Digicities

DATA STRUCTURES AND THEIR APPLICATION

James Allan,

Ali Hainoun,

Sebastian Stortecky,

Daniel Horak,

Camille Reynaud

Project End

ERA-Net Smart Energy Systems

This project has received funding in the framework of the joint programming initiative ERA-Net Smart Energy Systems, with support from the European Union's Horizon 2020 research and innovation programme.

Smart Energy Systems ERA-Net

INTERNAL REFERENCE

Deliverable No.:	D 2.1
Deliverable Name:	Final WP peer reviewed report on the data structures and their application.
Lead Participant:	Empa, AIT
Work Package No.:	2
Task No. & Name:	T2.1, T2.2, T2.3, T2.4, T2.5
Document (File):	D2 Data structures and their application_v0.4.docx
Issue (Save) Date:	2026-04-10

DOCUMENT STATUS

Version	Date	Author(s),	Description
0.1	2022-07-22	James Allan, Ali Hainoun, Sebastian Stortecky, Daniel Horak, Camille Reynaud	Initial draft and review
0.2	2025-10-22	James Allan	Integration of drafts and updated content
0.3	2026-03-24	James Allan	Final description of application of use cases, discussion and executive summary
0.4	2026-04-10	Ali Hainoun	Extension and refinmenet of AUC1, 2 under chapter 5

DOCUMENT SENSITIVITY

☒	**Not Sensitive**	Contains only factual or background information; contains no new or additional analysis, recommendations or policy-relevant
☐	**Moderately Sensitive**	Contains some analysis or interpretation of results; contains no recommendations or policy-relevant statements
☐	**Sensitive**	Contains analysis or interpretation of results with policy-relevance and/or recommendations or policy-relevant statements
☐	**Highly Sensitive Confidential**	Contains significant analysis or interpretation of results with major policy-relevance or implications, contains extensive recommendations or policy-relevant statements, and/or contain policy-prescriptive statements. This sensitivity requires SB decision.

TABLE OF CONTENT

About ERA-Net Smart Energy Systems

ERA-Net Smart Energy Systems (ERA-Net SES) is a transnational joint programming platform of 30 national and regional funding partners for initiating co-creation and promoting energy system innovation. The network of owners and managers of national and regional public funding programs along the innovation chain provides a sustainable and service oriented joint programming platform to finance projects in thematic areas like Smart Power Grids, Regional and Local Energy Systems, Heating and Cooling Networks, Digital Energy and Smart Services, etc.

Co-creating with partners that help to understand the needs of relevant stakeholders, we team up with intermediaries to provide an innovation eco-system supporting consortia for research, innovation, technical development, piloting and demonstration activities. These co-operations pave the way towards implementation in real-life environments and market introduction.

Beyond that, ERA-Net SES provides a Knowledge Community, involving key demo projects and experts from all over Europe, to facilitate learning between projects and programs from the local level up to the European level.

www.eranet-smartenergysystems.eu

1 Executive Summary

This deliverable documents the data vocabularies, ontology design, and semantic data structures developed and applied in the Digicities project. The work responds to a well-documented challenge in urban energy data management: the proliferation of domain-specific data models and ontologies that individually serve their intended purposes but collectively create significant interoperability barriers for service developers and energy planners who must work across these domains simultaneously.

A review of existing data vocabularies confirmed that no single ontology adequately spans the physical scales (device to city), temporal resolutions (sub-hourly to decadal), and stakeholder perspectives (utility operations, municipal planning, building management) encountered in urban energy planning. Developers adopting established standards face complex taxonomies, overlapping terminology, maintenance burdens from external updates, and frequent misalignment between predefined structures and specific use case requirements.

The Digicities core ontology addresses these challenges through a service-driven design that prioritises flexibility for individual applications while preserving the ability to map to established standards. The ontology is organised around Components and Attributes, with each modelled separately and then linked by a corresponding set of properties. This design allows temporal characteristics (historic, live, future time series) to be layered onto any attribute category without creating parallel hierarchies. A systematic mapping framework supports equivalence, hierarchical, and similarity relationships with external vocabularies, enabling simultaneous compliance with multiple domain standards without requiring universal adoption of any single one.

The ontology was applied across the project's use cases, where service-specific extensions were defined to accommodate hydropower scheduling, battery storage management, energy community simulation, and regional energy planning. Each application required different component types and attribute configurations, validating the flexible extension mechanism. Service data requirements were formalised as YAML specifications referencing ontology components and attributes, providing a machine-readable contract between the semantic layers and external analytical services.

This deliverable presents the review of existing vocabularies that informed the design (Section 3), the ontology architecture and its key design decisions (Section 4), the application of the ontology across use cases (Section 5), and a discussion of the design trade-offs, limitations, and opportunities for future development (Section 6).

2 Introduction

The purpose of this section is to explore and describe the data vocabularies and platforms for exchanging data for the energy planning of a city and its components.

2.1 Data models, formats and schemas

There are several ways to represent data. A data model is an abstract model that organizes the elements of the data (e.g. UML Diagrams). A data format is a formal encoding that applications can read and write (e.g. JSON, XML). A schema defines the structure of the data, where specific attributes of the data and their types are defined using key-value pairs (e.g. Name:string, Age:integer). Data can be validated against a schema to ensure the data structure is valid.

2.2 Semantic web technologies

Semantic web technologies is the collective term given to a set of standards and technologies, established by the World Wide Web Consortium, to enable a common framework to share data across applications. The objective of semantic web technologies is to achieve interconnectivity data across the internet and to enable automated reasoning of data by machines. The standards at the core of the semantic web are maintained by the World Wide Web Consortium; however, they have found applications outside of the web and have been applied in several sectors to exchange data (REFS NEEDED). They are also a core component of digital twins (REF NEEDED).

2.2.1 Resource Description Framework (RDF)

RDF is a methodology to describe graph data. RDF consists of triple-connected statements that consist of a subject, a predicate and an object. RDF has several serialization formats, some of which are explained below.

2.2.2 JSON-LD

JSON-LD is a linked data encoding based on the JSON data format. It includes several syntax keywords that are specific to JSON-LD. A full list of the keywords can be found in the JSON-LD specification (Sporny et al., 2018).

2.2.3 Web Ontology Language (OWL)

Ontologies describe taxonomies and classification networks. They provide context to data by providing classes of data to describe objects and the relationships between them.

2.3 Domains

A knowledge domain is a term used to describe the information a specific field or activity e.g. engineering, architecture. The scope and boundaries of domains are difficult to define and the boundaries of domains often overlap. Project stakeholders encounter several knowledge domains when working and developing services to work with city data.

2.4 Data vocabularies

Data vocabularies are terms and definitions used to describe concepts, properties and relationships of a knowledge domain. This work refers to data vocabularies as the collective term given to the terms and definitions in data models and ontologies.

2.5 Data scales

The data scale refers to either the physical scale or temporal resolution of the concepts stored in a vocabulary e.g. building, district.

2.6 Level of detail (LoD)

The level of detail is a formal way of describing how much information is contained within a data record. These levels are often define in the data model e.g. CityGML. Defining the required level of detail is important to avoid unnecessary complexity in data.

Data models have a specific purpose and therefore span a range of spatial scales. The scale of the data refers to the scales of the physical components that can be represent

3 REVIEW OF DATA VOCABULARIES

In this section, we review the data vocabularies used to describe the following domains encountered when working with city data:

- Geospatial boundaries
- Building
- Energy and time series
- IoT and real time

An overview of the different data vocabularies in relation to their physical scale and temporal scale are shown in Figure 1.

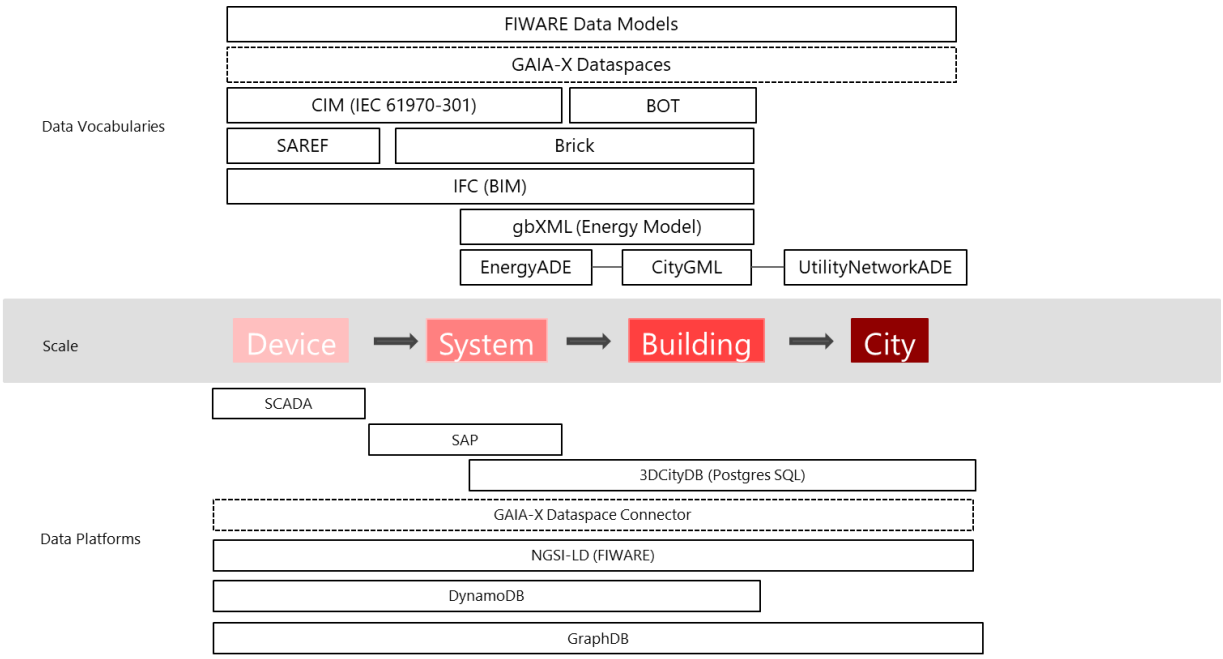

Figure 1: An overview of some of the data vocabularies and platforms that represent components of a city for energy planning

