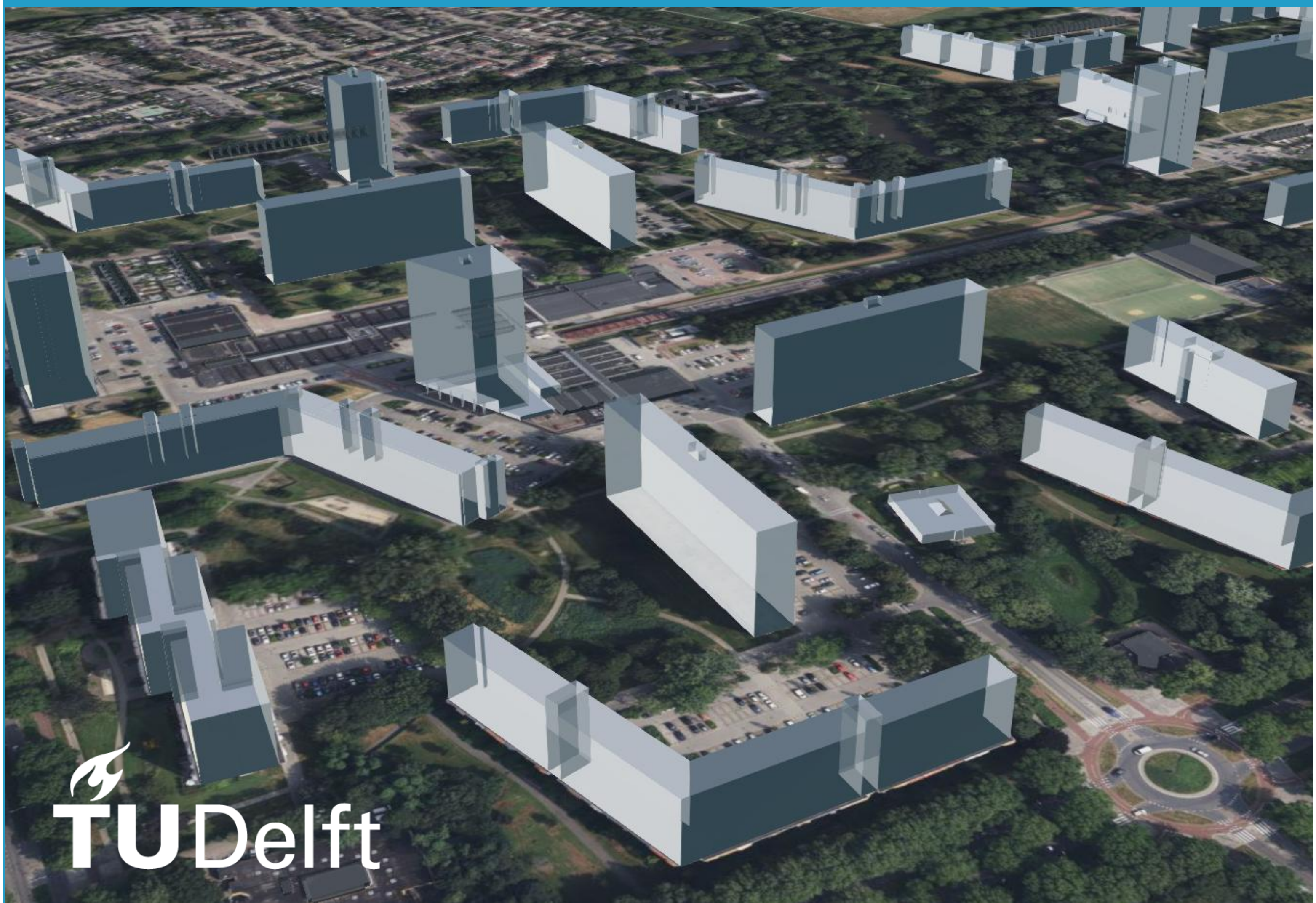
MSc Thesis in Geomatics

3D Visualization and Dissemination of Property Valuation Information Based on LADM Part 4

Xueheng Li

2025



MSc thesis in Geomatics

3D Visualization and Dissemination of Property Valuation Information Based on LADM Part 4

Xueheng Li

October 2025

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Xueheng Li: *3D Visualization and Dissemination of Property Valuation Information Based on LADM Part 4* (2025)

The work in this thesis was carried out in the:

The GIS technology group
Delft University of Technology

Supervisors:	Prof.dr. Peter van Oosterom
	Dr. Abdullah Kara
Co-reader:	Prof.dr. Christiaan Lemmen

Abstract

The Netherlands' Valuation of Immovable Property (WOZ) system underpins taxation, policy monitoring, and real-estate analytics, yet its dissemination remains largely two-dimensional, limited to cadastral footprints and single numeric attributes. This approach obscures vertical complexity in multi-storey buildings, prevents aggregation across spatial levels, and neglects temporal dynamics. With cities growing vertically and property valuation increasingly contested, there is an urgent need for methods that model and visualise WOZ units in three dimensions within a standards-compliant framework.

This thesis develops a prototype system for the 3D visualisation and dissemination of property valuation information, aligned with the Land Administration Domain Model (LADM) Part 4: Valuation Information. The system integrates cadastral (BAG), valuation (WOZ), and geometric (CityGML) datasets for a study area in Rotterdam, comprising 85 apartment complexes and 7,766 valuation units. A database schema grounded in LADM concepts links valuation units to floor-level and building-level groups, while a client–server architecture (PostgreSQL, Node.js/Express, Python, and Cesium/Vue) enables web-based interaction with valuation data across spatial and temporal scales.

Methodologically, the research contributes a heuristic algorithm for partitioning 2D floor plans of long-, L-, and square-shaped buildings into apartment units, followed by extrusion into 3D solids. A validation study using ground-truth floor plans from the Rotterdam building archive shows average correct prediction of 61–83% across building types. WOZ values are approximated at scale based on actual WOZ values of some units and dynamically visualised with colour-coded legends and temporal sliders.

The results demonstrate how LADM Part 4 semantics can be operationalised in a web-native environment, bridging legal identifiers, valuation attributes, and 3D geometries. Beyond technical feasibility, the prototype highlights new opportunities for transparent taxation, audit support, planning studies, and citizen participation. The findings underline both the potential and limitations of heuristic subdivision methods and provide a foundation for future extensions towards more complex building typologies, multi-part LA_BA Units, automated building detection.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Professor Peter van Oosterom and Dr. Abdullah Kara, for their invaluable guidance, encouragement, and constructive feedback throughout the entire course of this research. Their expertise and dedication have been instrumental in shaping both the academic quality of this thesis and my own growth as a researcher.

I am also deeply thankful to the stakeholders and collaborators who supported this project. In particular, I would like to thank Ruud Kathmann from the Waarderingskamer for his insights into the Dutch valuation system, and Pieter de Ruijter, Sjam Sardjoepersad, and Corné Helmons from the Municipality of Rotterdam for generously sharing data and engaging in collaborative discussions that greatly enriched the research. Their contributions were essential in grounding this study in practical, real-world contexts.

Finally, I owe my deepest gratitude to my mother, Li Lan. Her gentle yet unwavering love has been my greatest source of strength, giving me both the courage to make choices and the determination to continue moving forward.

Acronyms

1. Introduction

This chapter begins with a brief discussion of the motivation behind this graduation project and then explores the potential use cases of the Land Administration Domain Model (LADM) Valuation Information (Part 4). The main research questions are defined, including how to enhance the visualisation and understanding of LADM Part 4. Finally, this chapter provides a general overview of the thesis, outlining its structure and the key contributions of each chapter.

1.1. Motivation

The Netherlands' Valuation of Immovable Property (WOZ-value) database underpins property taxation, market analysis and policy monitoring, yet it is still disseminated almost exclusively through 2D parcel maps. In today's public viewer the cadastral outline of a property is projected on a flat basemap and a single numeric value appears on click. This workflow suffers from three major limitations. First, the visual language is opaque: users must inspect every record individually because colours are not employed to differentiate high- and low-valued units. Second, scale is ignored; values cannot be aggregated upward (e.g. from apartments to buildings), so patterns remain hidden. Third, the maps are static in time: stakeholders cannot explore how valuations have changed across, say, the past decade, nor correlate WOZ swings with socioeconomic or policy interventions.

A deeper—and arguably more fundamental—gap emerges once one looks upward literally. Dutch cities grow vertically while the official WOZ spatial unit remains a 2D footprint. Delineating the three-dimensional boundary of valuation objects (stacked apartments, mezzanines, underground parking, roof extensions) is indispensable for a future-proof 3D cadastre and for fair taxation that recognises volumetric reality. At present there is no operational application in the Netherlands—nor even a simple heuristic—that can predict or approximate the 3D extent of WOZ units from existing cadastral, building-permit or point-cloud data. Planners and assessors therefore often resort to manual interpretation of cadastral records or commission costly bespoke surveys to reconstruct three-dimensional property boundaries, introducing inefficiencies and inconsistencies that propagate through municipal valuation workflows and budgets [Stoter et al., 2019].

This thesis tackles these intertwined deficiencies by designing an LADM Part 4-compliant prototype that (1) colour-codes and aggregates WOZ-values across space and time, revealing trends at apartment, building and district levels, and (2) infers a plausible 3D boundary for each valuation unit, delivering the first end-to-end 3D valuation viewer for the Netherlands. Addressing the visual and volumetric blind spots simultaneously will not only improve transparency for taxpayers and professionals but also lay technical foundations for data-driven real-estate policy in the era of vertical urbanisation.

1.2. Potential use cases

The prototype developed in this thesis—a 3D, LADM Part 4-compliant platform for visualising and disseminating WOZ valuation information—serves as a proof of concept that demonstrates the potential of integrating spatial, semantic, and temporal dimensions. To illustrate its practical relevance, four concrete use cases are presented, highlighting who can benefit, what functionality is required, and how a 3D and time-enabled approach represents a significant advancement over current 2D practices.

1. Waarderingskamer audits: nationwide quality control

   The Waarderingskamer conducts annual audits of the 10 million WOZ objects. A cloud-hosted 3D dashboard that aggregates parcel-level values to building, street and neighbourhood levels gives the council an immediate visual of outliers or municipalities that diverge strongly from regional trends. This supports a risk-based audit strategy and shortens feedback loops.

2. Urban densification studies

   Planners evaluating rooftop extensions, infill projects or vertical zoning policies need to understand both existing and potential property value in three dimensions. By overlaying time-series WOZ data with BIM or LoD2/LoD3 city models, one can simulate how value redistributes when additional storeys are added, informing negotiations with developers about infrastructure levies or affordable-housing quotas.

3. Citizen transparency and participation

   Home-owners often dispute their valuation because the assessment criteria are opaque. A public viewer that colour-codes WOZ per floor and shows how a unit's value evolved over, say, the last ten years delivers the "explainability" demanded by the Dutch General Administrative Law Act [Government of the Netherlands, 1994], potentially cutting the €25–30 million that municipalities spend annually on appeal procedures.

4. Real-estate market analytics

   Brokers and investors can query aggregated WOZ growth at the building or district scale, filtering by construction year, usage type or renovation events. Coupled with transaction prices, this supports automated valuation models (AVMs) that are spatially and temporally consistent—something impossible when relying solely on 2D cadastral footprints.

Together these use cases demonstrate that a 3D WOZ information system is not merely a better map; it is an enabling infrastructure for fair taxation, smarter planning and data-driven public debate—reflecting the thesis objectives of multi-level dissemination, time-series analysis and 3D boundary prediction.

1.3. Research Questions

The central research question of this thesis is:

How can WOZ valuation information—including the boundaries of residential properties and their corresponding date and values—be effectively modelled and visualized

in three dimensions across spatial scales and time, within an LADM Part 4-compliant framework?

To address this main question, the following sub-questions are formulated:

1. Which methods and algorithms are most suitable for approximating the three-dimensional boundaries of individual WOZ units using only publicly available cadastral data?
2. How can WOZ information be systematically numbered, linked, and integrated with three-dimensional building units in a way that best reflects reality?
3. What approaches enable the extraction and population of WOZ values in an intuitive manner?
4. How can WOZ valuation information be aggregated and interactively visualized across both space and time in a dynamic and user-friendly way?

1.4. Thesis Outline

Chapter 1 provides an overview of the research background, motivation, and objectives, potential use cases and research questions.

Chapter 2 reviews existing literature and systems in the field, setting the context for the contributions of the current study. It shows how this study builds on and differs from previous work.

Chapter 3 primarily discusses the methodology for visualizing WOZ-units boundaries within a 3D land administration domain model.

Chapter 4 provides more detailed information on the design and implementation of this web-based 3D LADM prototype. It details the use of the Vue framework and Cesium for front-end development, with PostgreSQL database for backend operations.

Chapter 5 describes the result and reflection of the system.

Chapter 6 answers the research questions posed in the introduction and outlines potential directions for future research.

2. Related Work

The development of a 3D valuation information system that integrates cadastral, legal, and valuation data builds upon multiple research domains in geoinformation science and land administration. This chapter reviews the most relevant theoretical foundations, technical standards, and practical implementations that inform the design of this thesis. It begins by examining existing national platforms for property valuation, particularly the **WOZ-waardeloket!** (**WOZ-waardeloket!**), to identify their current capabilities and limitations in spatial representation and data accessibility. Subsequently, the discussion turns to the **LADM!** (**LADM!**) Part 4 standard, which provides the conceptual and semantic framework for modelling valuation information in relation to cadastral units.

The chapter then explores the design principles of land administration systems that incorporate 3D visualisation and valuation components, summarising key architectural patterns and functional requirements derived from prior studies. These insights establish the technical context for the prototype developed in this thesis. Furthermore, recent advances in geocoding multi-unit buildings are reviewed, highlighting how vertical address disambiguation and footprint-based segmentation techniques contribute to defining apartment-level valuation units. Finally, the chapter discusses the role of **CityGML!** (**CityGML!**) and 3D city models as a geometric foundation for integrating spatial and semantic information in cadastral and valuation applications. Together, these works provide the conceptual, methodological, and technical basis upon which this research builds.

2.1. WOZ-value Counter (WOZ-waardeloket) and WOZ Units

The WOZ-waardeloket is a publicly accessible web portal maintained by the Netherlands Council for Real Estate Assessment (Waarderingskamer) that allows users to view and compare the annual WOZ-value of residential properties throughout the Netherlands.

Key functionalities include:

- Lookup of WOZ values by property address.
- Display of some basic object characteristics (year of construction, usable floor area, purpose of use) sourced from register data (e.g., the Basic Registration of Addresses and Buildings (BAG)).
- Free-of-charge use for individual look-up (though automated bulk extraction is restricted).
- Comparison of key attributes among neighbouring or similar properties to support homeowners in assessing correctness of their WOZ value.

