the business layer or triggered by (critical) events identified by the simple or complex event detection module. The explanation generation service builds upon two main data sources: causality knowledge and the sequence of system events or system states, respectively. Causality knowledge is provided via the SENSE semantic model and can either be derived via means of rule-based symbolic Artificial Intelligence (AI) systems or by combining them with sub-symbolic components [6]. It models information about the interdependencies of events and system states. An example could be the explanation of why an Electric Vehicle (EV) is not fully charged when the user returns to his vehicle: the fact that this situation occurred because of a low loading current is probably not satisfying for the user but has to be investigated further. Low loading current may have been caused by low solar radiation/power production, other EVs charging in parallel, depleted battery of charging station’s battery buffer, or a combination thereof. Thus, the path followed within the causality knowledge graph is often a chain or tree of causalities leading to one or several root causes. Generating expressive explanations often requires exploring past system states, thus necessitating access to the sequence of system events or states provided by the semantic event and state log.

Another key aspect of the auditable SENSE architecture is the integration of the **audit box** as an additional service on the functional layer. The audit box connects to the semantic data and event broker and establishes an immutable log of detected events [7] and explanations generated thereof. For this task, it has an internal representation based on W3C PROV-O for provenance data representation and P-Plan for accountability planning. If necessary due to a severe malfunction of the SENSE system, liability issues, doubts about the resulted explanations or other reasons, this information can later be retrieved, allowing an auditor to retrace the exact sequence of event and explanation generations leading to this undesired situation.

As a final module in the functional layer, the **SENSE connector** allows external systems to connect to the SENSE architecture via a well-defined API. It provides access to the to the information required for retrieving events, explanations, and additional context information that might be relevant for the user and other software systems interacting with the SENSE architecture.

3.6 Business Layer

Interaction with the system is enabled by a **knowledge-driven conversational interface** module. It allows the user to interact with the explanation generation service via a natural language (chatbot) interface to explore the chain/tree of causalities until reaching an explanation that is satisfactory to the user. It may also redirect knowledgeable users to a graphical dashboard that provides in-depth information about signal values and events leading to an undesired system state. The conversational interface supports interactions with different user groups, allowing it to adjust the general language and the technical terms to match the expertise assumed for the specific user group but also disclose only the information that should be accessible.

4 Summary and Future Work

This document presented the auditable SENSE architecture as an instantiation of the GDTA concept. The architecture was designed according to the use cases and requirements of SENSE WP2 with a more general applicability to other CPSs beyond smart grids and smart buildings in mind. The architecture combines aspects of event detection, explainability and auditability with a knowledge-driven conversational interface. It can be implemented as a container-based application, which will be a major focus of WP5.

List of Abbreviations

Short	**Description**
AAS	Asset Administration Shell
API	Application Programming Interface
AI	Artificial Intelligence
BACnet	Building Automation and Control network
CPS	Cyber-physical System
DT	Digital Twin
DTDL	Digital Twin Definition Language
EV	Electric Vehicle
5D-DT	Five-Dimensional Digital Twin
GDTA	Generic Digital Twin Architecture
IIoT	Industrial Internet of Things
MQTT	Message Queuing Telemetry Transport
NGSI-LD	Next Generation Service Interface-Linked Data
OPC UA	Open Process Communications Unified Architecture
PV	Photovoltaics
RAMI 4.0	Reference Architectural Model Industrie 4.0
RDF	Resource Description Framework
SENSE	Semantics-based Explanation of Cyber-physical Systems
STL	Signal Temporal Logic
W3C	World Wide Web Consortium

References

[1] W. Kritzinger, M. Karner, G. Traar, J. Henjes and W. & Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline,* vol. 51, no. 11, pp. 1016-1022, 2018.

[2] T. Fei, M. Zhang and A. Y. C. Nee, "Five-dimension digital twin modeling and its key technologies," in *Digital Twin Driven Smart Manufacturing*, 2019, pp. 63-81.

[3] Q. e. a. QI, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems,* vol. 58, pp. 3-21, 2021.

[4] G. Steindl, M. Stagl, L. Kasper, W. Kastner and R. Hofmann, "Generic digital twin architecture for industrial energy systems," *Applied Sciences,* vol. 10(24), 2020.

[5] D. Jähnig, C. Moser, T. Frühwirth, K. Schreiberhuber, J. Kainz, D. Hauer, K. Diwold and M. Sabou, "SENSE Deliverable 2.1: Definition of Use Cases and User Stories," 2023.

[6] K. Schreiberhuber, M. Sabou, F. Ekaputra, P. Knees, P. Aryan, A. Einfalt and R. Mosshammer, "Causality Prediction with Neural-Symbolic Systems: A Case Study in Smart Grids," in *NeSy*, Certosa di Pontigamo, Siena, Italy, 2023.

[7] F. Ekaputra, A. Ekelhart, R. Mayer, T. Miksa, T. Šarčević, S. Tsepelakis and L. Waltersdorfer, "Semantic-enabled architecture for auditable privacy-preserving data analysis," *Semantic Web,* pp. 1-34, 2022.