3.1 Geospatial vocabularies

Geospatial data contains geographic or spatial information. Geographic Information System (GIS) is the software system used to work with geospatial data. GIS software enables layer-based analysis of vectors (e.g. points, lines, polygons) and raster data (e.g. satellite images, tiles). Geospatially resolved data is increasingly available and there are a large number of applications to manage and visualise the data. They are an essential component of data platforms for cities. A review of the different technologies for storing and analysing geospatial data found that document databases are the best platform for geospatial data processing; whereas graph databases and key-value databases are limited in both data structures and the processing options (Guo and Onstein, 2020).

3.1.1 GeoNames

Type: Web Ontology Language

Physical Scale: Geospatial names

Temporal Scale: NA

Geonames is a database that contains place names of for all regions across the world, in multiple languages. It is a collaborative effort and contains contributions from many organisations in each country. A comparison of the data providers to GeoNames for Switzerland and Austria is shown in Figure 2. The GeoNames

Ontology is designed to enable the use of geospatial information on the semantic web[1].

Austria

	Name	Description	Website	Licence
1	statistik austria	Statistik Austria CC-BY-3.0: Statistik Austria - data.statistik.gv.at	http://www.statistik.at/	CCBY3.0
2	data_gv_at	Offene Daten Österreich | data.gv.at	http://data.gv.at/	
3	steiermark	Open Government Data Land Steiermark	http://data.steiermark.at/	
4	landsalzburg	SAGIS - Salzburger Geographisches Informationssystem	http://www.salzburg.gv.at/	
5	noegis	Geoinformation-NÖGIS	http://www.noe.gv.at/	
6	land-oberoesterreich	Land Oberösterreich	http://www.land-oberoesterreich.gv.at/	
7	data.ktn.gv.at	Open Data des Landes Kärnten	http://data.ktn.gv.at	
8	bev	Bundesamt für Eich- und Vermessungswesen Österreichisches Adressregister, data of the record date 01.10.2018	http://www.bev.gv.at	
9	stadt_linz	Stadt Linz	https://www.data.gv.at/katalog/dataset/stadt-linz_statistischebezirkelinzab20140101	CCBY4.0

Switzerland

	Name	Description	Website
1	ch-zh-statistics	Statistisches Amt des Kantons Zürich	http://www.statistik.zh.ch
2	raiffeisen	Raiffeisen Schweiz	http://www.raiffeisen.ch
3	bfs	Bundesamt für Statistik - Schweiz	http://www.bfs.admin.ch/
4	swisstopo	Bundesamt für Landestopografie:swisstopo	http://www.swisstopo.admin.ch/
5	cadastre.ch	Amtliches Ortschaftenverzeichnis	http://www.cadastre.ch/internet/cadastre/de/home/products/plz.html
6	migros_retail	Migros Retail Company	http://www.migros.ch/
7	data_stadt_zh	OpenData ZH	https://data.stadt-zuerich.ch/
8	opentransportdata_swiss	Open Data Platform Swiss Public Transport	https://opentransportdata.swiss/
9	geolion	Geographisches Informationssystem GIS-ZH	http://geolion.zh.ch/opendata
10	opendata_swiss	Open Government Data Portal	https://opendata.swiss/
11	swisspost	Open Data Portal of Swiss Post	https://swisspost.opendatasoft.com
12	vgwr	Rohdaten Adressen, csv	https://data.geo.admin.ch/ch.bfs.gebaeude_wohnungs_register/

Figure 2: A comparison of the data sources in Austria and Switzerland that contribute to the GeoNames database.

3.1.2 GeoSPARQL

Type: Web Ontology Language

Physical Scale: Generic geospatial properties

Temporal Scale: NA

GeoSPARQL is a web ontology language for representing and querying geospatial data using semantic web technologies. It is developed and maintained by the Open Geospatial Consortium (OGC). It was created to enable geospatial data interchange with efficient geospatial queries (Battle and Kolas, 2011). GeoSPARQL only defines two subclasses of Spatial Object: Features and Geometries, which can both belong to spatial collections. A summary of the classes within GeoSPARQL are shown in Figure 3.

[1] https://www.geonames.org/ontology/documentation.html

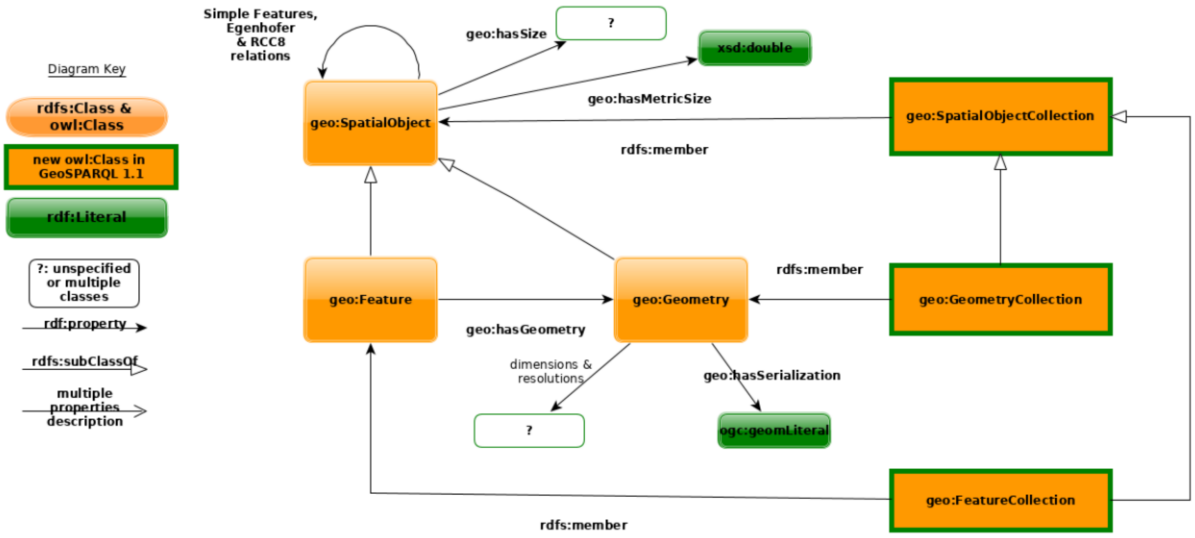

Figure 3: GeoSPARQL 1.1 ontology overview (Car and Homburg, 2022)

An example implementation of GeoSPARQL is shown in Figure 4. Note the reference to the Simple Features Access is set of standards that define common storage and access models of geographic features[2].