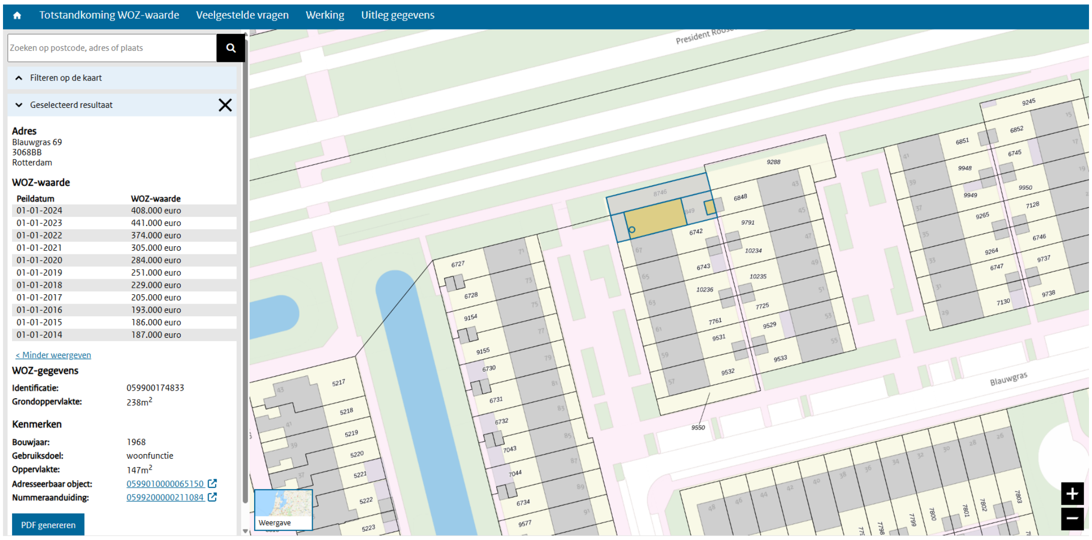


Figure 2.1.: The WOZ-waardeloket interface

From a geomatics and spatial information systems perspective, the portal provides open data access to valuation outcomes tied to locational objects (addresses). The visual interface enables users to engage with the valuation outcomes in a spatial context (address-based search, comparison).

the WOZ-waardeloket provides an open, map-based interface through which users can inspect the official property valuations (WOZ-waarde) determined annually by Dutch municipalities under the Wet Waardering Onroerende Zaken (WOZ Act). For each address, the portal displays the assessed value, year of construction, usable floor area, plot size, and designated land-use purpose, as shown in the Figure 2.1. Users can compare their property with similar ones in the neighbourhood to assess whether their valuation appears reasonable, supporting transparency and validation by homeowners. The interface offers intuitive functions such as an address search bar, a simple basemap, and a clear tabular layout, making it accessible to the general public even without specialised GIS knowledge [Waarderingskamer – Netherlands Council for Real Estate Assessment, 2025].

From a geomatics perspective, however, the platform has clear limitations. It represents valuation information only at the address level and omits the spatial and volumetric complexity of modern multi-storey buildings. No 3D geometry, floor-level detail, or explicit links between cadastral parcels and valuation units are shown, making it impossible to visualise how WOZ-units relate to underlying cadastral boundaries or ownership structures. Furthermore, the interface lacks multi-level aggregation (e.g., building $\rightarrow$ floor $\rightarrow$ apartment) and does not support temporal animation to illustrate valuation trends over time. Finally, bulk data access and advanced spatial analysis are restricted, which constrains the use of the portal for research, modelling, or large-scale policy evaluation.

In the Dutch property valuation system, a WOZ-unit represents the real-world object to which a WOZ value is assigned—typically a building, apartment, or other immovable property recorded by the municipality under the Wet Waardering Onroerende Zaken (WOZ Act). Each unit is assessed annually based on its estimated market value as of 1 January of the preceding year. The WOZ-unit thus functions as the fundamental valuation entity used for property taxation, monitoring, and comparison, integrating descriptive attributes such as building type, year of construction, surface area, and designated use.

The relationship between WOZ-units and cadastral units is close yet not one-to-one. The national cadastral register, maintained by the Dutch Cadastre (Kadaster), stores legal and spatial information about land parcels, building rights, and—in limited cases—volumetric property boundaries. Municipalities derive WOZ-units from addressable building entities recorded in the Basisregistratie Adressen en Gebouwen (BAG) and link them to cadastral parcels for spatial reference. In practice, a WOZ-unit may correspond to an entire parcel, a single building, or a sub-part of a multi-storey structure such as an individual apartment. However, this linkage remains implicit in the public WOZ-waardeloket viewer: the portal does not display cadastral geometries, parcel identifiers, or 3D subdivisions. As a result, the spatial relationship between valuation units and their legal or cadastral boundaries remains opaque to end-users, highlighting the need for integrated 3D visualisation that bridges the gap between valuation and cadastral domains Waarderingskamer – Netherlands Council for Real Estate Assessment [2025].

2.2. Land Administration Domain Model Part 4

The Land Administration Domain Model (LADM) is an ISO standard (ISO 19152) that provides a conceptual framework for describing the legal, administrative and spatial aspects of land. Its recent extension, **LADM Part 4: Valuation Information (ISO 19152-4:2024)** [ISO/TC 211, 2024], specifically addresses the domain of property valuation. According to the ISO documentation [ISO/TC 211, 2024], the scope of Part 4 is to define valuation information in relation to land administration, covering the identification of valuation units, the assessment process (single or mass appraisal), the recording of transaction prices, the generation of sales statistics, and the handling of appeals. The standard provides a shared vocabulary to improve communication and interoperability between valuation authorities, cadastral agencies, and other stakeholders, while offering an extensible foundation for the design and development of efficient valuation information systems.

The development of LADM Part 4 was guided by four principles: (i) integrating common elements from valuation systems worldwide, (ii) maintaining alignment with FIG's *Cadastre 2014* framework [Kaufmann and Steudler, 1998], (iii) ensuring practical usability by keeping the model simple and adaptable, and (iv) conforming with both ISO/TC 211 geospatial standards and international property valuation standards. In this way, LADM Part 4 does not aim to replace existing national valuation systems but rather to harmonise their semantics and enable consistent data exchange.

Figure 2.2 illustrates the core classes of the valuation module and their relations with the LADM core. The most relevant classes are summarised as follows:

- **VM_ValuationUnit**: Represents the object of assessment (e.g., parcel, building, apartment). It anchors valuation information to the legal or physical space defined in the core LADM.
- **VM_Valuation**: Records the outcome of a valuation, including the assessed value, the date, and the method used. It allows multiple valuations of the same unit across time.
- **VM_TransactionPrice**: Stores actual transaction prices from property sales, providing empirical benchmarks to calibrate valuation models and support transparency.

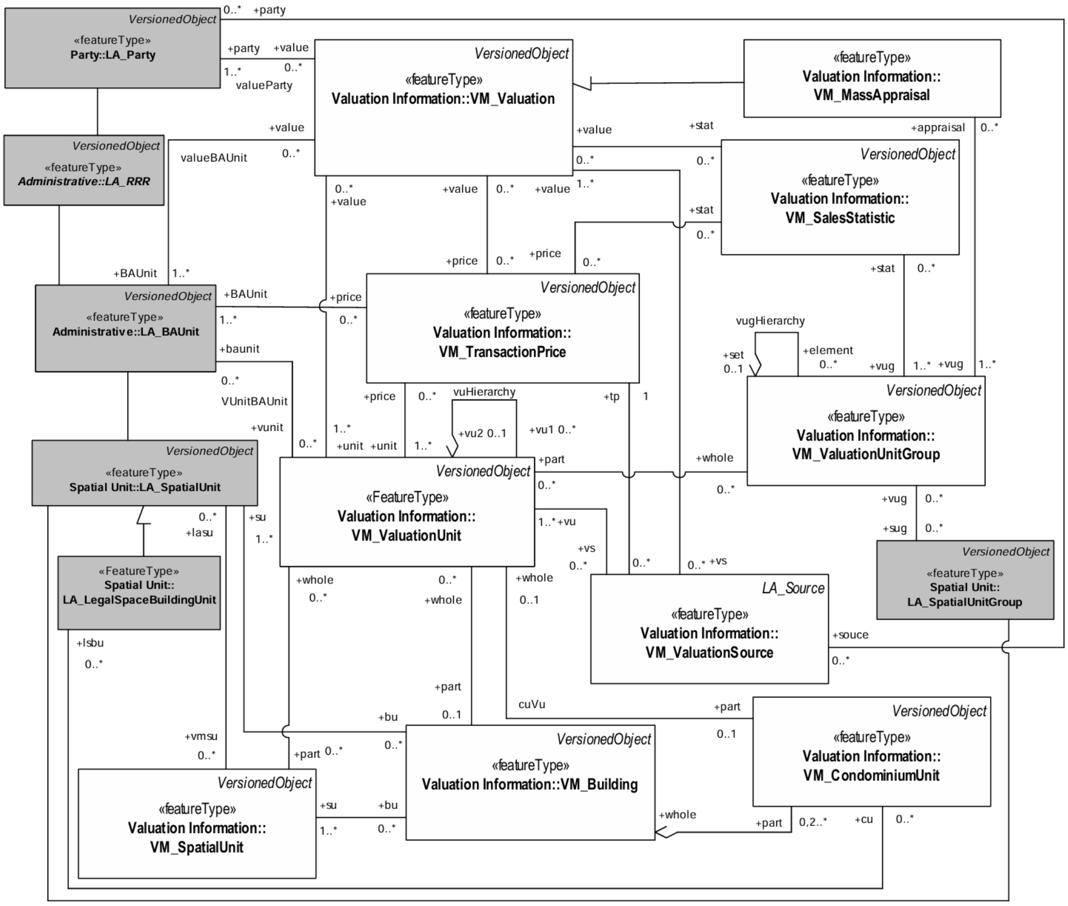


Figure 2.2.: Classes of the LADM Valuation Information and their relations with the core LADM

- **VM_ValuationUnitGroup**: Supports hierarchical aggregation of units (e.g., apartment → floor → building → neighbourhood), which could be seen in figure 2.3. This enables the publication of indices and average values at multiple levels.
- **VM_Building** and **VM_CondominiumUnit**: Specialisations of valuation units, explicitly representing buildings and individual condominium units.
- **VM_SalesStatistic**: Represents aggregated statistical indicators (e.g., mean value, price index) derived from valuations or transactions, typically used for market analysis or policy monitoring.
- **VM_MassAppraisal**: Captures information about the models, parameters, and approaches applied in large-scale valuation processes.
- **VM_ValuationSource**: Refers to the data sources or legal instruments on which a valuation is based (e.g., cadastral registers, transaction databases, planning permits).
- **VM_SpatialUnit**: Provides the spatial representation of valuation units, ensuring that valuation data can be linked to the geometries of parcels, buildings, or other legal

spaces defined in LADM.

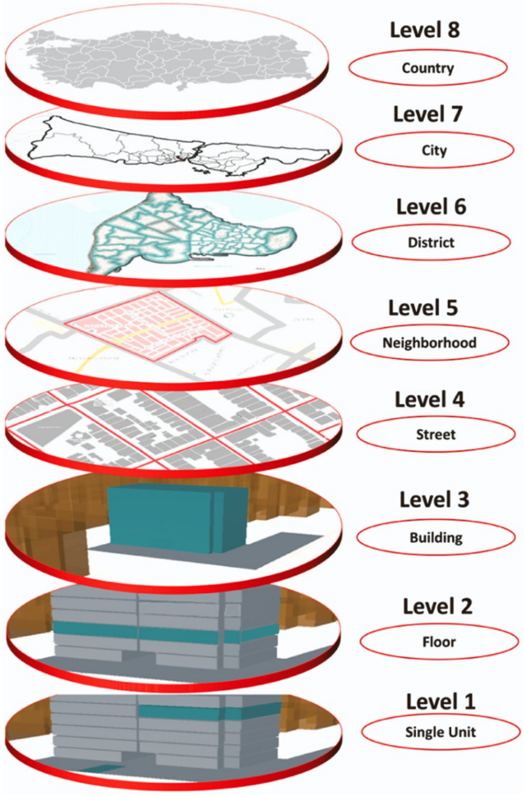


Figure 2.3.: Aggregation of valuation units into valuation unit groups

By introducing these classes, LADM Part 4 establishes a semantic and structural foundation for integrating valuation information with land administration systems. This facilitates use cases such as taxation, expropriation compensation, land consolidation, insurance assessment, real-estate financing, and market analytics, while enabling both detailed unit-level records and aggregated indicators to be modelled consistently.

2.3. Land Administration System Design

Designing a Land Administration System (LAS) that can handle volumetric property rights requires more than simply "adding a Z-coordinate" to a 2D cadastre; it demands a coherent set of requirements, an architecture able to satisfy them, and an implementation path that balances performance with interoperability. LADM Part 4 provides the semantic backbone for valuation data, but turning the specification into a working information service demands a dedicated prototype design. Four recent studies together sketch a complete blueprint: [Kara et al., 2021] requirements for web-based dissemination; [Kara et al.] translate those requirements into a first LADM–VM–compliant viewer; [Kara et al., 2020] demonstrate how 3D analytics (e.g. viewshed) enrich the valuation register; and [Kara et al., 2023] extend the viewer to multi-level statistics and unit groups.

2.3.1. System architecture

The seminal requirements analysis by [Shojaei et al., 2013] groups the needs of a 3D cadastre into three categories—cadastral, visualization and non-functional features—summarised in their Table A.1.

[Shojaei et al., 2013] translate those abstract requirements into a three-layer visualisation workflow (presentation, application and data access) — depicted in the figure 2.4 — which doubles as a high-level LAS design pattern.

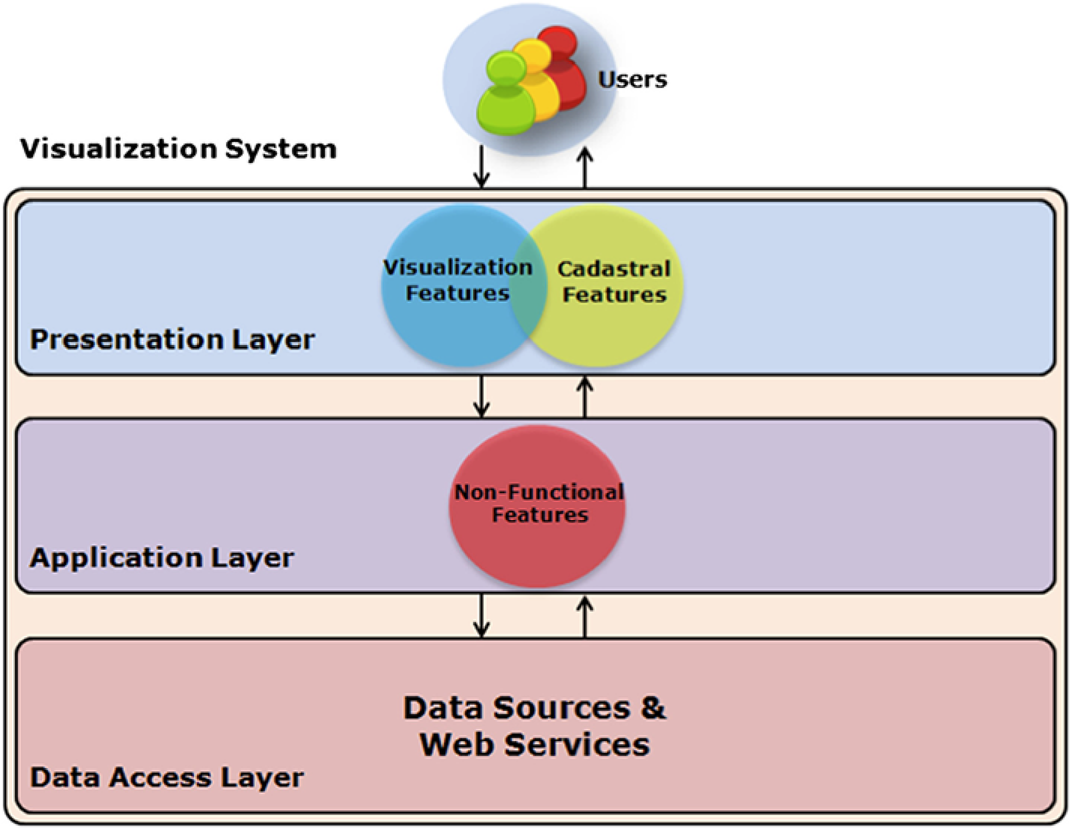


Figure 2.4.: A general architecture for 3D cadastral visualization systems[Shojaei et al., 2013]

By separating concerns the model allows each tier to evolve independently (e.g. swapping a rendering engine without touching the database schema) and supports future service-oriented extensions recommended by [Hildebrandt and Döllner, 2010].

Furthermore, using a U–D–V matrix, [Kara et al., 2021] classify needs as User, Data and Visualisation requirements. These requirements underpin every design choice in the subsequent prototypes.

1. User - Two principal audiences emerge: the general public/owners who demand intuitive thematic maps and professionals (assessors, planners) who need query and analysis tools.

2. Data - Systems must handle single-part properties, thematic attributes (value) and temporal versions.

3. Visualisation - Features such as colour ramps, occlusion management, addressing searching and aggregation to valuation-unit groups are flagged as essential.

The 2022 prototype [Kara et al.] pioneers a three-tier web stack (PostgreSQL/PostGIS – Flask API – CesiumJS front-end) that satisfies the requirement for plug-in-free access and heavy-dataset streaming. Building on that, the 2023 extension [Kara et al., 2023] introduces:

1. 3D-Tiles tiling for Level of Detail (LoD)-controlled delivery of millions of apartment solids.

2. Dynamic aggregation services that materialise unit-group statistics on-the-fly (e.g. average WOZ by floor).

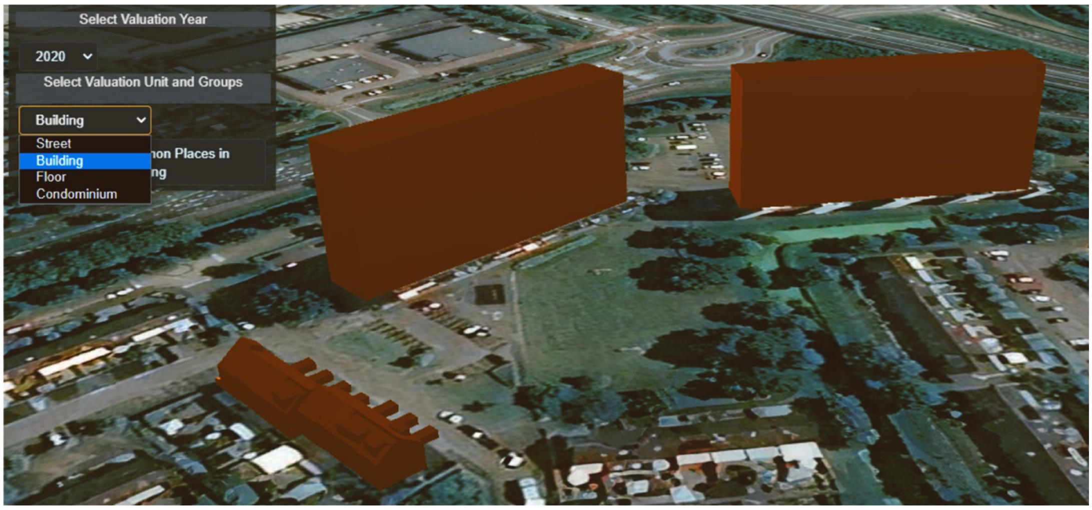


Figure 2.5.: Specified levels for visualisation of valuation information

Figure 2.5 illustrates the most recent user interface, featuring colour-coded valuation units, an aggregation dropdown, and a temporal slider for exploring changes over time. Despite these functionalities, the current generation of three-dimensional unit boundaries across spatial levels still depends heavily on manual intervention. Moreover, the scale of the existing case study is limited, encompassing only two building complexes with eight residential units. To address these limitations, this thesis proposes the design of a prototype that introduces greater automation into the workflow and expands the coverage to a larger study area. Such an approach enhances both the practicality and the representativeness of the system, bringing it closer to operational applicability.

2.3.2. Implications for this thesis

The converging evidence from the four papers legitimises a prototype strategy that is LADM-VM compliant, analytics-aware and web-native. Concretely, this thesis will:

1. Reuse the VM class schema and Dutch country profile [Kara et al., 2019] as the database backbone.
2. Adopt the aggregation taxonomy (unit → group) to publish floor-level and neighbourhood-level WOZ statistics.
3. Implement a Cesium/Vue front-end with façade slicing, temporal slider and value-change palettes.
4. Pre-compute viewshed-based quality indicators to test their explanatory power in the evaluation chapter.

By aligning with the prototypes and design recommendations reviewed here, the thesis positions itself within the cutting edge of 3D valuation information systems while extending the discourse with a fully open-source, Part 4-verified Dutch implementation.

2.4. Multi-unit Building Address Geocoding

Address geocoding traditionally assumes that every civic number occupies a unique ground-level centroid. In dense Dutch apartment blocks, however, dozens of stacked units share the same 2D parcel footprint. Classic techniques—linear interpolation along a street segment, parcel-centroid assignment or address-point mapping— therefore collapse, because they ignore floor height, façade orientation and irregular unit numbering schemes (e.g. 12-A vs 12-3).

Early attempts to solve this "vertical ambiguity" relied on detailed indoor reference data. [Lee, 2004], [Lee, 2009] first interpolated distances along internal corridors and then snapped estimates to surveyed unit centroids, but his workflow presupposed a full 3D Building Information Model (BIM) of the building, something rarely available at national scale. No open-data method existed that could geocode apartments across many buildings without such privileged input, leaving emergency routing, e-commerce logistics and 3D cadastre initiatives without a reliable locator.

[Nottrot et al., 2023] addresses this gap by proposing a two-stage pipeline that needs only publicly available Dutch key registers (BAG, Basic Registration of Large-Scale Topography (BGT)), building outlines and basic address attributes. The figure 2.6 shows a baseline linear interpolation distributes apartment identifiers evenly around the exterior polygon, assigning each unit (i) a clockwise or counter-clockwise façade segment and (ii) an estimated floor derived from total building height. Second, an explainable machine-learning model—CatBoost gradient-boosted decision trees—learns residual patterns from 4700 labelled apartments, using features such as standardised house number, estimated floor count, apartment floor area and parcel size.

The comparative evaluation shows that CatBoost cuts horizontal (x,y) median error from 17.9 m to 12.5 m and combined 3D error from 18.6 m to 14.2 m, while preserving the 3 m vertical (z) accuracy inherited from the uniform-floor-height assumption. Precision in predicting which façade side hosts the entrance jumps from 0.60 to 0.95, a key benefit for emergency dispatchers who must choose the correct doorway.

Although CatBoost does not materially improve floor-level prediction—likely because of sample scarcity above the 10th storey—it statistically outperforms interpolation in overall positional accuracy (Wilcoxon $p < 0.001$).

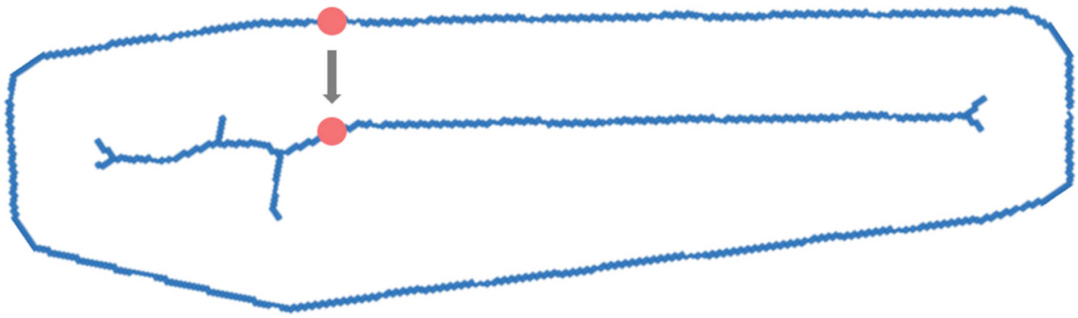


Figure 2.6.: A full shift from building outline to medial axis

The study also introduces a one-dimensional façade parameterisation (-100 ... +100) that is agnostic to complex floor-plan geometry, enabling uniform modelling across courtyard and L-shaped blocks while sidestepping the need for indoor scans.

By coupling Optical Character Recognition on notarial deeds and floor plans with automatic building-orientation detection, the authors demonstrate an end-to-end pipeline that can scale beyond the 70 buildings used for training, albeit with noted generalisability limits to multi-parcel structures and courtyard-only units.

For this thesis, which aims to visualise WOZ valuation units in 3D, the work of [Nottrot et al., 2023] offers two key insights. First, façade-based interpolation combined with boosted-tree refinement provides a pragmatic alternative to full BIM when predicting the 3D boundaries of stacked valuation objects. Second, the reported 14 m median error establishes a realistic benchmark against which the boundary-prediction component of this research can be evaluated, particularly in the context of integrating apartment-level WOZ values into an LADM Part 4–compliant 3D cadastre viewer.

2.5. CityGML and 3D City Models[Kutzner et al., 2020], [Open Geospatial Consortium (OGC), 2021]

CityGML is an international OGC standard for the representation, storage and exchange of virtual 3D city models [Gröger and Plümer, 2012]. It defines both the geometric and semantic aspects of urban objects, such as buildings, terrain, transportation infrastructure and vegetation, structured across multiple Levels of Detail (LoD0–LoD4), as shown in the figure 2.7. The LoD concept allows users to choose an appropriate trade-off between geometric complexity and semantic richness, depending on the application. This semantic–geometric integration distinguishes CityGML from purely geometric 3D formats (e.g., Object File Format (OBJ), 3D Studio File Format (3DS)), which only describe shape and appearance but lack meaningful thematic attributes. As a result, CityGML has become a widely adopted foundation for applications in land administration, urban planning, disaster management, and property valuation.

From a cadastral perspective, CityGML is particularly relevant because it provides a structured way to represent buildings as volumetric objects that can be systematically linked to legal spaces and valuation units. This capacity makes CityGML suitable for integration with

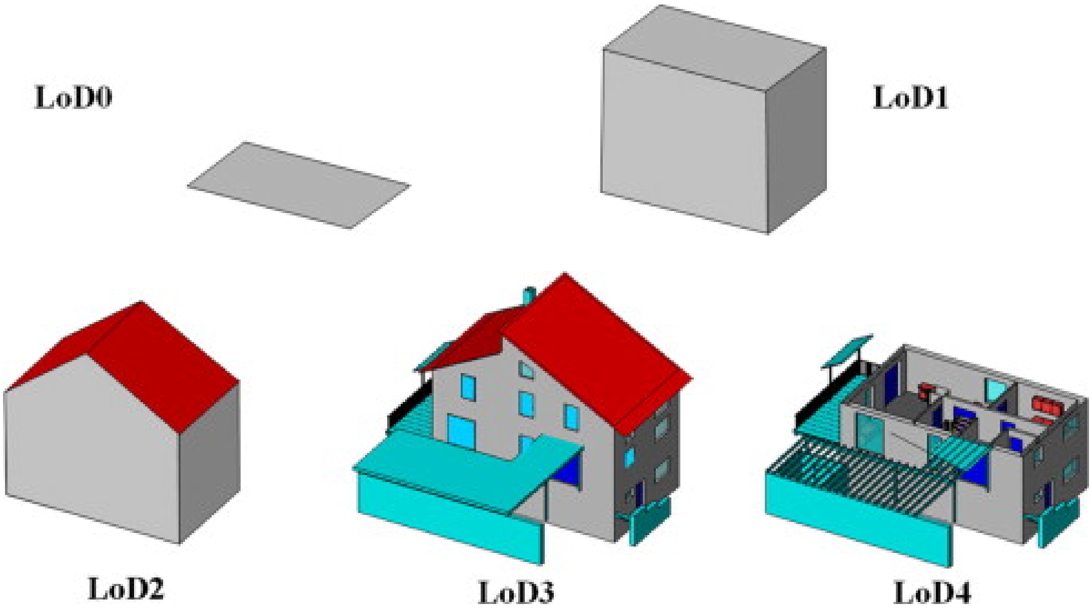


Figure 2.7.: Multiple Levels of Detail (LoD0–LoD4)[Gröger and Plümer, 2012]

LADM-based frameworks, supporting 3D cadastre and valuation workflows. In practice, CityGML LoD2 and LoD3 building models are especially valuable: LoD2 captures realistic roof structures and building shells, while LoD3 introduces detailed façade openings and structural elements. These levels enable the subdivision of complex building geometries into units that can be associated with property rights, restrictions, and responsibilities. Moreover, the flexibility of CityGML's Application Domain Extensions (ADEs) allows the schema to be adapted to specific thematic requirements, including cadastral or valuation information [Kutzner et al., 2020]. The development of ADEs for land administration and taxation illustrates the growing convergence between CityGML and standards such as LADM.

Recent research has explored CityGML's potential for property valuation and taxation workflows. [Goetz, 2013], for instance, demonstrated how CityGML can integrate heterogeneous building and socio-economic datasets, highlighting its interoperability across domains. [Biljecki et al., 2015] provided a systematic review of 3D city model applications and identified property valuation and taxation as an emerging field of interest, while [Biljecki et al., 2016] investigated the effect of LoD on energy demand estimation and valuation-related analyses. Their findings indicate that the level of geometric detail significantly influences the outcomes of valuation and energy modelling, underscoring the importance of choosing appropriate LoDs in practice.

Despite these strengths, CityGML also faces challenges in large-scale operational use. First, the availability of detailed LoD3 or LoD4 models is limited, especially for existing building stock, where only generalized LoD1 or LoD2 data may be available. Second, the storage and streaming of complex CityGML datasets pose technical challenges, often requiring conversion into more efficient delivery formats such as CityJSON or 3D Tiles. Third, semantic alignment with national base registers (e.g., BAG in the Netherlands) is not straightforward and requires careful identifier matching to ensure consistent linkage between 3D geometries and legal or valuation units.

For the purposes of this thesis, CityGML serves as a crucial geometric backbone. The Rotterdam municipality provided LoD2 building models in CityGML format for the study area in Prins Alexander, which were re-projected, harmonised and transformed into watertight solids before being integrated into the LADM Part 4-compliant database. These solids formed the starting point for the 3D subdivision workflow, enabling the derivation of apartment-level volumes that could be linked to WOZ valuation records. In this sense, CityGML not only provided a geometric basis for constructing the 3D valuation units, but also demonstrated its role as an interoperable bridge between spatial data infrastructure and property valuation systems. By leveraging CityGML, this research contributes to the growing body of evidence that semantic 3D city models are indispensable for building operational 3D cadastres and property valuation services.