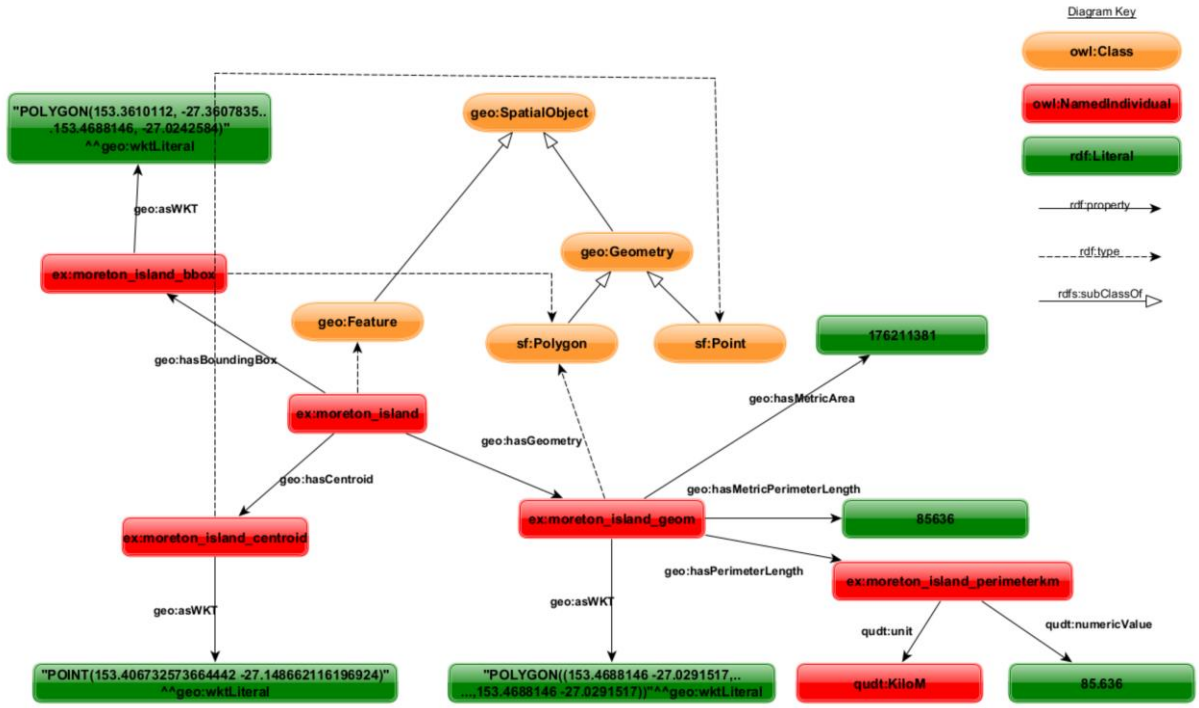

*Figure 4: **Excerpt of the GeoSPARQL 1.1 ontology including one example feature** (Car and Homburg, 2022)*

[2] Simple features RDF XML schema: http://schemas.opengis.net/sf/1.0/ (accessed 21/03/2023)

3.1.3 Formalising spatial scales for urban energy modelling

The spatial scales are mostly administrative constructs that are specific to a region or country. For example, there is no universal definition of a postcode, district or canton. These will mostly vary from country to country. A spatiotemporal ontology for administrative units of Switzerland (SONADUS) was created (Gantner, 2011); however, the ontology became large and complex due to adherence to rules in the upper ontology (Gantner et al., 2013). One of the largest resources of linked data on the internet is DBPedia, which also contains information on the Swiss municipality borders. A study showed that querying Swiss regions with the GeoNames terms and creating manual links to DBpedia, returned acceptable performance (Grütter et al., 2017).

3.1.4 Hexagonal hierarchical geospatial indexing system (H3)

H3 is a hierarchical geospatial indexing system for geographic data. Indexed data can be joined across different datasets and aggregated at different levels of precision. The H3 index is an unsigned 64-bit integer representing any H3 object.

3.2 Building vocabularies

3.2.1 Real Estate Core

Type: Web Ontology Language

Physical Scale: Device and building scale

Temporal Scale: Relationships connect external time series database

The Real Estate Core (REC) Ontology is a Web Ontology Language that was developed to support energy usage analysis/optimisation and presence analysis (Hammar et al., 2019). It is designed to accommodate all of the data requirements of real estate management. The REC is comprised of two base models that can be extracted. The REC has the following modules:

- Metadata
- Core
- Agents
- Building
- Device
- Lease

There are currently efforts taking place to harmonize the metadata standards of Brick and REC to establish clearer semantic boundaries (Wallin, 2022).

3.2.2 Brick Ontology

Type: Web Ontology Language

Physical Scale: Device and building scale

Temporal Scale: Relationships connect external time series database

The objective of the Brick ontology is to capture the concepts and relationships necessary to operate a BMS across a heterogeneous set of buildings (Balaji et al., 2016). The core concepts and relationships of the initial release of the Brick Schema are shown in Figure 6.

The concepts covered by the Brick ontology span the energy systems and building topology. While other building data ontologies either offer a focus on smart and connected appliances or building energy systems, BRICK integrates and formalizes many concepts from both domains, as outlined in https://brickschema.org/

Table 1 Comparison of BRICK to other building data ontologies Source: https://brickschema.org/

Modeling Support	**Brick**	**Project Haystack**	**IFC**	**BOT**	**SAREF**
HVAC Systems	**yes**	**yes**	**yes**	no	no
Lighting Systems	**yes**	partial	**yes**	no	no
Electrical Systems	**yes**	**yes**	**yes**	no	no
Spatial Information	**yes**	no	**yes**	**yes**	no
Sensor Systems	**yes**	**yes**	generic	no	**yes**
Control Relationships	**yes**	no	generic	no	no
Operational Relationships	**yes**	no	generic	no	no
Formal Definitions	**yes**	no	**yes**	**yes**	**yes**

BRICK has already been applied in the creation of a data model for the GG4 building at AIT. The focus here was on maintenance and sensor devices.

3.2.3 GML and CityGML

Type: Conceptual Data Model

Physical Scale: Building, City

Temporal Scale: Support for time series and can be connected to external databases

The Geography Markup Language (GML) is an XML-based language for describing geographical features. GML is an open standard maintained by the OGC. CityGML extend the concepts of GML for the representation and exchange of 3D city models. The CityGML is a conceptual data model used to represent virtual 3D city and landscape models (H. Kolbe et al., 2021). The CityGML conceptual data model is

comprised of modules that represent different elements of a city. The CityGML conceptual model is extendable through application domain extensions (Biljecki et al., 2018).

An OWL ontology for the CityGML 2.0 schema (University of Geneva, 2023) was unable to represent all of the classes for a test city dataset (Charlottenburg-Wilmersdorf district of Berlin); this lead to a proposed extension of the ontology to represent the required classes in the test dataset (Chadzynski et al., 2021). The authors named the ontology "OntoCityGML" and concluded it could act as the schema to serve a semantic twin to the 3DCityDatabase software; however, at the time of writing, the ontology has not published. 3DCityDB is a geo database to store represent and manage 3D city models using a relational database (Kolbe et al., 2013). At the time of writing, 3DCityDB is compatible with the CityGML 2.0 datasets.

The latest version of CityGML, CityGML 3.0, can now be qualified by a relation type identifiable using URI (e.g. using the sameAs relation from OWL), which allows for mapping to RDF triples (Kutzner et al., 2020).

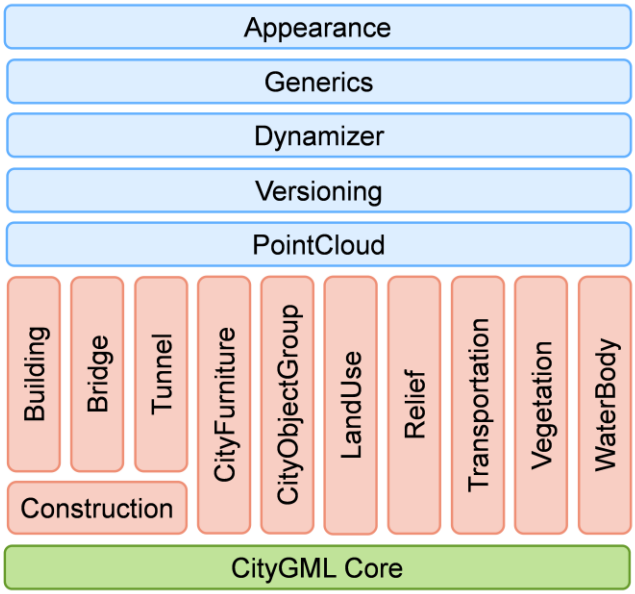

Figure 5: CityGML 3.0 module overview. Source (H. Kolbe et al., 2021)

Applications and technologies work with different modules of the CityGML conceptual model by defining an implementation specification. This contains the results of a set of tests to demonstrate conformance in representing the modules shown in Figure 5. The only mandatory tests for any application or technology is the CityGML core. CityGML 3.0 contains a new Dynamizer module, which enables: data structures to represent time series data, overwriting of static attributes and explicit linking of sensor and observation data. This module aims to help the integration of IoT devices and the time-series data generated by scenario modelling.

3.3 Energy and time series vocabularies

3.3.1 Ashrae 223p

Type: Data Standard

Physical Scale: Building

Temporal Scale:

Ashreae 223p aims to provide a data standard or tagging dictionary that allows interoperability between building data. This explicitly includes other building data schemas like BRICK, Project Haystack, and most importantly BACnet, which are directly involved in the development of the Ashrae 223p dictionary. While the development does not seem to be finished, it is aimed to be adopted as ISO standard.

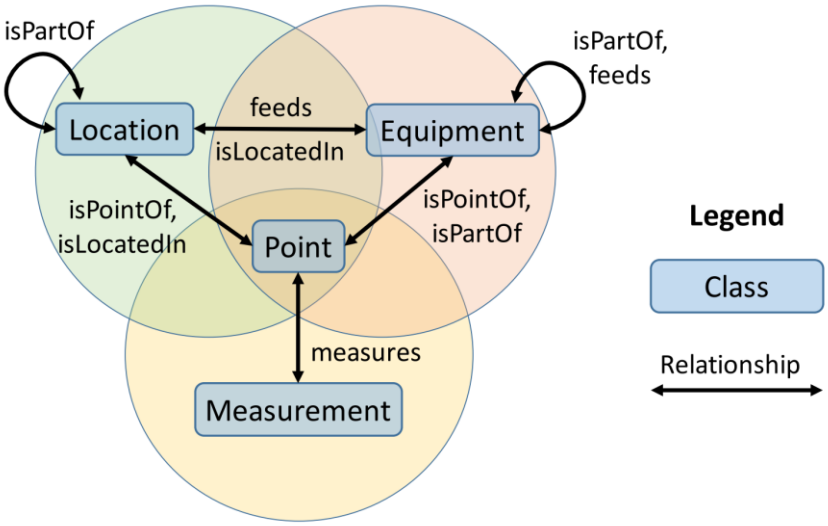

Figure 6: Information concepts in Brick and their relationship to a data point. Source (Balaji et al., 2016)

3.3.2 DTDL

Type: Web Ontology Language

Physical Scale: Device and building scale

Temporal Scale: Relationships connect external time series database

The DTDL is comprised of six metamodel classes that are used to describe the behaviour of all digital twins (Azure, 2022). These metamodel classes are Interface, Telemetry, Property, Command, Relationship, and Component. DTDL is implemented in JSON-LD. The Azure Digital Twin Platform incorporates the DTDL and is the language used in the WillowTwin ontology for buildings ("WillowInc/opendigitaltwins-building," 2026). The adoption of DTDL in WillowTwin is shown in Figure 7.

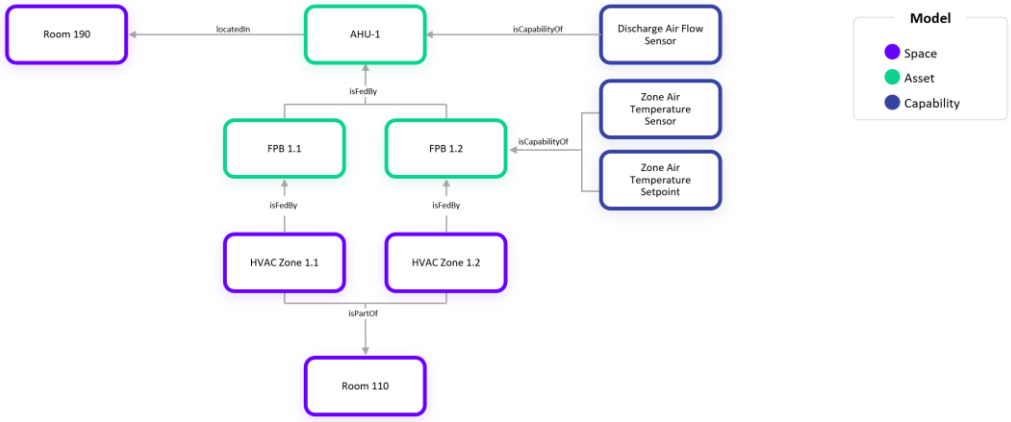

Figure 7: A representation of how classes are linked in the WillowTwin open digital twin definition language for buildings and real estate. Source[3].

4 DIGICITIES ONTOLOGY

The Digicities ontology was developed to accommodate the diverse services and models that require data from the semantic layers. The reviews of domain ontologies (3.5 Review of data vocabularies) recognised that there are many ontologies representing different parts of the energy domain; however, service developers still encounter issues when trying to apply a specific ontology or data model to their own use case or application. Therefore, rather than trying to define another domain ontology, the focus was placed on accommodating individual service needs by providing an internal flexible data model, the Digicities Core Ontology. The ontology is written using the Web Ontology Language (OWL) to provide a framework for semantic reasoning. This core model focuses on defining consistent structures for attributes and then linking to them to components. It also provides a mechanism to link term with those in established domain ontologies. The role of the Digicities core ontology is shown in Figure 8.

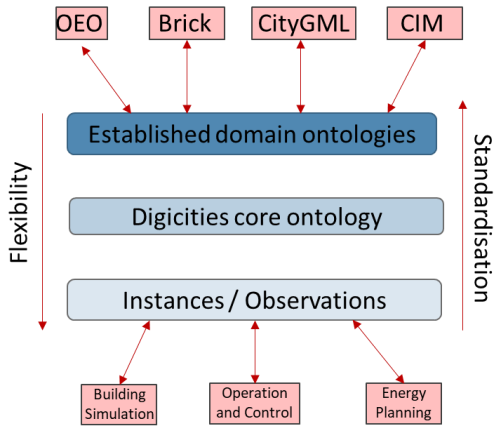

Figure 8: Digicities core ontology

[3] https://developers.willowinc.com/docs/resource-guide-digital-twin [Accessed 24/03/2026]

This section details the design of the ontology and its intended use. The ontology is also detailed in the REFORMERS D5.2 Data Space for Digital Twins, where the ontology is structuring input data for models and services integrated into the digital twin. This section details the design of the ontology and its intended use.

4.1 Ontology Architecture

The Digicities core ontology establishes a hierarchical structure organised around five primary component categories based on functional analysis of energy systems: Physical Infrastructure (devices, networks, storage), Process Components (conversion, transport, storage processes), Flow Components (energy carriers, materials, information), Spatial Components (locations, geographic context), and Actor Components (stakeholders, operators, regulators). This organisation provides semantic precision whilst maintaining flexibility for diverse energy applications.

The Device classification includes sensors, actuators, controllers, meters, and switches. Process hierarchies distinguish between conversion, transport, and storage processes, whilst Flow components differentiate between Energy Carrier Flows, Material Flows, and Information Flows. The hierarchical structure implements inheritance relationships where universal properties apply through the root Component class, whilst specialised properties address specific component categories.

4.2 Attribute Classification System

The attribute taxonomy implements a comprehensive classification system organised by fundamental data type rather than temporal behaviour. This approach recognises that temporal characteristics apply across attribute categories, enabling flexible modelling patterns where individual attributes can exhibit both time-invariant and time-varying properties.

Physical Attributes represent quantitative properties with defined measurement units, integrating with QUDT vocabularies through the hasDefaultUnit property. Cost Attributes address economic properties through SimpleCostAttribute (fixed values) and UnitBasedCostAttribute (unit-based pricing) subclasses, supporting currency specification through FIBO vocabularies. Curve Attributes represent functional relationships through coordinate pair collections with mandatory x-axis and y-axis unit specifications. Custom Physical Ratio Attributes support ratio-based measurements with distinct numerator and denominator units for derived quantities that are not included in QUDT.

Categorical Attributes handle discrete states through controlled vocabularies, Geospatial Attributes address location-based properties for spatial analysis, Event Attributes capture temporal occurrences with configurable precision (annual to timestamp resolution), and Simple Value Attributes represent unitless quantities such as configuration parameters.

4.3 Temporal Data Handling

The ontology addresses temporal representation through three distinct TimeSeries subclasses: HistoricTimeSeries (measured past data), LiveTimeSeries (real-time

measurements), and FutureTimeSeries (forecasted projections). Dynamic attributes maintain simultaneous references to multiple temporal variants through specialised data properties, enabling comprehensive temporal analysis where individual attributes reference historical validation data, real-time monitoring streams, and future scenario projections within unified semantic structures. Common temporal metadata properties include temporal bounds (startTime, endTime), temporalResolution, and data location information.

4.4 Interoperability and Mapping

The ontology incorporates established vocabularies whilst implementing systematic mapping capabilities to support interoperability with domain standards. Integration with QUDT enables standardised unit representation and dimensional analysis, whilst FIBO integration ensures standardised currency handling. The mapping framework implements three relationship types: equivalence relationships (owl:equivalentClass, owl:equivalentProperty) for semantic identity, hierarchical relationships (rdfs:subClassOf, rdfs:subPropertyOf) for taxonomic connections, and similarity relationships (skos:closeMatch) for semantic proximity. This approach supports energy system models requiring simultaneous compliance with multiple domain standards, such as building energy models requiring SAREF alongside electrical grid models utilising CIM standards.

4.5 Scenario Management

The Scenario class provides the foundation for alternative system representation, with the derivedFrom property establishing hierarchical relationships between scenarios. Component inclusion operates through the usedInScenario property applicable to components, attributes, and component links, enabling different system configurations within unified knowledge bases. The ComponentLink class represents relationships between components that vary across scenarios, supporting different system topologies and operational patterns. The Assumption framework implements systematic scenario parameterisation through targetComponent, targetAttribute, and modifier properties, enabling scenario definition through assumption sets rather than complete model duplication.