3. Methodology

This chapter explains the methodological framework adopted to achieve the research objectives. As illustrated in Figure 3.1, the workflow is organized into four sequential stages:

1. Data acquisition – gathering the spatial, legal and valuation datasets required for analysis;
2. Data integration – harmonising and linking the heterogeneous sources into a single LADM-compliant database;
3. Algorithm design – developing the procedures for 3D boundary prediction;
4. Visualization Implementation – defining legends and colors, and implementing animation for models across different levels, and adding addressing searching function;
5. Accuracy evaluation – assessing geometric and thematic correctness against the actual floor plans.

3.1. Data Acquisition

The first methodological step was to assemble the datasets needed for analysis. Because the research aims at an operational, city-scale application rather than a laboratory proof-of-concept, partnering with a Dutch municipality was essential. The Municipality of Rotterdam expressed strong interest and offered an introduction to its cadastral and valuation registers as well as technical support for data provisioning.

During a preparatory workshop with the municipality's geo-information team, the existing 3D viewer was reviewed. This platform already integrates BGT topography, BAG building and address data, and WOZ valuations into a browser-based scene. While it renders individual buildings at a high level of detail, it does not yet support the visualisation or querying of individual WOZ objects within multi-storey complexes—precisely the gap that this thesis seeks to address.

We jointly specified the data package required for the prototype, resulting in the deliverables listed later in Table 3.1. The agreed study area is a residential district in Prins Alexander, comprising 55 apartment complexes and 7766 registered WOZ units. This neighbourhood provides a representative mix of stacked dwellings, making it well-suited for testing the 3D boundary-prediction and visualisation workflows described in subsequent sections.

Because of privacy constraints, the municipality is unable to release the ten-year time series of WOZ assessments. To fill this gap, the historical valuation figures will be obtained independently from the publicly accessible Wozwaardeloket portal, which publishes de-identified WOZ values for Dutch addresses with buildings and private properties.

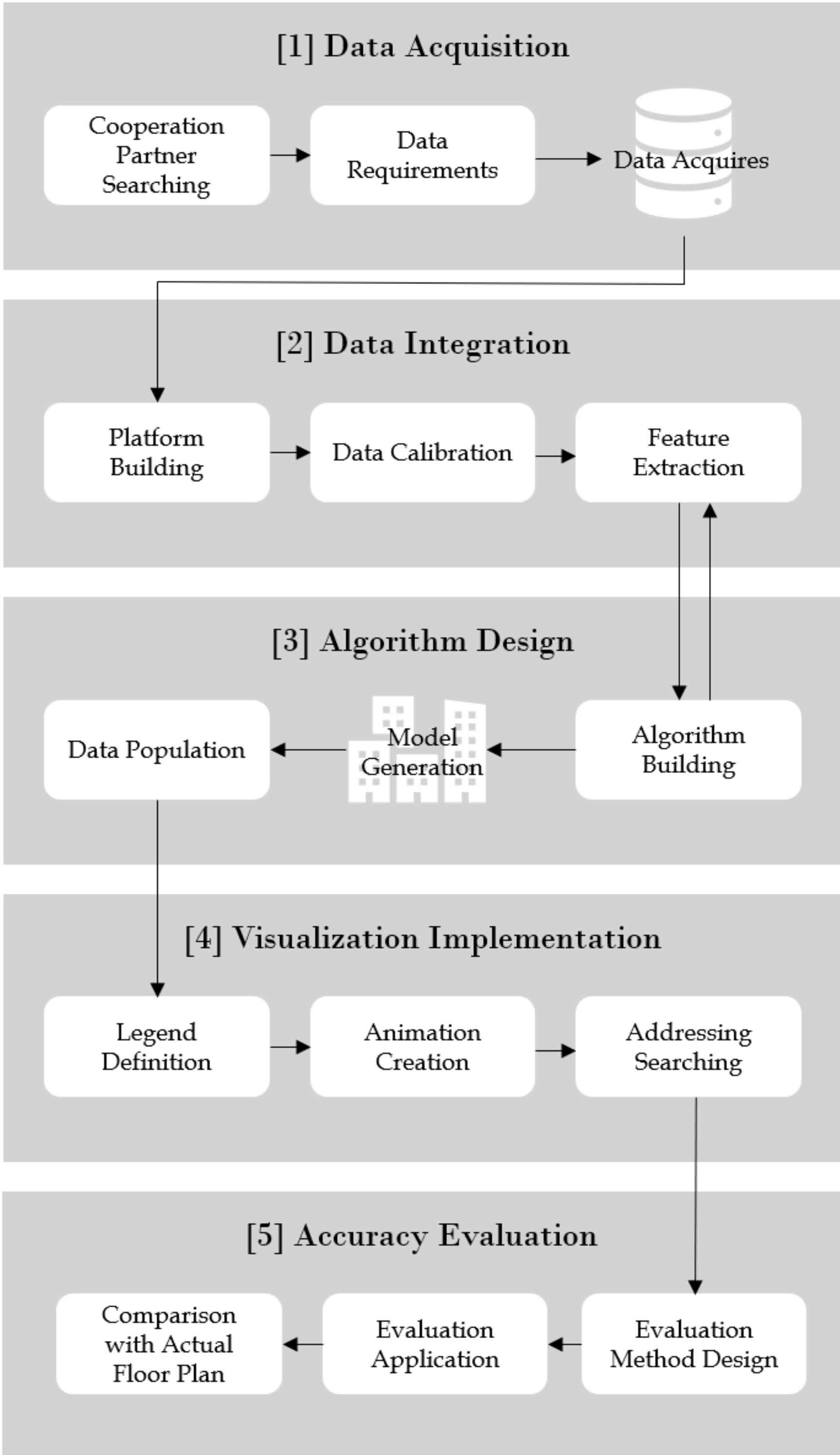


Figure 3.1.: Flowchart of methodology

Table 3.1.: Datasets requirements sent to Rotterdam municipality in the neighborhood Prins Alexander

Type (Residential Buildings)	**Format**	**Time**
3D Building Models	CityGML	Current
Number of Floors	Number	Current
Floor Plan	2D graph	Current
WOZ-Object	Addresses	Current
	WOZ-Value	Recent 10 years
	Identificatie Kenmerken	Current
	Bouwjaar	Current
	Gebruiksdoel (only *woonfunctie*)	Current
	Oppervlakte (m^2)	Current
	Adresseerbaar Object (BAG → building unit)	Current
	Nummeraanduiding (BAG → address)	Current

3.2. Data Integration

Once the source datasets were secured, the next task was to fuse them into a single, transaction-ready repository that conforms to the Valuation Information Model (VM package) of LADM Part 4. Figure 3.2 summarises the extract–transform–load (ETL) pipeline; the text below explains each step.

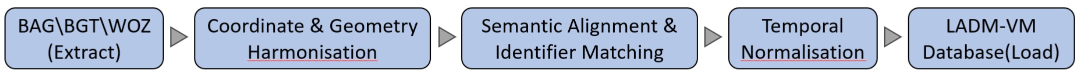


Figure 3.2.: The Extract–Transform–Load (ETL) Pipeline

3.2.1. Coordinate and Geometry Harmonisation

CityGML building models were re-projected from the Dutch national CRS Rijksdriehoekscoördinaten (Dutch National Triangulation Coordinate System) (RD) New (EPSG:28992) [EPSG: Geodetic Parameter Registry, 2023a] to WGS84 [EPSG: Geodetic Parameter Registry, 2023b] , ensuring sub-centimetre alignment within the Cesium rendering environment. To comply with the ISO 19107 solid geometry profile required by LADM, CityGML multi-surface geometries were further dissolved into watertight solid objects using the `ST_MakeSolid` function.

To reduce rendering load during web delivery, LoD2 roof and façade geometries were simplified into barycentric triangular meshes and exported as 3D Tiles. The original solid geometries in WGS84, however, were preserved in PostGIS for analytical queries requiring full volumetric precision.

3.2.2. Semantic Alignment and Identifier Matching

Table 3.2.: Linking the Dutch base registers to the LADM Valuation Information Model

Register	Primary key	LADM mapping	Linking rule
BAG – panden	`bag_pand_id`	`LA_SpatialUnit`	One-to-one with CityGML building solid (`bag_pand_id` attribute)
BAG – adresseerbaar objecten (AON)	`bag_aon_id`	`VM_ValuationUnit`	Spatial join to associated `bag_pand_id`; inherits vertical extent from the subdivision algorithm in Section 3.3
WOZ records	`woz_object_id`	`VM_Valuation`	Join on `bag_aon_id`; many-to-one relation (multiple yearly valuations per unit)

Table 3.2 summarises the links between Dutch base registers and the LADM Valuation Information Model. Referential integrity is enforced through foreign-key constraints: deleting a BAG object cascades to its dependent valuation units, ensuring database consistency across annual BAG updates.

3.2.3. LADM-VM Logical Schema

The Valuation Information Model (VM package) of LADM Part 4 extends the core LADM standard (ISO 19152-1 and 19152-2), which defines the foundational classes for parties, rights, restrictions, responsibilities (RRRs), and spatial units. While the core LADM establishes these building blocks, the VM package adds a value component that supports systematic integration of valuation data.

Using Enterprise Architect, the VM UML diagram was converted into a PostGIS logical schema via the Model-Driven Architecture (MDA) plug-in. Key tables include:

1. `vm_valuation_unit` (eg. `unit_id`) – apartments, storage boxes, garages
2. `vm_valuation` (composite eg. `unit_id`, `valuation_date`) – yearly WOZ values, valuation approach, assessor ID
3. `vm_unit_group` – dynamic aggregations (building, street, district) generated by materialised views
4. `la_spatial_unit` – reference to BAG geometry stored as solid or surface

All monetary fields are typed as `numeric(13,2)` and accompanied by an ISO 4217 `currency_code` field to ensure conformance with core LADM requirements. Integrity checks disallow negative surface areas and enforce that valuation dates fall within the life span of each unit.

The integration process produces two artefacts that feed subsequent workflows:

1. **PostgreSQL database** – serves as the authoritative repository for all WOZ information across different spatial levels.
2. **3D-Tiles service** – streamed via Cesium Ion proxy; attributes include building ID, unit ID, enabling client-side styling without additional REST calls.

Together, these outputs form the bridge between the database backbone and the interactive 3D visualisation environment, ensuring both analytical robustness and efficient dissemination.

3.3. Algorithm Design and Implementation

This stage translates the harmonised database into an operational 3D valuation viewer. The implementation follows a two-pipeline architecture:

1. a server-side processing pipeline that predicts apartment solids, enriches them with derived indicators, and publishes them as 3D Tiles;
2. a client-side rendering pipeline that styles, aggregates, and animates those tiles interactively in the browser.

The decision to employ the **3D Tiles** format is motivated by both technical efficiency and interoperability. 3D Tiles, an open standard developed by the Open Geospatial Consortium (OGC) and originally proposed by Cesium, is specifically designed for *streaming massive heterogeneous 3D geospatial datasets* over the web. Unlike monolithic formats (e.g., CityGML or OBJ), which require complete models to be downloaded before use, 3D Tiles employ a hierarchical tiling structure with multiple Levels of Detail (LoD). This allows the client to request only the portions of the dataset that are currently visible and at the resolution appropriate for the camera distance.

From a geomatics perspective, these characteristics are crucial when visualising valuation units at multiple scales (from individual apartments to whole districts), as they ensure smooth navigation in the browser without overwhelming bandwidth or memory. Furthermore, 3D Tiles are widely supported in web-based virtual globe environments such as CesiumJS, guaranteeing compatibility with open-source tools and future extensions. Thus, the choice of 3D Tiles aligns with the thesis objective of creating a scalable, web-native, and standards-compliant platform for disseminating valuation information.

3.3.1. 3D Boundary-prediction Workflow

The objective is to build a watertight solid for every BAG AON (Adresseerbaar Object) in the study area, even though only the gross building solid is known a priori. The workflow consists of 2 sequential modules implemented in PostgreSQL 17.4 and Python 3.13.

1. Vertical partitioning

   Input = CityGML LoD2 solid + `bag_aantal_verdiepingen` (floor count). A uniform-height assumption divides the Z-extent into N horizontal slabs of equal height

   $$h_{\text{floor}} = \frac{z_{\text{max}} - z_{\text{min}}}{N}.$$

An offset of 0.15 m is added above the terrain to avoid numerical overlap with the BGT ground surface.

2. Horizontal allocation

   The exterior shell is unfolded to a 2D façade ribbon. House numbers are linearly interpolated along the ribbon, in which the house numbers are generally getting larger from north to south and from east to west; each segment is then extruded back to 3D between z_i and z_{i+1}. This yields one *prismatic cell* per apartment address.

3.3.2. Client-side Rendering and Interaction

The front end is written in Vue 2 + CesiumJS 1.115:

1. Styling engine – WOZ values are visualised through a colour scheme defined by legend-based ranges, with the colours dynamically updating over time to reflect temporal changes.
2. Temporal slider – binds the viewer clock to the valuation_date attribute; a timeline toggle allowing users changing the time to the target year.
3. Group filter – dropdown lists multiple vm_unit_group from database; selecting one triggers a flyTo and applies aggregation colouring.

3.3.3. Implementation Stack Summary

Table 3.3.: Technology stack of the prototype

Layer	**Technology**	**Purpose**
Database	PostgreSQL 17 & PostGIS 3.3	Storage, spatial functions, bi-temporal logic
Processing	Python 3.13 (GeoPandas, `py3dtiles`), PL/pgSQL	Geometry generation, indicator computation
API	FastAPI 0.110	REST endpoints, JSON-LD encoding
Client	Vue 2, CesiumJS 1.115, Tailwind CSS	Rendering, UI/UX

The table 3.3 summary the stacks used in the implementation.

With the algorithms implemented and deployed, the prototype can now (i) infer plausible 3D boundaries for all 7766 WOZ units in Prins Alexander, (ii) stream them efficiently to the browser, and (iii) let users explore value patterns interactively through space, time and aggregation level. The next section evaluates how accurately these predicted solids match the true apartment geometry derived from floor-plan drawings.

3.4. Accuracy Evaluation

The final stage of the methodology benchmarks the prototype against authoritative reference data in order to quantify its reliability for property valuation workflows. Accuracy is evaluated along two complementary dimensions:

1. **Geometric fidelity** – the extent to which the predicted apartments reproduce the true legal plan.

2. **Thematic correctness** – the extent to which WOZ values and derived indicators remain consistent once re-allocated to the predicted geometry.

3.4.1. Ground-truth construction

Authoritative floor-plan drawings were obtained from Rotterdam's building-permit archive for 23 apartment buildings (approximately 42% of the study area). These were provided as PDF files at 1:200 scale. Each plan is imported into AutoCAD and converted into vector entities, then exported as `DXF` files.

The result is a ground-truth dataset of close to 600 legal units (as shown in the Figure **??**) spanning a range of floor counts (0–20), plan shapes (long, L-shaped and square buildings).