4.6 Implementation Process

The knowledge graph construction comprises two main components: classes and attributes collected via standardised Excel templates specifying component names, attributes, and semantic structure information; and physical relations handled through the linksComponent object property and subproperties linking physically connected infrastructure. Service data requirements are defined using YAML specifications that reference Digicities components and attributes, maintaining semantic consistency with formal ontology definitions whilst providing practical specification formats validated against domain and range restrictions.

5 APPLICATION ACROSS USECASES

The Digicities ontology was applied across the project's use cases to validate whether the flexible, service-driven design described in Section 4 could

accommodate fundamentally different data requirements within a single coherent framework. Each use case required the core ontology to be extended with application-specific component types and attribute names. These extensions were defined through standardised Excel ingestion templates, where each worksheet represents a component type, columns define attributes, and a multi-row header structure specifies the attribute type (Physical, Historic, Live, Future, Categorical, Event, SimpleCost, UnitBasedCost, CustomPhysicalRatio, SimpleValue, ClassObject, Resource, Annotation), units, currencies, and relationship predicates. An ingestion script processes these templates into validated RDF/TTL graphs conforming to the ontology structure. Five ontology extensions were completed across the Swiss and Austrian use cases, summarised below.

5.1 Swiss UC1 - Hydropower Plant Scheduling

This use case required a data structure for training LightGBM-based forecasting models and operating the HPP scheduling controller described in the national final report. Five component types were defined, with a strong emphasis on dynamic attributes carrying both Historic and Live temporal variants, reflecting the dual requirement of historical data for model training and live feeds for real-time control.

Table 2: Components and attributes for Swiss UC1

Component	**Inst.**	**Attributes**	**Attribute Types**
HydroTurbine	1	MaxRatedPower, Power, PowerDecisions	Physical (MegaW); Historic + Live (KiloW)
HydroReservoir	1	MinimumVolume, MaximumVolume, Volume, Inflow, FlowPredictions	Physical (M3); Historic + Live (M3, M3-PER-SEC)
ElectricityGrid	1	ElectricityDemandProfile, InstalledPVCapacity, PowerSimulated	Historic + Live (KiloW)
Weather	2	OpenMeteoWeather	Historic + Live (UNITLESS)
MLModel	2	Step, NumberOfPredictions, n_minutes_in_the_past, ModelPath	Physical (minute, UNITLESS); SimpleValue

The Weather component bundles multiple meteorological variables into a single resource file, with the internal structure handled by the consuming ML service rather than decomposed into individual ontology attributes. The MLModel component captures model configuration parameters (prediction step, horizon, training data path) as Physical and SimpleValue attributes, demonstrating the ontology's capacity to represent computational artefacts alongside physical infrastructure. Assigning a parent class for MLModel required a design decision: it does not represent physical infrastructure, but was placed under the component hierarchy as a functional element of the operational system.

5.2 Swiss UC2 – Energy Planning

The energy planning use case collected requirements for the ontology extension using three separate templates, each representing a different stage or stakeholder in the planning workflow (Existing System, Third-Party API, Technology Database). This also illustrated the data integration challenge the ontology is designed to address: even within a single use case, there was potential that the same concepts were represented differently. Additional effort will be required to identify and and manage the overlap between them.

5.2.1 Existing System - Municipal Data for Existing Models

The data structure for this source was derived from the utility's existing energy model, containing 115 instances across 9 component types. It represents a detailed system representation spanning energy carriers, supply technologies, conversion and storage technologies, transportation, and transmission.

Table 3 Components and attributes for Swiss UC2 - municipal data for existing models

Component	**Inst.**	**Key Attributes**	**Attribute Types**
EnergyCarrier	12	EnergyCost	UnitBasedCost (KiloW-HR, CHF)
Region	57	*(spatial identifiers)*	—
SupplyTechnology	4	EnergyDemandProfile, Lifetime, EnergyEfficiency, InterestRate, EnergyCap, OMAnnualCosts	Historic (KiloW); Physical (YR, ONE, KiloW-HR)
PrimarySupply	10	EnergyDemandProfile, OMAnnualCosts	Historic (KiloW); UnitBasedCost
ConversionTechnology	16	AnnualDemandProfile, Lifetime, EnergyEfficiency, InterestRate, EnergyCap, OMAnnualCosts	Historic; Physical (YR, ONE, KiloW-HR)
StorageTechnology	2	As ConversionTechnology	As above
Transportation	5	EnergyDemandProfile, TransportEfficiency	Historic; Physical
Transmission	3	EnergyEfficiency	Physical (ONE)
EnergyDemandProfile	6	EnergyDemandProfile	Historic (KiloW)

5.2.2 Data Requirements for Third-Party API

This is the data structure required by Sympheny's regional energy planning service. It uses a richer set of attribute types - including ClassObject relationships with fedBy/feeds predicates to define energy carrier flows between converters and storage and contains 55 instances across 6 component types.

Table 4 Components and attributes for Swiss UC2 – API requirements

Component	**Inst.**	**Key Attributes**	**Attribute Types**
EnergyCarrier	15	EnergyCost	UnitBasedCost (KiloW-HR, CHF)
Region	5	*(labels)*	Annotation
EnergyConsumer	5	ElectricityDemandProfile	Historic (KiloW)
EnergyGenerator	9	Power, SolarPotentialProfile	Historic (KiloW); CustomPhysicalRatio (KiloW/M2)
EnergyConverter	20	InterestRate, Efficiency, RatedPower, CAPEXPerRatedPower, CAPEX, Lifetime, TRL, OPEX, OPEX_CAP, OPEX_ENERGY, CO2_CAP, Input, Output	Physical; SimpleCost; UnitBasedCost (CHF); SimpleValue; ClassObject (fedBy/feeds)
EnergyStorage	1	Capacity, ChargingEfficiency, DischargingEfficiency, Charge_MAX, Discharge_MAX, SOC_MIN, SOC_MAX, StandbyLosses, Lifetime, Efficiency, TRL, CAPEX, OPEX, OPEX_CAP, CO2_CAP, Input, Output	Physical; SimpleCost; UnitBasedCost; CustomPhysicalRatio (ONE/hour, KiloGM/KiloW-HR); ClassObject

This was the most attribute-dense data requirements: the EnergyStorage component alone carries 17 distinct attributes spanning 6 attribute types. The ClassObject attributes define the energy flow topology that the Sympheny optimisation model requires, linking storage and converter components to their input and output energy carriers.

5.2.3 Technology Database

This data structure represents a shared technology database, intended to be published as a semantic data product, containing 1,067 instances across 5 component types. This database is an output of the energy modelling activities in the ReFuel.ch project[4]. The structure uses to cross-referencing between components to indicate linkages, for example, the fedBy/feeds relationship between EnergyConverter and EnergyCarrierFlow.

[4] https://www.sweet-refuel.ch/

Table 5 Components and attributes for Swiss UC2 – Refuel.ch Technology Database

Component	**Inst.**	**Key Attributes**	**Attribute Types**
Material	11	*(labels and descriptions)*	Annotation
EnergyCarrier	33	EnergyCarrierType	Categorical
EnergyConverter	181	Occurs, Introduced, TechRegion, MainOutput, Lifetime, TRL, CAPEX, OPEX_CAP, OPEX_ENERGY, CO2_CAP	ClassObject (occursDuring, locatedIn, hasOutput); Event; Physical (YR); SimpleValue; SimpleCost (CHF); UnitBasedCost (MegaW, CHF); CustomPhysicalRatio (KiloGM/MegaW)
EnergyCarrierFlow	421	Efficiency, Input, Output	Physical (ONE); ClassObject (fedBy/feeds)
MaterialFlow	421	ContainsCarrier, Efficiency, Input, Output	Annotation; Physical (ONE); ClassObject (fedBy/feeds)

In addition to this, the graph database, containing ontology and the instances, is used in the back end of a data exploration application to generate inputs to an energy modelling software, a screenshot of this prototype application is shown in Figure 9. This demonstrates the flexibility of the ontology to drive applications outside of the Digicities platform.

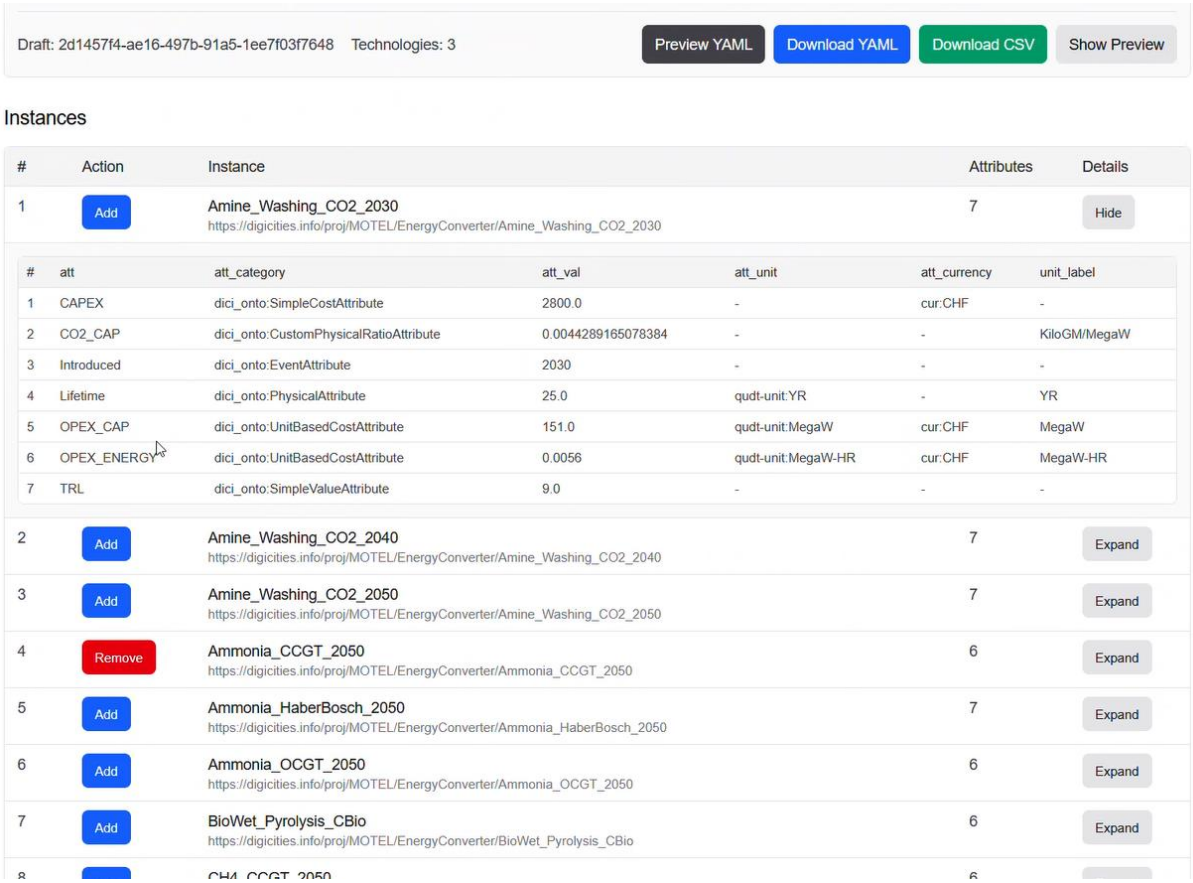

Figure 9: A protype data exploration platform connected to the GraphDB containing technology data. Credits Application: Tycho Noah Frei, Data curation: Barton Chen & Dennis Beerman. Project: SDSC Motel.

5.3 Austrian UC1 — Zero Energy Building Building Energy Dashboard

This use case required a data structure for the energy consumption and generation data for a building-level performance dashboard, with component types conceptually aligned to the Brick ontology (Section 3.2.2) distinction between energy consumers and generators. It contains 25 instances across 5 component types.

Component	**Inst.**	**Key Attributes**	**Attribute Types**
Location	1	*(label)*	Annotation
ThermalEnergyGenerator	4	Heat	Historic (kWh)
ThermalEnergyConsumer	13	Heat	Historic (kWh)
ElectricEnergyGenerator	3	Power	Historic (kWh)
ElectricEnergyConsumer	5	Power	Historic (kWh)

This is the simplest structure in terms of attribute diversity. each energy component carries only a label and a single Historic Power time series — but it validates the ontology's ability to accommodate a Brick-aligned taxonomy within the Digicities framework. The component names (ThermalEnergyGenerator, ElectricEnergyConsumer) are linked to Brick concepts, demonstrating that the

extension mechanism can accommodate external vocabulary alignment at the component level even when the underlying ontology structures differ.

5.3.1 Semantic building model

In order to represent building systems in a semantically rich way that allows automated data processing an reasoning, it was decided to model the building and its systems using the Brick schema. Brick "consists of an **extensible dictionary** of terms and concepts in and around buildings, a set of **relationships** for linking and composing concepts together, and a **flexible data model** permitting seamless integration of Brick with existing tools and databases". Thusly defined terms, relationships and data models (so called Tbox) are used to **instantiate** a concrete set of objects (ABox) that represent real-world objects. Instantiations of brick concepts that relate to real-world entities are stored in a **knowledge base**. Knowledge base can be a **static file** in one of the common triple-formats (OWL, TTL, RDF) used as a reference, or an **active database** (e.g. Fuseki) that can be **queried** (e.g. using a SPARQL language) and that may provide reasoning services over data stored in it. Reasoning over knowledgebases is based on axioms defined for certain brick terms (e.g. subsumption relationships, conjunction and disjunction between classes as well as restrictions such as existential and universal). Reasoning may yield new, implicit entities and relationships (in contrast to those explicitly defined).

Brick is commonly used to develop and link representations of a built environment (locations, buildings, spaces, zones) and its topology (spatial relationships such as containment, adjacency, connectivity,...), building systems (lighting, ventilation, heating, cooling, water delivery,...), and components of those systems (lightbulbs, automated window shades, heater, cheillers, heatpumps, ...), how components are linked together to form systems (ducts, pipes, wires), as well as the ways in which they interact with the building and building spaces (lightbulb having a location in a space, heating system delivering heat to a space,...). In addition to that, essential is the ability to instantiate and assign "points" to thusly defined elements and link them to time series stored in the file or a database. In Brick schema points are further classified in sensors, setpoints, alarms, commands, parameters and statuses.

This allows rich relationships between real world objects, their rigorously defined and classified digital representations, and recordings of building's, environmental and equipment properties to be established and used for advanced data processing.

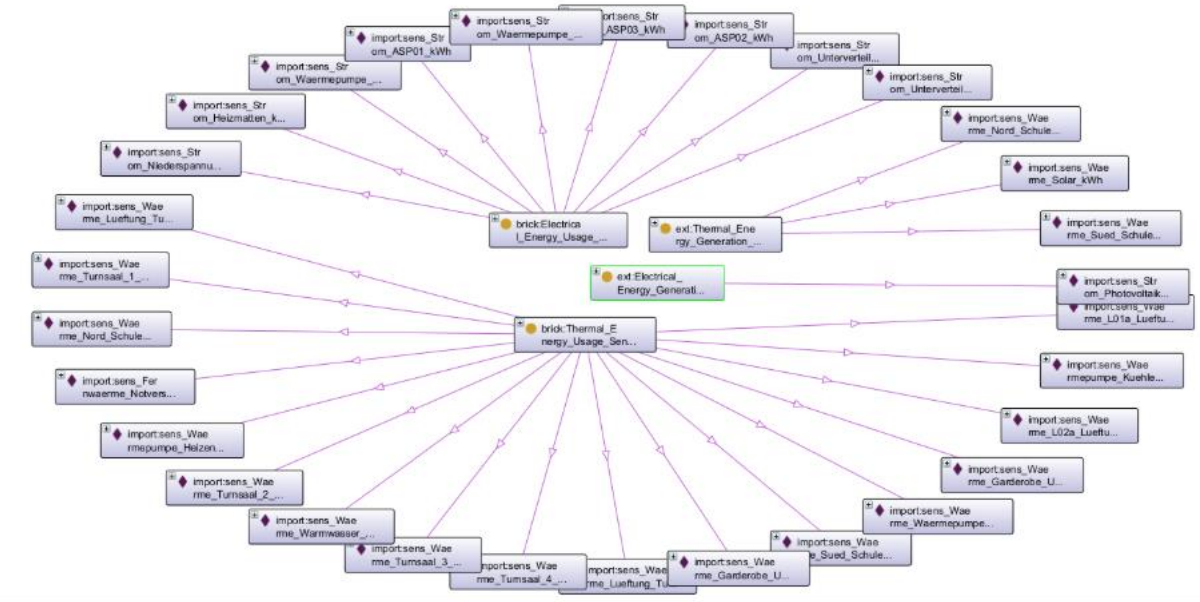

Figure 10: Energy sensors (meters) installed in building from AUC1.

Ultimately, all instantiated entities are explicitly or implicitly linked to additional defining information. In case of datapoints that includes details about measured units, observation frequencies, and similar, all described in a standardized way thereby guaranteeing quality of input information to downstream processes.

5.3.2 Data provisioning

The cleaned data is then stored and processed according to the user's data request. For example, the data for example can be requested for a certain meter or a virtual totalizing meter over the entire building. Further, the time is to be entered and corresponding to the entered information a data set is prepared by extracting the specific data from the platform and organizing it in a useful way.

5.3.3 Data integration into the Digicities data exchange platform (DEP)

In order to integrate the cleaned data into DEP, it was first necessary to align with the data semantic description. Initial AUC1 semantic description was developed using the Brick schema, while DEP uses the Digicities ontology. As AUC1 uses a limited number of underlying classes, aligning the two ontologies entailed establishing correspondence/mapping (e.g. owl:sameAs or rdfs:subclassOf), betwe en respective classes and relationships.