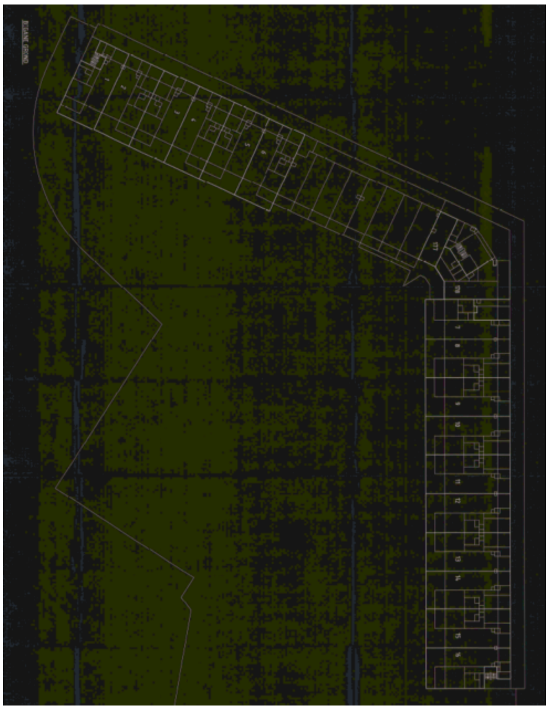


Figure 3.3.: A vectorized floor plan of a L-shaped residential building

3.4.2. Sampling strategy

To ensure representativeness while keeping the evaluation tractable, 3 building was selected from each geometric archetype: long-shaped, L-shaped, except square-shaped buildings, cause the floor plans are not available according to the Rotterdam Municipality. These buildings typically contain between 150 and 200 apartment units each. The predicted subdivisions generated by the prototype were compared against their ground-truth counterparts. For every unit, the percentage of spatial overlap between predicted and reference geometry was calculated. The overall building accuracy was derived by averaging these overlap percentages across all units in the sample.

3.4.3. Geometric metrics

2 complementary indicators were implemented to quantify shape agreement between the predicted (p) and ground-truth (g) geometries (Table 3.4).

Table 3.4.: Geometric accuracy metrics

Symbol	Metric	Formula	Interpretation
d_H	Hausdorff distance	$d_H = \max\left\{\max_{x \in S_p} \min_{y \in S_g} \|x - y\|, \max_{y \in S_g} \min_{x \in S_p} \|x - y\|\right\}$	Maximum surface gap between prediction and ground truth.
d_c	Centroid offset	$d_c = \|c_p - c_g\|$	Positional bias between predicted and true centroids.

Here$S$ and c denote volume, surface set and centroid, respectively, with subscripts p (predicted) and g (ground truth). Calculations were performed in PostGIS 3.3 (`ST_3DIntersection`, `ST_Volume`, `ST_HausdorffDistance`) and Python 3.13 (NumPy for centroid offsets).

Following Nottrot et al. [2023], the acceptance thresholds were defined as: IoU $\geq$ 0.70 **or** $E_v \leq 15\%$ *and* $d_H \leq 0.50$ m.

3.4.4. Thematic metrics

To assess thematic correctness, attributes attached to each predicted valuation unit were compared with those of the ground truth. For each unit, the relative difference was calculated as:

$$\Delta\text{attr}_i = \frac{A_i^{\text{pred}} - A_i^{\text{true}}}{A_i^{\text{true}}} \times 100\%.$$

where A_i represents an attribute such as WOZ value, floor area, or a derived indicator (viewshed, solar potential, or noise exposure). Aggregated statistics include the **Mean Absolute**

Percentage Error (MAPE) and the 95th percentile. For WOZ values, errors were additionally normalised to €/m^2 in order to decouple valuation bias from area bias.

3.4.5. Results aggregation and hypothesis testing

Metric distributions were summarised by their median, inter-quartile range (IQR) and 90th percentile to reduce sensitivity to outliers. A **Wilcoxon signed-rank test** was used to test the null hypothesis that the median volumetric error equals zero ($\alpha = 0.05$). In addition, stratified one-way ANOVA was employed to evaluate variation in accuracy across floor levels and building shapes, with post-hoc Tukey tests identifying significant subgroup differences. **The post-hoc Tukey test** (also known as the Tukey Honest Significant Difference test) performs pairwise comparisons between group means after an ANOVA indicates that at least one group differs significantly. It controls the family-wise error rate by adjusting for multiple comparisons, ensuring that the detected differences—for instance, between accuracy levels of long-, L-, and square-shaped buildings—are statistically robust.

Through this multi-layered evaluation strategy, the study quantifies not only the geometric fidelity of the predicted solids but also the extent to which any residual errors affect the consistency of valuation attributes. In this way, the evaluation closes the loop between 3D modelling accuracy and its practical impact on property-tax workflows, ensuring that the prototype can be judged not merely as a geometric tool but as a valuation information system.

4. System Design and Implementation

This chapter details the technical realisation of a web-native platform for disseminating WOZ valuation information in three dimensions. Building on the conceptual requirements outlined in the previous chapters, we translate the LADM Part 4 (Valuation Information Model) into a working system that supports end-to-end data flow: from ingestion and harmonisation of Dutch base registers to interactive 3D visualisation and analysis. The presentation follows the system's layers.

Architecture Overview introduces the architectural overview, motivating a lightweight single-page client (Vue + Cesium) connected to a modular service tier (Node.js/Express) and a spatially enabled persistence layer (PostgreSQL/PostGIS). Then the backend design is described, including API routing, connection pooling, and the orchestration of a Python-based DXF processing pipeline for plan subdivision. The data model and persistence section formalise how apartment-level identifiers, floor registries, and valuation attributes are stored and indexed for common access patterns. Subsequent sections cover the frontend design and UI, georeferencing adjustments required for consistent WGS 84 rendering, and the DXF generation function that bridges CAD inputs with web delivery. The section concludes with the algorithmic principles for 2D partitioning and the extrusion workflow that generate apartment solids for 3D Tiles streaming, and clarifies how the implementation aligns with key LADM Part 4 classes to ensure semantic interoperability and extensibility.

4.1. Architectural Overview

The prototype is developed as a single-page web application that integrates 3D geovisualization capabilities. The client side is implemented using Vue 2 with Element-UI for interface components, while Cesium provides the interactive 2D/3D scene environment for exploring spatial data. The front-end modules include functions for visualization, drawing, styling, and the upload of DXF files, enabling users to interact with the spatial content in a flexible way.

The client communicates with a Node.js/Express API, which manages the data exchange between the front-end and the back-end services. The API exposes endpoints to query, retrieve, and send spatial information, and it ensures that the front-end remains lightweight while delegating computation and storage tasks to the server.

On the server side, the system connects to a PostgreSQL database, which stores the property-related datasets used in the prototype. The database schema includes object-level and floor-level information, supporting detailed queries on building properties. A connection pool allows efficient communication between the Node.js server and the PostgreSQL database.

For handling DXF files, a dedicated Python worker is employed. This component uses libraries such as ezdxf and shapely to parse, transform, and convert uploaded DXF drawings into a structured JSON format, which can then be integrated into the web visualization.

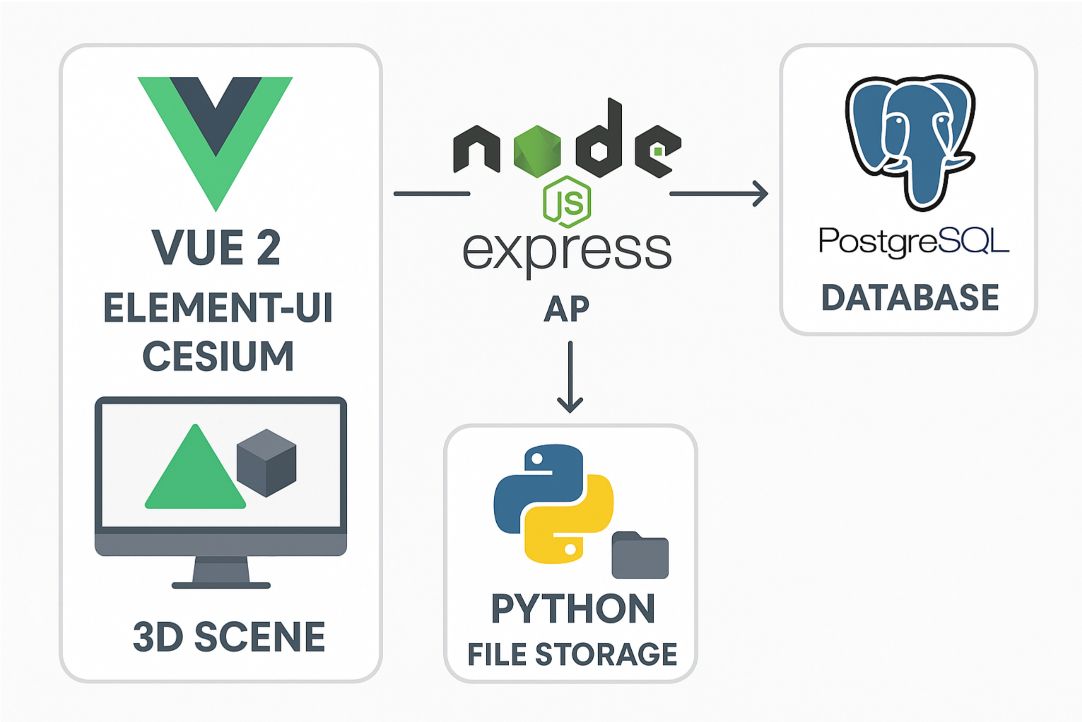


Figure 4.1.: Architecture Overview

Finally, a local file storage system is used to manage temporary uploads and generated outputs, ensuring that the DXF files and their converted products are available for subsequent visualization and analysis within the application.

Together, these components establish a coherent workflow: the front-end enables user interaction and visualization, the Node.js/Express API orchestrates communication, the PostgreSQL database provides structured data management, the Python worker processes specialized DXF inputs, and the local storage supports file handling. This architecture ensures that heterogeneous geospatial data can be efficiently processed, stored, and visualized in both 2D and 3D within the prototype.

4.2. Backend design

The backend of the prototype serves as the central coordination layer that connects the client interface, the database, the DXF processing pipeline, and the file storage system, which is illustrated in the Figure 4.2. Its design follows a modular approach, where the web server provides a uniform API while delegating specialized tasks such as database queries and DXF file transformations to dedicated components.

At its core, the server application is built on Node.js with the Express framework, providing a lightweight and asynchronous environment for handling multiple client requests. The application is configured with middleware for cross-origin resource sharing, authentication, and error handling, ensuring secure and reliable communication with the front-end client.

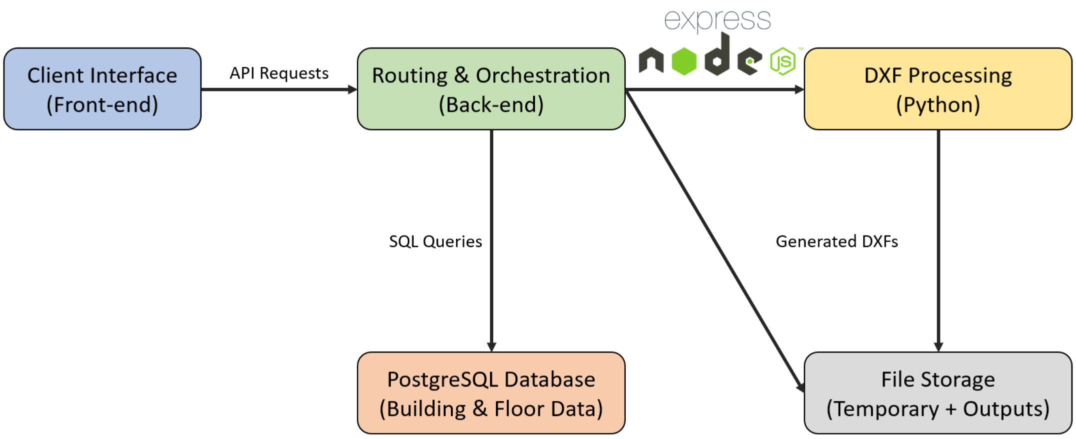


Figure 4.2.: How the backend interacts in the workflow

The API endpoints are mounted under a unified /api namespace, offering both retrieval and submission functionalities.

The routing logic exposes endpoints that allow users to query building information and floor identifiers, retrieve detailed records, and upload DXF files for further processing. These routes are designed around common workflows of property valuation visualization, where the user first selects a building of interest and then requests the system to generate floor-level representations. To safeguard server performance, the upload service enforces a maximum file size and stores incoming data temporarily in memory before delegating further handling.

Each API request is dispatched to a handler module, which translates the client's request into corresponding operations. For database interactions, the handlers make use of a PostgreSQL connection pool, ensuring efficient reuse of connections and balanced workload management when multiple queries are issued in parallel. Data access patterns are deliberately kept simple, relying primarily on parameterized SELECT statements. In particular, grouping by attributes such as bwlg_vb0 allows the system to efficiently derive counts of apartments per floor, a key metric for linking WOZ valuation records with building structure.

A crucial responsibility of the backend is orchestrating the DXF processing pipeline. When a DXF file is uploaded, the server first writes the file to a temporary location. It then analyses the relevant building and floor groupings from the database and, for each group, spawns a dedicated Python worker process. This worker employs external libraries to parse the DXF geometry, subdivide it according to the requested archetype, and write out new per-floor DXF files. Once the processing is complete, the server cleans up the temporary inputs and collates the outputs, returning a structured summary to the client that includes the number of generated files.

This backend design ensures that responsibilities are clearly separated: the Node/Express service focuses on routing and orchestration, the database layer provides authoritative building and floor information, and the Python module handles specialized geometric transformations. Through this division of labour, the prototype achieves both flexibility and scalability. The architecture makes it possible to extend the system with additional endpoints, to incorporate further valuation attributes, or to integrate more advanced processing algorithms, while maintaining a consistent interface to the front end.

4.3. Data model and persistence

The persistence layer of the prototype is designed to provide a structured and reliable foundation for storing both the property valuation data and the spatial references required for visualization. Four tables are central: woz_objects (apartment-level source data), floors (floor model registry), apartment (apartment model registry), and woz_valuation (valuation attributes). The backend exposes read endpoints that query these tables directly to support interactive exploration and DXF-driven updates.

id [PK] inte	wozobjectnr character varying	straat character varying (2	hnr characte	hltr characte	pstc character v	pandid character varying (20)	vboid character varying (20)	numid character varying (20)	bwjr integer	bag_gebruiksdo character varyin	woz_gel characte	woz_gebruik character va	bwlg_vb0 integer	laagste_bwlg_pnd integer	hoogste_bwlg_pnd integer	aant_bwlg_pnd integer
1	059900172175	André Gideplaats	385		3069EK	0599100000035588	0599010400023032	0599200100023847	1971	woonfunctie	1181	Galerijflat	6	0	8	9
2	059900104226	André Gideplaats	423		3069EL	0599100000035588	0599010000024071	0599200000179518	1971	woonfunctie	1181	Galerijflat	7	0	8	9
3	059900262133	André Gideplaats	421		3069EL	0599100000035588	0599010000195418	0599200000179517	1971	woonfunctie	1181	Galerijflat	7	0	8	9
4	059900038272	André Gideplaats	417		3069EL	0599100000035588	0599010000268265	0599200000179515	1971	woonfunctie	1181	Galerijflat	7	0	8	9
5	059900294772	André Gideplaats	415		3069EL	0599100000035588	0599010000089212	0599200000179514	1971	woonfunctie	1181	Galerijflat	7	0	8	9
6	059900070659	André Gideplaats	411		3069EL	0599100000035588	0599010000199465	0599200000179512	1971	woonfunctie	1181	Galerijflat	7	0	8	9
7	059900070470	André Gideplaats	409		3069EL	0599100000035588	0599010000038992	0599200000179511	1971	woonfunctie	1181	Galerijflat	7	0	8	9
8	059900280270	André Gideplaats	407		3069EL	0599100000035588	0599010000138713	0599200000179510	1971	woonfunctie	1181	Galerijflat	7	0	8	9
9	059900056326	André Gideplaats	399		3069EK	0599100000035588	0599010000208062	0599200000179506	1971	woonfunctie	1181	Galerijflat	6	0	8	9
10	059900071091	André Gideplaats	397		3069EK	0599100000035588	0599010000067713	0599200000179505	1971	woonfunctie	1181	Galerijflat	6	0	8	9
11	059900022762	André Gideplaats	393		3069EK	0599100000035588	0599010000035590	0599200000179503	1971	woonfunctie	1181	Galerijflat	6	0	8	9
12	059900159000	André Gideplaats	371		3069EK	0599100000035588	0599010000016200	0599200000179493	1971	woonfunctie	1181	Galerijflat	5	0	8	9
13	059900116257	André Gideplaats	367		3069EK	0599100000035588	0599010000292450	0599200000179491	1971	woonfunctie	1181	Galerijflat	5	0	8	9

Figure 4.3.: The woz_objects table in PostgreSQL Database

The first table, referred to as woz_objects, which is shown in the Figure 4.3, acts as the primary source of building and floor information. Each row corresponds to a single apartment unit and stores identifiers such as the building key (pandid) and the floor indicator (bwlg_vb0), the descriptions of all attributes in the table are shown in the table 4.1. This one-row-per-apartment structure ensures that the unit is the fundamental element of the system, which aligns with the requirement of disseminating valuation information at the most granular level. By maintaining direct references to building and floor identifiers, the table supports straightforward aggregation upwards to building or group levels, thereby serving both detailed and summary queries.