Original data represented by cleaned and time-aligned csv file containing multiple columns (datapoints) was split into individual files with time index and they were stored in the Nextcloud storage. File names and datapoint names were used to establish link between data. Use case semantic description in turtle format was integrated in the project Graphdb instance.

To be able to request data from a client application, an API needed to be configured which defines which data could be requested by the client application. In this case we linked the API to the AUC1 building. To demonstrate end-to-end data flow, simple client application (Energy dashboard) was created that can calculate building level KPIs similar to AUC1 Digicities dashboard.

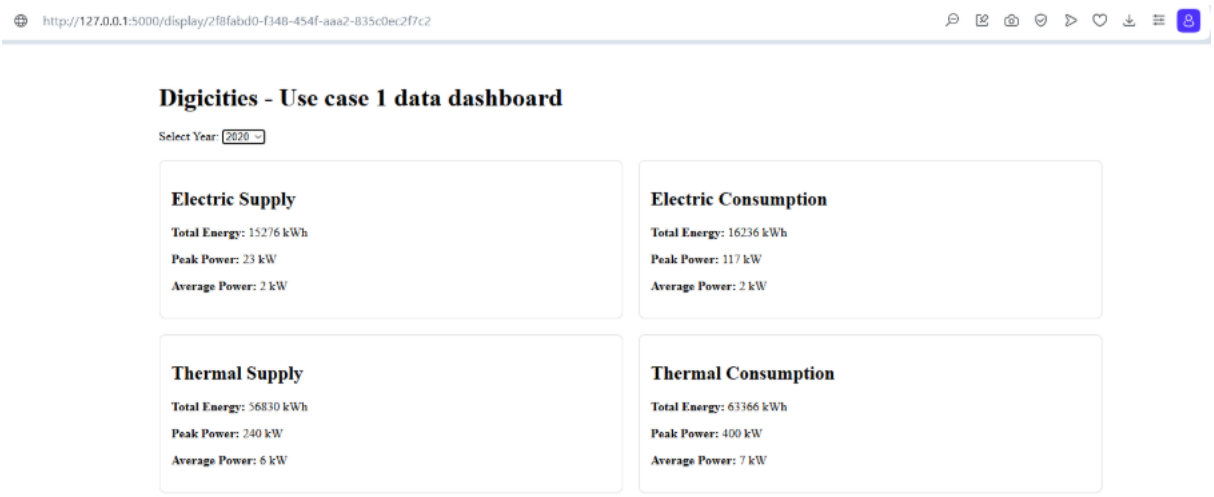

Figure 11: Simple energy dashboard application demonstrating integration of external service exchanging data with the Digicities platform. KPIs shown are being calculated across data from AUC1 building.

5.4 Austrian UC2 — Zero Energy Building Building Energy Dashboard

This use case represents an office building at AIT site, so-called "FUTURE BASE" in Vienna. It deals with the energy data on heating and cooling of the building. To heat or cool the offices, concrete core activation is used.

Several factors specify the operation modes for heating and cooling the Future Base. Sensor data was collected for over a year time range including temperature and humidity meters, window opening sensors, as well as weather data. The sensor data was mapped to rooms with the help of a BRICKs semantic model (Section 3.2.2). Based on the dataset a machine learning model was trained to predict the humidity and temperature of a single room for a 3 days time horizon.

Component	**Inst.**	**Key Attributes**	**Attribute Types**
Location	1	*(label)*	Annotation
Temperature	36	Internal Temperature	Historic (kWh)
Humidity	36	Humidity	Historic (kWh)
WindowsOpening	36	Open/close	Historic (kWh)
WetherData	1	External temperature	Historic (kWh)

5.4.1 Data-preparation:

Two sensor data source was used for the analysis: The HVAC sensor data from the existing BACnet amounting to 2700 timeseries with 47 months of history, and IoT sensor network for the room climate that collected 1200 timeseries with a 17 months of history.

To describe the semantic relations in the building a BRICK model was built containing building topology and HVAC system components as well as elements for the IoT sensors. The connections between the components are described by triples.

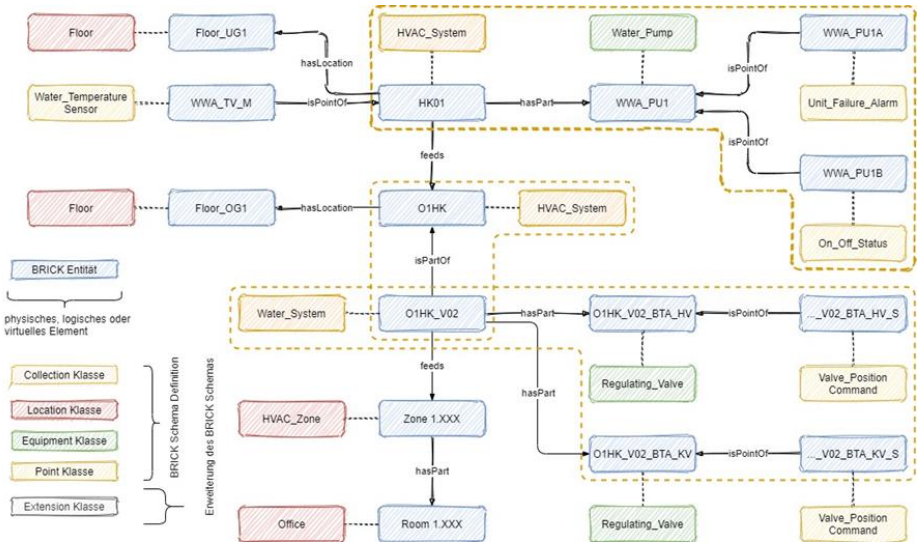

Figure 12: Brick model for AUC2

The BRICK ontology was used to find the sensors corresponding to the rooms (sensor to room mapping).

All together 19 rooms from the 1-st floor, and 17 rooms from the 3. floor were extracted from the dataset. For each room 50 features were prepared including room temperatures window/door contact values, outside temperatures, temperature set values, heating return and supply temperature, heating/cooling volume flows. Based on the time-history of these features 3 quantities are predicted: temperature of the concrete core, room temperature and the humidity in the room.

5.5 Observations and Challenges

The process of extending the ontology from these templates revealed several challenges that have implications for the platform's future development.

5.5.1 Identifying parent component classes

Each new component type required assignment to a parent class in the core ontology's hierarchy. In many cases this was straightforward, e.g. a HydroTurbine is clearly a subclass of physical infrastructure, and an EnergyCarrierFlow belongs under Flow Components. In other cases, the classification required judgement, for example, the MLModel component did not fit into the five primary categories so it was placed as a functional element of the operational system. The EnergyDemandProfile component in the existing municipal energy model is

arguably an attribute rather than a component in, but the utility's existing data structure treated it as an independent entity, and preserving that structure during ingestion took priority. The choice of pragmatism vs precision is a luxury given by the flexibility of the Digicities ontology; however, such points will need to revisited, debated and iterated to ensure alignment and interoperability of extensions if widespread application is achieved. The question is how much this revision would disrupt backwards compatibility of applications dependent on an old model. This is something to be considered in future work.

5.5.2 Decomposing embedded structures

A recurring challenge was that source data frequently embeds multiple conceptually distinct entities within a single record. For example, energy and material flows, are commonly represented as attributes of a converter rather than as independent entities. The Digicities approach required expanding these embedded structures into separate component instances. In the case of the technology database, this involved creating distinct EnergyCarrierFlow and MaterialFlow instances linked to their parent EnergyConverter through ClassObject relationships using the fedBy/feeds predicates. This decomposition is conceptually sound, flows genuinely are distinct entities with their own attributes (efficiency, carrier type) that differ from the converter's attributes (lifetime, cost, rated power). However, it represents a significant departure from how technology databases and service APIs typically represent and request this data. The willingness of publishers and services providers to accept these new conventions would need additional evaluation.

5.5.3 Attribute naming divergence.

Each use case introduced highly specific attribute names reflecting the terminology of individual service providers or domain experts e.g. PowerDecisions, FlowPredictions, SolarPotentialProfile, CAPEXPerRatedPower, Charge_MAX, SOC_MIN, StandbyLosses, OMAnnualCosts, TransportEfficiency. In addition, these came with unconventional QUDT unit descriptions for example CO2_CAP (an example CustomPhysicalRatio attribute) has the default units KiloGM/MW. While this specificity ensures exact alignment with each service's API expectations, it result in similar concepts receiving different names. Future efforts should ensure that a suitable hierarchy is established to ensure that attributes can be referenced at the highest level of abstraction for example, TransportEfficiency and EnergyCarrierEfficiency could be subclasses of Efficiency.