The second table, floors, functions as a registry of the 3D assets that represent floor models within the Cesium viewer. Each record contains an identifier for the asset, as well as spatial placement parameters such as geographic coordinates, elevation, and orientation. These attributes enable accurate georeferencing of floor models, ensuring that the outputs of the DXF pipeline can be positioned correctly in the virtual environment.

The apartment table provides the bridge between apartment-level models and valuation records. It maps a per-apartment model assetid to the wozobjectnr used throughout the valuation datasets. This enables the client to resolve which valuation attributes apply when an individual apartment model is selected in the 3D scene.

Finally, woz_valuation stores the valuation attributes, queried by wozobjectnr. The API supports fetching valuations for one or multiple apartments in a single request, which is used to annotate units and summarize selections on the client side.

Table 4.1.: WOZ dataset attributes and their descriptions

Attribute	Description
WOZOBJECTNR	Unique identifier of the WOZ object (valuation unit). Used in taxation.
STRAAT	Street name (from BAG address register).
HNR	House number (huisnummer).
HLTR	House letter (huisletter) or suffix (e.g., "A", "B"). Empty if not applicable.
PSTC	Postal code.
PANDID	Unique BAG building ID (bag_pand_id), referencing the physical building.
VBOID	BAG *verblijfsobject* ID (bag_verblijfsobject_id) – the unique identifier for an addressable unit (apartment, office, shop).
NUMID	BAG *nummeraanduiding* ID – the identifier of the address label (links street, house number, postal code).
BWJR	*Bouwjaar* – year of construction of the building.
BAG_GEBRUIKSDOEL	BAG usage purpose of the unit (e.g., woonfunctie = residential use, kantoorfunctie = office).
WOZ_GEBRUIKSCODE	WOZ usage code (numeric classification used in valuation, e.g., 1181 = residential type).
WOZ_GEBRUIKSCODE_OMS	Description of the WOZ usage code (e.g., Galerijflat = gallery-access apartment).
BWLG_VBO	Floor indicator (e.g., which level the unit belongs to).
LAAGSTE_BWLG_PND	Lowest building floor (for the building, in the dataset).
HOOGSTE_BWLG_PND	Highest building floor (for the building, in the dataset).
AANT_BWLG_PND	Total number of building floors.

Given the way the API filters and retrieves data in practice, the main access patterns include: distinct `pandid` queries for building listings, composite `(pandid, bwlg_vb0)` filters for floor-level scoping, direct `wozobjectnr` lookups for valuation retrieval, and primary-keyed access to `AssetID` within the floor and apartment registries. The `AssetID` serves as the unique identifier embedded in the Cesium models, enabling direct linkage between database records and 3D visual entities. In this prototype, the `floors` table designates `AssetID` as its primary key. Given that `pandid`, `bwlg_vb0`, and `wozobjectnr` are the most frequently queried attributes, creating indexes on these columns is recommended to enhance query performance, particularly when scaling to larger datasets.

Overall, the schema intentionally keeps only what is needed to connect valuation identifiers to the 3D assets. Keeping entities at the apartment level (woz_objects, apartment) with a thin registry of floors (floors) and a direct valuation table (woz_valuation) avoids duplication while preserving clear links between as-built geometry, unit identifiers, and valuation attributes. This design leaves room to extend with time-stamped valuations or additional LADM Part 4 entities without disrupting current client workflows.

4.4. Frontend design and UI

The frontend of the prototype is implemented as a single-page application that combines modern web interface components with a high-performance 3D geovisualisation engine. Its purpose is to provide users with an intuitive environment for exploring property valuation data, navigating between two- and three-dimensional perspectives, and initiating specialised functions such as DXF generation.

The application is built on the Vue 2 framework, with Vue Router and Vuex managing navigation and state across different interface modules. Styling and interactive controls are provided through the Element-UI library, while a global stylesheet ensures a consistent appearance throughout the application. This modular structure not only improves maintainability but also allows individual interface elements to evolve independently of the core viewer.

At the centre of the user experience is the Cesium-based 3D viewer, which forms the primary environment for spatial and temporal exploration. This component renders the building models and valuation data, and integrates custom drawing and styling utilities to support thematic visualisation. These utilities allow interactive highlighting of buildings and floors, dynamic styling based on valuation attributes, and flexible rendering of custom geometries. Together, they ensure that both apartment-level details and aggregated statistics can be communicated effectively through visual means.

In addition to visualisation, the frontend provides a dedicated interface for DXF file management. Through a specialised component, users can select a DXF file from their local machine, specify the building to which it relates, and choose a subdivision archetype. This component also handles communication with the backend by transmitting the file and associated parameters, while providing feedback to the user through confirmation messages once the files have been successfully processed. By integrating this workflow seamlessly into the interface, the system bridges traditional CAD-based representations with the web-based 3D viewer.

All communication with the backend is mediated through HTTP requests managed by Axios. These requests follow a simple API contract, enabling the retrieval of building identifiers, the submission of DXF files, and the receipt of structured responses. The reliance on a standardised communication library ensures that the client remains responsive and capable of handling multiple simultaneous operations, while also simplifying error handling and user notifications.

Overall, the frontend design balances technical complexity with usability. It combines a robust 3D visualisation engine, a clear interface for data upload and processing, and a lightweight communication layer. This design ensures that users can not only explore spatial valuation data interactively but also contribute new geometric inputs, making the system both a viewer and a participatory tool in the management and dissemination of property valuation information.

4.5. Georeferencing adjustments

When visualising the CityJSON models in the local Cesium web viewer, small planimetric and vertical offsets were observed relative to their expected positions (Figure 4.4). These discrepancies are consistent with mismatches between the source *coordinate reference system* (CRS) and the viewer's rendering frame: the Dutch *Amersfoort / RD New* grid (EPSG:28992) and orthometric heights (NAP) versus Cesium's WGS 84, which uses ellipsoidal coordinates for rendering.

(a) Horizontal offset (b) Vertical offset

Figure 4.4.: Offsets observed in the horizontal and vertical directions

In this prototype, the offsets were mitigated by applying a manual reference adjustment within the web viewer. A more robust production workflow would apply an explicit CRS transformation prior to streaming (e.g., RD New → WGS 84 with the official RDNAPTRANS procedure, and NAP → ellipsoidal heights), ensuring consistent planimetric and vertical alignment without post hoc correction.

4.6. DXF generation function

An essential feature of the prototype is the ability to generate floor-specific DXF files from a single building footprint supplied by users. This functionality is required to approximate the geometric subdivision of multi-storey buildings, so that each floor can be visualised and evaluated individually within the 3D environment. The process follows a clear client–server workflow, beginning with a user action in the front end and culminating in the creation of multiple DXF outputs stored on the server.

The workflow is triggered in the front end through the dedicated upload interface, where the user selects a DXF file, specifies its associated building identifier, and indicates the intended floor plan archetype. These inputs are transmitted as form data to the backend through an API request. On receipt, the server temporarily stores the file, associates it with the corresponding building record, and inspects the database to determine the number of units per floor. This information forms the basis for orchestrating the subsequent subdivision process.

The subdivision itself is performed in a Python module, which is invoked from the Node.js backend as a worker process. The Python script reads the uploaded DXF file using the ezdxf library, focusing on core geometric entities such as lines and lightweight polylines. These primitives are converted into intermediate data structures that facilitate further computational operations. Depending on the archetype chosen by the user, one of several tailored subdivision routines is applied. In the case of L-shaped layouts, the algorithm reorganizes line segments into a repeating pattern that reflects the number of apartments per floor. For

Long-shaped layouts, an alternative routine ensures that the radial symmetry of the plan is preserved during subdivision. For more complex Square-shaped plans, the algorithm first computes an internal splitting line through geometric intersection operations, then clips the polygons into distinct units accordingly.

After the relevant subdivision has been applied, new DXF entities are constructed to represent the derived apartments. These entities are written out into separate DXF files, each corresponding to one floor group within the building. The files are named according to a convention that combines the building identifier and the floor indicator, allowing them to be easily linked back to database records and visualised within the Cesium client.

Once the DXF files have been generated, the backend performs a final validation step by checking the number of outputs against the expected counts derived from the grouped apartment records. A structured response is then returned to the client, confirming the success of the operation and reporting the number of generated files. The outputs themselves remain stored on the server's filesystem for later access and integration.

Throughout this workflow, a number of conventions and constraints are enforced to maintain consistency. Specific DXF layers are reserved for building geometry and metadata, ensuring compatibility with common CAD environments. Numeric rounding rules are applied to minimise floating-point artefacts in the subdivision process, and floor counts are always cross-checked with the authoritative database to avoid discrepancies between geometry and valuation data. These measures ensure that the generated outputs are both technically valid and semantically aligned with the rest of the prototype.

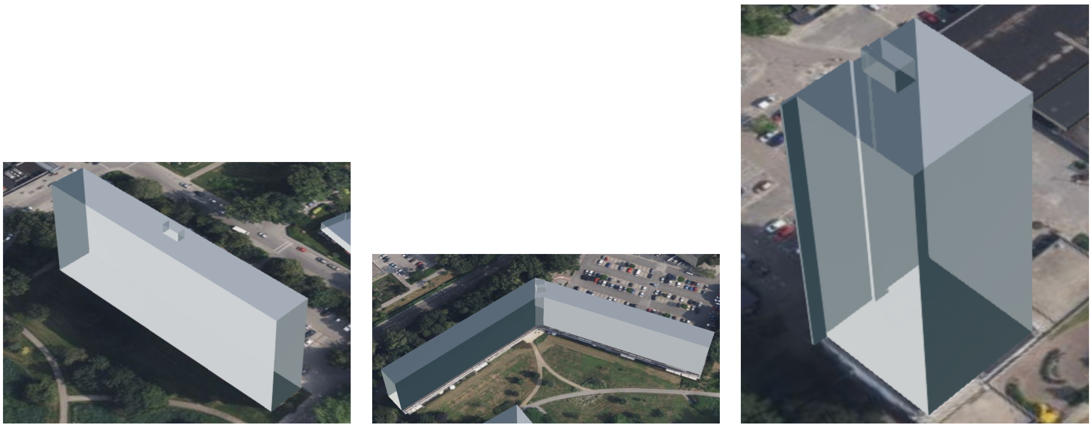
4.7. Algorithmic principles for partitioning 2-D plans

The subdivision of floor plans into apartment units represents one of the key algorithmic contributions of the prototype. Because detailed interior layouts are often unavailable in cadastral datasets, the system employs heuristic methods to generate plausible partitions from simplified geometric inputs. The algorithms are designed to handle three common building archetypes—*long-shaped*, *L-shaped*, and *square-shaped*—ensuring geometric validity, fairness in unit allocation, and computational robustness.

These three archetypes were selected for two main reasons. First, they exhibit relatively simple geometries that can be processed efficiently by the subdivision algorithm, enabling higher accuracy in the generated partitions. Second, these building forms were widely adopted in architectural design between the 1960s and 1990s in the Netherlands, making them representative of the prevalent residential typologies found in Dutch post-war urban housing.

The first attempt was a naïve linear partition: the interior space was divided into equal shares by the number of units, with cut segments whose lengths matched the short side of the building footprint. However, this strategy did not guarantee closed interior polygons; slight misalignments produced small gaps and slivers between segments—i.e., topological errors. As a result, the exported 2D DXF floor plans could not be reliably extruded into 3D models in downstream software. The current method was therefore developed to address these issues by enforcing closure and topological validity prior to export.

The pipeline begins with the parsing of the DXF model space. The external shell of the building is extracted from polylines stored on the "buildings" layer, while auxiliary guide



(a) A Long-Shaped Building (b) A L-Shaped Building (c) A Square-Shaped Building

Figure 4.5.: Three building archetypes based on the plan geometry

lines on other layers are interpreted as references for corridors, sides, or central axes. These entities are normalised into numeric arrays so that all downstream operations can be handled in a purely vector-based framework. This abstraction enables efficient computation and avoids CAD-specific dependencies.

Once the input geometry is structured, internal subdivision lines are generated. For each archetype, proportional division is applied to side or central guides, producing a set of interior points that mark the intended separation between units. These seed lines are then slightly extended or offset to guarantee intersections with the building shell. To ensure that all boundaries respect the legal footprint, every candidate line is clipped against the polygonal shell, with empty or fragmented results discarded. The outputs are ultimately written back into DXF format as a combination of the original shell and a set of new lines representing unit dividers and corridors.

A set of geometric utilities underpins these operations. Proportional division places interior points at fixed ratios along a segment, ensuring equal frontage allocation among units. Directional extension and offsetting shift endpoints along their unit vectors by small margins, preventing coincident geometry and guaranteeing intersections with the shell during clipping. Parallel translation allows guide lines to be repositioned toward corridors, while orthogonal projections and perpendiculars introduce corridor offsets or diagonal adjustments where required. Polygonal clipping finalises the process by constraining all boundaries strictly within the footprint.

On this foundation, three archetype-specific methodologies have been developed.

- In L-type plans, two side guides and a central corridor line define the two arms of the footprint. Offsetting these lines inward anchors the corridor, after which proportional division distributes units symmetrically between the arms. Partition lines connect corresponding stations on the inner and outer guides, producing a balanced subdivision around the corridor. Parts of the code is shown in the Figure 4.6.
- In Long-type plans, the two opposing side guides are offset to form a central corridor. Unit counts are divided evenly across the corridor bar, and proportional division gen-

```python
def updateStartL(data, num):
    sidePOi = []
    centerPoi = []
    lineList=[]
    dxfData=[]
    corridorList = []
    for item in data:
        if item['type'] == 'side':
            poi= []
            for point in item['points']:
                poi.append(point)
            sidePOi.append(poi)
        elif item['type'] == 'center':
            for point in item['points']:
                centerPoi.append(point)
        elif item['type'] == 'LWPOLYLINE':
            dxfData.append(item)
        elif item['type'] == 'corridor':
            pp = []
            for point in item['points']:
                pp.append((point[0], point[1]))
            corridorList.append(pp)
    a = (sidePOi[0][1][0], sidePOi[0][1][1])
    b = (sidePOi[0][0][0], sidePOi[0][0][1])
    c = (centerPoi[1][0], centerPoi[1][1])
    a1 = (sidePOi[1][1][0], sidePOi[1][1][1])
    b1 = (sidePOi[1][0][0], sidePOi[1][0][1])
    leftTopPOI=get_parallel_point(a, b, c)
    rightTopPOI = get_parallel_point(a1, b1, c)
    leftTopPoints=divide_line(sidePOi[0][0][0], sidePOi[0][0][1], leftTopPOI[0], leftTopPOI[1], num//2)
    leftBottomPoints = divide_line(sidePOi[0][1][0], sidePOi[0][1][1], centerPoi[1][0], centerPoi[1][1], num // 2)
    rightTopPoints = divide_line(sidePOi[1][0][0], sidePOi[1][0][1], rightTopPOI[0], rightTopPOI[1], num // 2)
    rightBottomPoints = divide_line(sidePOi[1][1][0], sidePOi[1][1][1], centerPoi[1][0], centerPoi[1][1], num // 2)
    for index, item in enumerate(leftTopPoints):
        lineList.append([item, leftBottomPoints[index]])
    for index, item in enumerate(rightTopPoints):
        lineList.append([item, rightBottomPoints[index]])
    lineObj = {}
    newLineList = []
    for value in lineList:
        newline = calculate_point(value[0], value[1], 'top', 2)
        interruptLine = clip_segment_by_polygon(newline, dxfData[0]['points'])
        newLineList.append(interruptLine)
    # print(newLineList)
    for item in corridorList:
        newLineList.append(item)
    lineObj['type'] = 'LINE'
    lineObj['points'] = newLineList
    dxfData.append(lineObj)
    return  dxfData
```

Figure 4.6.: A Part of 2D partition code for L-shaped type

erates transverse partitions between the two sides. The result is a ladder-like structure with evenly spaced units orthogonal to the corridor.

- In Square-type plans, diagonals drawn between the endpoints of the side guides establish a framework around a central intersection point. Auxiliary rails derived from perpendicular translations of these diagonals help manage the asymmetry of the footprint. Units are distributed across the two arms, with proportional division applied separately on each side. Additional short "spine" segments reinforce separation near the central junction, ensuring that the resulting subdivisions reflect the distinctive geometry of the square shape. A part of code shows in the Figure 4.7 how to calculate the diagonal intersection for square shaped buildings.

All corridors are specified with a clear width of 1.2 m, complying with the requirement that public corridors serving more than three dwellings must have a minimum width of 1.2 m, which is regulated in the Housing regulations in the Building Decree existing dwellings. Enforcing this constraint improves geometric validity and enables the 2D partitioning algorithm to succeed for the vast majority of floor plates.

Following the 2D partitioning, house numbers are assigned based on common Dutch addressing conventions. In general, numbering increases from east to west and from north to south along a street in Rotterdam. Apartments with house letters are placed adjacent to the corresponding unit with the same base number, and the letters follow alphabetical order.

Correctness is enforced by ensuring that exactly N–1 interior lines are generated for N units, with counts split appropriately when multiple arms are involved. Small offsets are introduced systematically to reduce degeneracies such as overlaps or tangencies, while clipping guarantees that no partition extends beyond the shell. Robustness is further supported through default fallbacks for degenerate cases, longest-segment selection for ambiguous intersections, and rounding of coordinates to a fixed precision.

From a geomatics perspective, the use of guidelines and proportional division provides a pragmatic approximation of cadastral subdivision in the absence of detailed floor plans. The algorithms reproduce equity in frontage and adjacency to corridors. Orthogonality and symmetry are enforced through parallel and perpendicular constructs, while polygonal clipping secures spatial legitimacy by keeping all generated partitions strictly within the legal boundary. Together, these methodological choices allow the prototype to derive meaningful unit boundaries that are computationally efficient, geometrically consistent, and aligned with cadastral correctness, even when operating with minimal input data.

4.8. Extrusion to 3D WOZ Object Units

The two-dimensional DXF floor plans generated by the subdivision function can be further transformed into volumetric representations of WOZ object units. This process is carried out in SketchUp, where the DXF partitions are imported and overlaid with georeferenced base maps in SKP format. These base maps, which can be obtained through online services such as Cadmapper, ensure that the floor plan geometry is spatially aligned with the urban context.

Within SketchUp, the extrusion tool is employed to extend the 2D partitions vertically into three-dimensional solids. The resulting models provide an approximate yet consistent representation of WOZ object units, capturing both their footprint and their volumetric extent.

```python
def line_intersection(entities_data, distance = 0.6):

    line1 = []
    line2 = []
    for index, item in enumerate(entities_data):
        if item['type'] == 'side' and index == 1:
            line1 = item['points']
        if item['type'] == 'side' and index == 2:
            line2 = item['points']

    A = line1[0]
    B = line1[1]
    C = line2[0]
    D = line2[1]

    diagonal1 = (A, D)
    diagonal2 = (B, C)

    (x1, y1), (x2, y2) = diagonal1
    (x3, y3), (x4, y4) = diagonal2

    den = (y4 - y3) * (x2 - x1) - (x4 - x3) * (y2 - y1)

    ua = ((x4 - x3) * (y1 - y3) - (y4 - y3) * (x1 - x3)) / den
    ub = ((x2 - x1) * (y1 - y3) - (y2 - y1) * (x1 - x3)) / den

    if 0 <= ua <= 1 and 0 <= ub <= 1:
        x = x1 + ua * (x2 - x1)
        y = y1 + ua * (y2 - y1)

        P1, D1 = get_perpendicular_point(A, B, (round(x, 4), round(y, 4)), distance)
        P2, D2 = get_perpendicular_point(C, D, (round(x, 4), round(y, 4)), distance)

        translation_vector1 = (P1[0] - D1[0], P1[1] - D1[1])
        translation_vector2 = (P2[0] - D2[0], P2[1] - D2[1])

        A_trans, B_trans = translate_segment(A, B, translation_vector1)
        C_trans, D_trans = translate_segment(C, D, translation_vector2)
        lineObj = {}
        lineObj['left'] = A_trans, B_trans
        lineObj['right'] = C_trans, D_trans
        return lineObj

    return None
```

Figure 4.7.: A Part of 2D partition code for Square-shaped type(calculating diagonal intersection)

Once extruded, the models can be exported directly to web-compatible formats using the Cesium publishing extension integrated in SketchUp. This extension embeds the necessary geographic reference, allowing the models to be seamlessly visualised within the web-based 3D viewer.

Through this workflow, the prototype bridges the gap between algorithmically derived 2D floor plans and interactive 3D representations. The extrusion process not only enhances the visual comprehension of valuation units but also prepares the models for integration into a broader geovisualisation environment, supporting both spatial analysis and dissemination of valuation information.

4.9. WOZ-Value Population

The WOZ-value dataset is publicly accessible through the national portal wozwaardeloket . While this platform provides authoritative valuation information, it does not offer functionality for bulk downloads. Given the scale of the study area, which comprises 7,765 WOZ objects, it was not feasible to manually retrieve the complete dataset at unit level. To address this limitation, an approximation strategy was adopted to capture valuation trends across the research area.

Specifically, a linear interpolation approach was applied within each building and floor. For every floor, the WOZ-values associated with the apartments holding the smallest and largest house numbers were manually recorded. These two anchor values were then used as boundary conditions to estimate the intermediate values, under the assumption that WOZ-values vary approximately linearly with respect to house number ordering. Although simplified, this method provides a pragmatic means of reconstructing valuation distributions at scale, allowing temporal and spatial patterns of WOZ-values to be analysed without requiring exhaustive manual collection.

4.10. LADM Part 4 alignment in the implementation

The system design has been guided by the principles and semantics established in the Land Administration Domain Model (LADM) Part 4, which provides a standardized framework for representing valuation information within land administration systems. Although the prototype is not a full implementation of the standard, it deliberately maps its data structures and outputs to key LADM Part 4 concepts, thereby demonstrating both conceptual alignment and the potential for further extension.

Within the database, the woz_objects table is modelled as the equivalent of the ValuationUnit class in LADM Part 4. Each record represents an apartment unit, carrying identifiers that link it to its parent building (pandid) and floor level (bwlg_vb0). This design allows each valuation object to be associated not only with its physical building context but also with its role in aggregated groupings, thereby mirroring the hierarchical structure foreseen in the valuation model.

The DXF generation pipeline further strengthens this alignment by producing geometric artefacts that can be interpreted as spatial representations of these valuation units. For each combination of building and floor, the system derives a dedicated DXF file that captures the two-dimensional floor subdivision. Although simplified to planimetric geometry, these files can be treated as proxies for the SpatialUnit class in LADM Kara et al. [2019], providing a geometric basis that connects valuation records with their physical footprint. This creates a direct link between valuation data and the spatial artefacts needed for visualization and further analysis.

The floors registry contributes an additional layer of integration by recording placement parameters such as geographic coordinates, elevation, and orientation. This registry allows each floor model to be positioned accurately within a three-dimensional environment, effectively embedding the valuation units into a georeferenced spatial framework. In LADM Part 4 terms, this corresponds to supporting the spatial realisation of valuation units, enabling them to be visualised and disseminated within a 3D cadastre context.

While the current prototype primarily focuses on static valuations and geometric approximations, the design leaves room for future extensions. In particular, temporal persistence and versioning could be incorporated into the schema to capture changes in property values across time, in line with the bitemporal mechanisms recommended in ISO 19152-4. Similarly, the inclusion of event histories would enable the system to record valuation adjustments, appeals, or re-assessments, further strengthening compliance with the standard.

Through these mappings, the prototype illustrates how a lightweight, web-based system can adhere to the conceptual framework of LADM Part 4. By embedding valuation units, spatial artifacts, and 3D placement into a coherent workflow, the implementation demonstrates both the feasibility and the practical benefits of aligning property valuation dissemination with international land administration standards.

5. Result

This chapter presents the outcomes of the developed system and the evaluation of its performance in visualising, processing, and analysing property valuation information in three dimensions. It begins by demonstrating the functionality of the web interface, which integrates hierarchical visualisation, temporal animation, and interactive querying of WOZ valuation data across the Building, Floor, and Apartment levels. Subsequently, the chapter reports the quantitative results of the 2D partitioning algorithm and its geometric accuracy assessment against reference floor plans obtained from the municipality. Together, these results validate the feasibility of linking valuation information with spatial building representations, supporting both exploratory analysis and future integration within an LADM Part 4–compliant 3D cadastre framework.

5.1. Web Interface

The web interface is implemented with Vue and CesiumJS and presents the study area on a satellite-imagery basemap with three hierarchical levels of detail: *Building*, *Floor*, and *Apartment*. Users interact with the 3D scene via mouse-based pan, orbit, and zoom. Level-of-detail toggles are provided as three buttons in the top-left corner; activating one level removes the other level's primitives from the scene, so only the models relevant to the active level are rendered. The camera state is not reset during level switches, so the viewpoint remains effectively constant while changing levels.

At the **Building** level, pre-published 3D tiles are loaded to provide an overview of the built form. Selecting a building triggers a right-hand information panel titled "Building Information" as shown in the Figure 5.1 , and the corresponding information listed in the Table 5.1.

At the **Floor** level, the panel "Floor Information" in the Figure 5.3ashows the corresponding information listed in the Table 5.1 as well. The listed WOZ objects are restricted to that floor. If a selection has no residential units, the panel displays a message to that effect and the timeline remains hidden.

At the **Apartment** level, the panel "Apartment Information" in the Figure 5.3bpresents the information in the Table 5.1. And Valuation fields are specific to the selected unit.

Table 5.1 summarises the fields rendered by the panels at each level. In the implementation, certain items are derived: "Number of Units" counts the WOZ objects returned for the current selection; "Floors" concatenates the minimum and maximum building floor indices; "WOZ-Value (Mean)" is the arithmetic mean of the selected units' values for the chosen year; and "Annual Increase in Value" is computed as the percent change relative to the previous year. Above attributes could be seen in the Figure 5.3

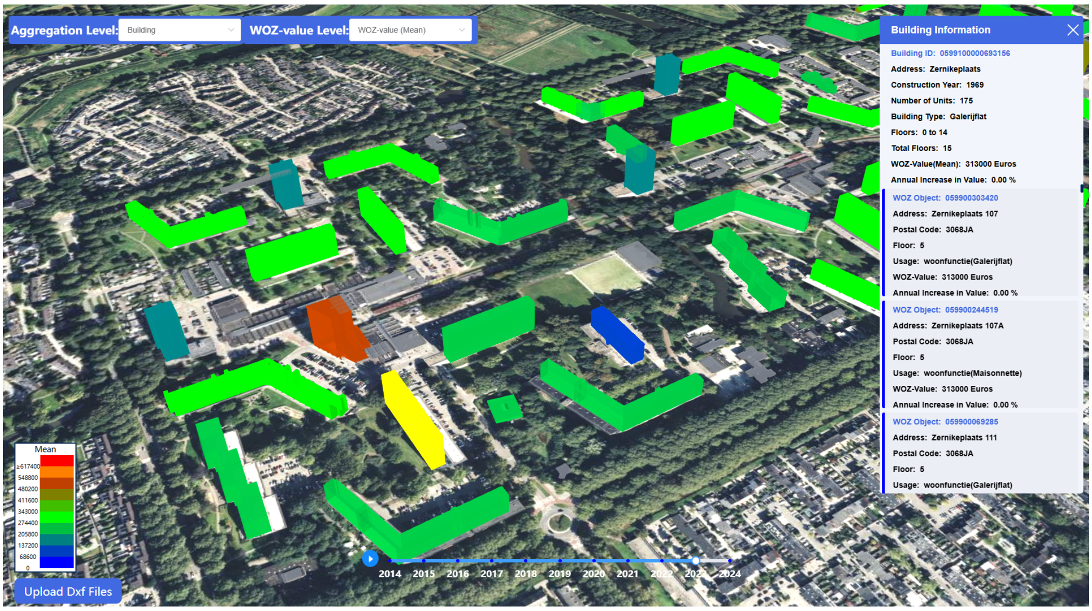
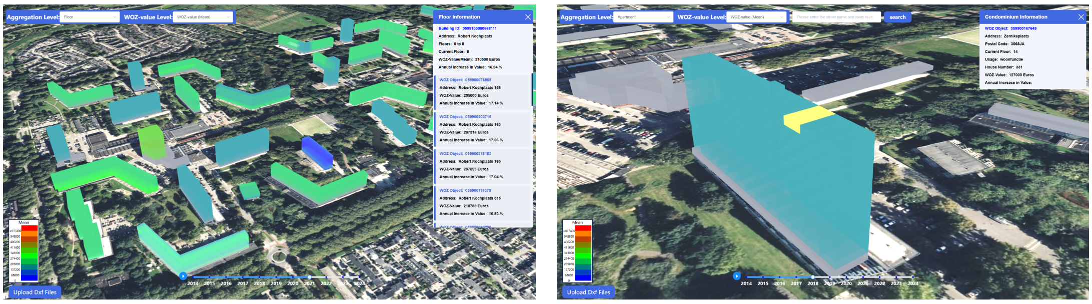


Figure 5.1.: Building level



(a) Floor level (b) Apartment level

Figure 5.2.: The web interface at different levels of detail: floor and apartment.