5.5.4 Managing interoperability across extensions.

As multiple extensions accumulate the challenge of managing overlap and ensuring consistency across the extension space will become increasingly significant. Two extensions may independently define components or attributes that refer to the same real-world concept under different names, with subtly different parent class assignments or attribute type choices. Because all extensions share the same core ontology foundation, reconciling them should more manageable than handling interoperability across multiple imported ontologies. Nevertheless, the process of identifying overlaps, deciding on names, and merging or aligning extensions would be meticulous and time-consuming. There is clear scope for AI augmentation in this

process, which could include automated detection of semantically similar attribute names, suggestion of candidate parent classes based on attribute patterns, and flagging of potential conflicts between extensions could substantially reduce the manual effort, provided that human domain expertise remains integrated into the loop for validation and final decision-making. The combination of machine-assisted pattern recognition and expert judgement represents a promising direction for maintaining ontology quality as the extensions are created.

6 DISCUSSION

The review of data vocabularies in Section 3 confirmed that no single ontology spans the physical scales (device to city), temporal resolutions (sub-hourly to decadal), and stakeholder perspectives (utility operations, municipal planning, building management) encountered in urban energy planning. CIM addresses grid operations but not building topology. Brick covers building systems and device relationships but not urban-scale spatial hierarchies or energy planning scenarios. CityGML provides rich 3D city representation but lacks native support for energy time series and was found to have incomplete OWL coverage. SAREF handles IoT device interoperability but not the technological characterisation and cost modelling that energy planning requires. Each vocabulary serves its intended domain, but the integration points are unclear. The only solution is to create global ontologies where multiple ontologies are imported, and extensions are used for anything else that cannot fit. This results in composite ontologies that are challenging to maintain after the specific application ends.

The application across the Digicities use cases helped evaluate whether the Digicities ontology's extendable, service-driven approach provides a viable alternative to adopting domain vocabularies, and the workflow from template specification through ontology extension to knowledge graph population and service submission was completed for all use cases. The attribute classification by data type was consistently applied across all five templates. The same Physical Attribute structure accommodated static configuration parameters (reservoir volume limits, technology lifetimes), slowly evolving values (installed PV capacity), and rapidly varying measurements (demand, inflow). The temporal dimension was handled the TimeSeries subclass mechanism, allowing a single attribute to carry simultaneous Historic, Live, and Future references.

The process of extending the ontology revealed that the most challenging aspect was not defining new attribute types but rather the structural decisions involved in identifying parent component classes (Section 5.5.1) and decomposing embedded component structures in the source datasets (Section 5.5.2). Future guidance for data publishers and service developers establish conventions for how multi-entity records should be structured and managed. The challenge of appropriate attribute naming (Section 5.5.3) will become increasingly apparent as the number of extensions grows. Managing interoperability and convergence of these extensions (Section 5.5.4) is expected to be a meticulous challenge that could be augmented by AI methods. One possible option is to have a controlled vocabulary layer between

the core ontology and service-specific extensions. This would provide the governance structure needed to manage this growth. For example, rather than each analytical tool defining its ideal data schema and placing the burden on users to source matching data, services could be designed to consume the attributes commonly available in the platform's vocabulary layer – driven by the publishers of data products. This would essentially result in building services around available data rather than dealing with bespoke requirements and field names. This is an idealistic future scenario that would require implementation and adoption by the multiple players spanning the value chain.

Finally, the absence of formal semantic reasoning beyond basic class hierarchy validation was felt most acutely in the multi-template UC2 scenario. Automated detection of naming overlaps, consistency checking across templates describing the same physical system, and inference of missing attributes from related components would all have been valuable but require reasoner integration and careful management of computational complexity as knowledge graphs grow. Combined with the AI-assisted governance tools discussed above, this represents the most significant area for future development of the ontology framework.

7 CONCLUSION

The Digicities core ontology provides a integration layer for a service-driven framework where components and attributes are defined according to application requirements, structured by a consistent data-type-based classification, and linked to established vocabularies through incremental mapping. The ontology was extended across five ingestion templates spanning hydropower scheduling, building energy monitoring, municipal energy planning, third-party API integration, and a shared technology database, generating approximately 1,269 component instances, exercising 12 of the 14 defined attribute types, and validating the workflow from template specification through ontology extension, knowledge graph population, and service submission. The extension process confirmed that the attribute classification by data type and orthogonal temporal handling are sound design decisions, but also revealed that the most consequential challenges are not structural but organisational: identifying appropriate parent classes requires domain judgement that resists full automation, decomposing embedded data structures into separately linked component instances departs from how technology databases are conventionally published, and the accumulation of service-specific attribute names creates semantic overlap that the current framework does not prevent. Addressing these challenges could involve publishing conventions that acknowledge decomposition requirements, a controlled vocabulary layer that encourages convergence toward canonical attribute names, AI-assisted detection of naming conflicts with human expertise in the loop, and eventual integration of semantic reasoning beyond basic class validation. The consortium's consideration of an open-source release would accelerate this work by opening the ontology and its extensions to broader scrutiny, contribution, and real-world application across a wider range of applications, within and outside the energy domain.

8 ACKNOWLEDGEMENTS

We would like to thank the need-owners and collaborators that have supported this project from the beginning. The generation of text in the report was assisted by AI tools Grammarly and Claude.ai for readability and structure.

9 REFERENCES

Azure, M., 2022. Digital Twins Definition Language [WWW Document]. URL https://github.com/Azure/opendigitaltwins-dtdl/blob/b3f36cf3fb6062a9da68ed4b085a17c7a1144763/DTDL/v2/dtdlv2.md (accessed 9.28.22).

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R., Kjærgaard, M.B., Srivastava, M., Whitehouse, K., 2016. Brick: Towards a Unified Metadata Schema For Buildings, in: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, BuildSys '16. Association for Computing Machinery, New York, NY, USA, pp. 41–50. https://doi.org/10.1145/2993422.2993577

Battle, R., Kolas, D., 2011. Geosparql: enabling a geospatial semantic web. Semantic Web J. 3, 355–370.

Biljecki, F., Kumar, K., Nagel, C., 2018. CityGML Application Domain Extension (ADE): overview of developments. Open Geospatial Data Softw. Stand. 3, 13. https://doi.org/10.1186/s40965-018-0055-6

Car, N.J., Homburg, T., 2022. GeoSPARQL 1.1: Motivations, Details and Applications of the Decadal Update to the Most Important Geospatial LOD Standard. ISPRS Int. J. Geo-Inf. 11, 117. https://doi.org/10.3390/ijgi11020117

Chadzynski, A., Krdzavac, N., Farazi, F., Lim, M.Q., Li, S., Grisiute, A., Herthogs, P., von Richthofen, A., Cairns, S., Kraft, M., 2021. Semantic 3D City Database—An enabler for a dynamic geospatial knowledge graph. Energy AI 6, 100106.

Gantner, F., 2011. A spatiotemporal ontology for the administrative units of Switzerland.

Gantner, F., Waldvogel, B., Meile, R., Laube, P., 2013. The B asic F ormal O ntology as a Reference Framework for Modeling the Evolution of Administrative Units. Trans. GIS 17, 206–226.

Grütter, R., Purves, R.S., Wotruba, L., 2017. Evaluating topological queries in linked data using DBpedia and GeoNames in Switzerland and Scotland. Trans. GIS 21, 114–133.

Guo, D., Onstein, E., 2020. State-of-the-art geospatial information processing in NoSQL databases. ISPRS Int. J. Geo-Inf. 9, 331.

H. Kolbe, T., Kutzner, T., Stephen Smyth, C., Nagel, C., Roensdorf, C., Heazel, C., 2021. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. Open Geospatial Consortium.

Hammar, K., Wallin, E.O., Karlberg, P., Hälleberg, D., 2019. The RealEstateCore Ontology, in: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan,

A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web – ISWC 2019, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 130–145. https://doi.org/10.1007/978-3-030-30796-7_9

Kolbe, T.H., Nagel, C., Herreruela, J., 2013. 3d city database for citygml. Add. 3D City Database Doc. Version 2.

Kutzner, T., Chaturvedi, K., Kolbe, T.H., 2020. CityGML 3.0: New Functions Open Up New Applications. PFG – J. Photogramm. Remote Sens. Geoinformation Sci. 88, 43–61. https://doi.org/10.1007/s41064-020-00095-z

Sporny, M., Longley, D., Kellogg, G., Lanthaler, M., Lindström, N., 2018. JSON-LD 1.1 [WWW Document]. URL https://json-ld.org/spec/FCGS/json-ld/20180607/#syntax-tokens-and-keywords (accessed 9.29.22).

University of Geneva, 2023. OWL version of the CityGML [WWW Document]. OWL Version CityGML. URL http://cui.unige.ch/isi/onto//citygml2.0.owl (accessed 3.27.23).

Wallin, E., 2022. On-demand webinar: A new standard is now available to use for harmonization between two smart building metadata standards. RealEstateCore. URL https://www.realestatecore.io/webinar-new-standard-brickrec/ (accessed 9.27.22).

WillowInc/opendigitaltwins-building, 2026.

FUNDING

This document was created as part of the ERA-Net Smart Energy Digicities project, which was funded through the through the framework of the joint programming initiative ERA-Net Smart Energy Systems' focus initiative Digital Transformation for the Energy Transition, with support from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883973.