Temporal exploration. When any of the above selections is active, a timeline slider appears centered along the bottom edge of the interface. The slider spans calendar years 2014–2024 in integer steps. Moving the slider emits a year change event that triggers valuation queries against the backend for the selected WOZ object identifiers. For a building or floor selection, the interface computes: (i) the mean WOZ value across the selected units for the chosen year (displayed in Euros), and (ii) the year-over-year change (in %) relative to the previous year, shown only when the previous-year value is non-zero. For an individual apartment, the interface displays the unit's WOZ value and its year-over-year change.

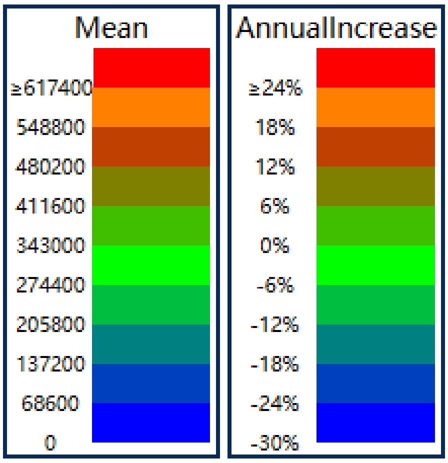
Visualization and Animation. A dynamic colour scheme comprising ten continuously varying hues was designed to visualise valuation attributes across multiple aggregation levels, as illustrated in Figure 5.3. Within the web interface, users can flexibly switch between different aggregation levels (Building, Floor, and Apartment), select WOZ-value indicators as

Table 5.1.: Attributes at building, floor, and condominium levels

Level	Field	Description / Example
Building		
	Building ID	Unique identifier
	Address	Street + number, city
	Construction Year	Year built
	Number of Units	Count of dwelling units
	Building Type	e.g., Gallery flat
	Floors	e.g., 0 to 8
	Total Floors	e.g., 9
	WOZ-Value (Mean)	Mean assessed value
	Annual Increase in Value	Year-over-year change
	WOZ Objects	All the WOZ objects that are in the building
Floor		
	Building ID	Parent building identifier
	Address	Street + number, city
	Floors	e.g., 0 to 8
	Current Floor	e.g., 1
	WOZ-Value (Mean)	Mean assessed value on this floor
	Annual Increase in Value	Year-over-year change
	WOZ Objects	All the WOZ objects that are on the floor
Condominium		
	Condominium ID	Unique identifier
	Address	Street + number, city
	House Number	e.g., 385
	Postal Code	e.g., 3069EK
	Floor	Floor index
	Usage	e.g., *woonfunctie*
	WOZ-Value	Assessed value
	Annual Increase in Value	Year-over-year change

described in Table 5.2, and explore changes over time through the timeline control. The colour-coded rendering enables intuitive interpretation of valuation information, facilitating rapid identification of value patterns and anomalies across scales.

In addition, an animation function located beside the timeline automatically visualises the temporal evolution of the selected attribute. By activating the play button, users can observe year-over-year trends of the chosen indicator under the active aggregation level, revealing spatial and temporal dynamics over the past decade. To further enhance user interaction, an address-based search tool is provided at the Apartment level, allowing users to locate a specific apartment by entering its address. This combination of visualisation, temporal animation, and interactive search offers a comprehensive and engaging means to analyse and disseminate WOZ valuation information within a 3D environment. A few screenshots(Figure 5.4) shows how Annual Increase changes in Floor level in the research field.



(a) Mean value (b) Annual increase

Figure 5.3.: The legends of visualization across different levels

Table 5.2.: Description of WOZ-value indicators used in visualization

WOZ-value Level	Description
WOZ-value (Mean)	The average WOZ-value for all apartments under the chosen aggregation level.
WOZ-value (Highest)	The highest WOZ-value for all apartments under the chosen aggregation level.
WOZ-value (Lowest)	The lowest WOZ-value for all apartments under the chosen aggregation level.
Annual Increase	Year-over-year change, computed based on the chosen year and the previous year.

DXF upload and floor subdivision. An *Upload Dxf Files* button in the bottom-left opens a dialog titled "Generate DXF files". The dialog accepts a single `.dxf` footprint file and requires the user to choose a building archetype from {L, Long, Square}. Upon clicking *Start Processing*, the interface displays a non-blocking overlay "The file is being processed......", then uploads the file, the inferred building identifier (parsed from the filename), and the selected archetype to the server. On success, the interface reports the total number of generated files. The server writes floor-by-floor 2D subdivision plans to its uploads directory; these plans are topologically closed, which are shown in the Figure 5.5. And base maps in SketchUp format could be downloaded at the website cadmapper with reference system embedded, so users do not need to adjust the reference manually, so models can be directly imported into 3D modeling tools (e.g., SketchUp) extruded into interior 3D models, as shown in the Figure 5.6. Because the plans retain the building/floor identifiers (WOZ object numbers and floor indices), the models can be published to Cesium with consistent identifiers (automatically generated and embedded in models during publishing) for integration with the viewer.

(a) 2015

(b) 2018

(c) 2021

(d) 2024

Figure 5.4.: Annual increase on the floor level in different years

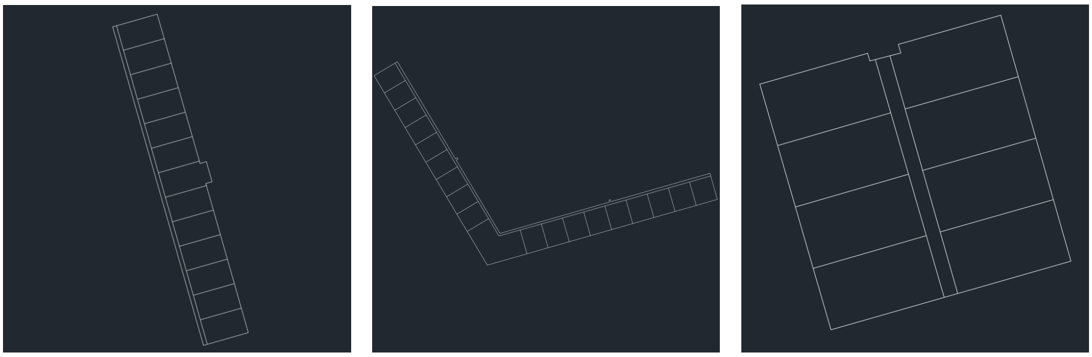
Interaction design. The three panels (building, floor, condominium) are reactive: opening a panel shows the timeline, changing the year updates monetary values and percentages in-place, and closing a panel clears the selection and hides the timeline. The overall design keeps the scene stable during level switches, minimizes occlusion by rendering only one level at a time, and foregrounds the identifiers and valuation attributes necessary for exploratory analysis.

WOZ-value visualization The WOZ-value will be shown in different colors by the different ranges across temporal and spatial levels. Users could filter the colors by their needs, for examples, the average, lowest or highest value. The work will be completed between P4 and P5.

5.2. Accuracy Evaluation

5.2.1. Evaluation Worklfow

The accuracy evaluation was conducted to assess how well the proposed 2D partitioning algorithm approximates the actual layout of apartments in multi-storey buildings. Since the algorithm was designed to work with minimal input data, it is essential to benchmark its results against authoritative reference material. To ensure representativeness, one building from each of the three implemented archetypes—Long-shaped, L-shaped, and Square-



(a) Long-Shape Partition (b) L-Shape Partition (c) Square-Shape Partition

Figure 5.5.: Partitioning results for three building archetypes: Long, L, and Square-shaped.

(a) SketchUp base map (b) Base map with imported plan (c) Extruded model

Figure 5.6.: The workflow of model extrusion

shaped—was selected for detailed evaluation. These case studies collectively cover the main structural configurations observed in the study area.

The evaluation followed a four-step workflow:

1. **Ground-truth construction.** The original 2D plan drawings, obtained from the municipal building-permit archive in PDF format, were imported into AutoCAD and vectorised to generate reference floor plans (Figure 5.7). These reference plans were treated as the ground truth for comparison.

2. **Overlay with predicted partitions.** The algorithmically generated DXF outputs were imported into AutoCAD and carefully aligned with the ground-truth plans, allowing a direct visual and quantitative comparison between predicted and actual apartment boundaries.

3. **Area-based overlap assessment.** For each floor, the area of residential units was computed from the ground-truth plans. As shown in Figure 5.8, these areas are shaded with diagonal hatching. The predicted partitions were then overlaid, and the correctly

matched areas—where predicted and reference polygons coincided—were calculated. These overlapping regions are highlighted in white in Figure 5.9.

4. **Computation of accuracy rate.** For every building, the correctly predicted area was divided by the total ground-truth residential area across all floors. The resulting ratio, expressed as a percentage, was adopted as the *accuracy rate* of the partitioning algorithm. This metric captures how effectively the algorithm preserved the size and arrangement of legal units in the absence of detailed floor plans.

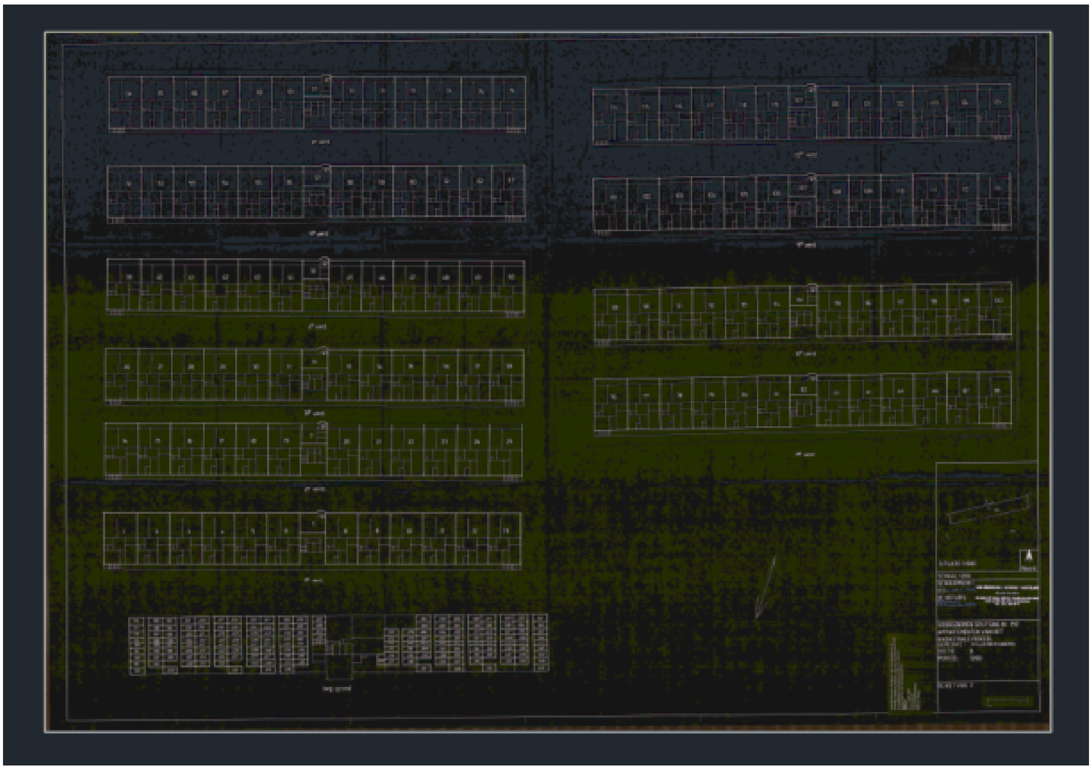


Figure 5.7.: Imported ground-truth 2D floor plans

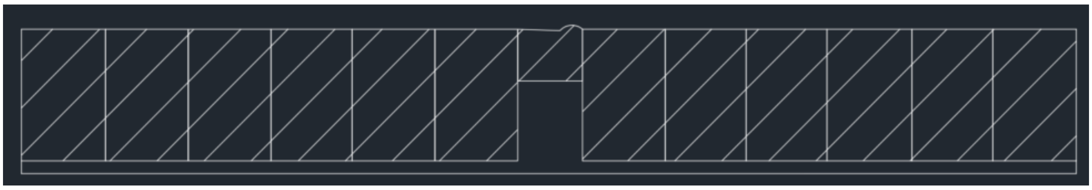


Figure 5.8.: Ground-truth plan for the 1st floor, with residential units shaded

This evaluation provides a quantitative measure of geometric fidelity while also highlighting the spatial distribution of residual errors. By repeating the procedure across multiple archetypes, the robustness and generalizability of the algorithm can be assessed. In line with prior studies on 3D cadastre approximation (e.g. Nottrot et al., 2023), the chosen metric reflects both geometric validity and thematic correctness, since misaligned partitions directly affect the accuracy of valuation unit geometric boundaries. The reported accuracy rates,

Figure 5.9.: Overlay of predicted partitions and ground-truth units for the 1st floor

therefore, serve as a benchmark for the applicability of the algorithm in operational settings, where a balance between computational efficiency and cadastral correctness must be achieved.

5.2.2. Evaluation Results

To assess the accuracy of the proposed 2D plan partitioning algorithm, representative buildings from two different archetypes were selected: a Long-shaped building and an L-shaped building. These case studies allow for testing the algorithm under different geometric conditions while ensuring that the sample is representative of the building typologies commonly found in the study area. For each case, the algorithmically generated floor plans were compared against the ground-truth layouts obtained from municipal PDF floor plans, which were vectorised as described in Section 3.3. The evaluation metric is defined as the ratio between the overlapped residential floor area and the ground-truth residential floor area, expressed as a percentage for each floor and averaged across the building. The validation process was carried out manually using AutoDesk software.

Table 5.3.: Accuracy evaluation of 2D partitioning (Long-shaped building)

Floor	Ground Truth Size (m^2)	Overlapped Size (m^2)	Correct Prediction (%)
1	1209.90	1053.73	87.09
2	1209.90	938.33	77.55
3	1209.90	1065.61	88.07
4	1209.90	939.85	77.68
5	1209.90	942.97	77.94
6	1209.90	1061.92	87.77
7	1209.90	1061.92	87.77
8	1209.90	939.85	77.68
9	1209.90	939.85	77.68
10	1209.90	1061.92	87.77
11	1209.90	1061.92	87.77
12	1209.90	942.97	77.94
13	1209.90	1065.61	88.07
14	1209.90	939.85	77.68
Total	**16938.60**	**14016.30**	**82.75**

Long-shaped building. 3 Long-shaped residential buildings (Building ID are 0599100000633396, 0599100000603801 and 0599100000702511) are chosen as the test cases. The result of building ID 0599100000633396 is presented in the Table 5.3. It consisting of 175 apartment units distributed over 14 floors. The correct prediction per floor ranges between 77.55% and 88.07%, with most floors achieving values above 85%. The overall correct prediction for the entire building is 82.75%, which demonstrates that the algorithm performs reliably for elongated footprints where unit layouts follow a highly regular corridor-based distribution. Occasional drops in accuracy (e.g., Floors 2, 4, 8, and 9) are associated with irregularities in the ground-truth floor plan, such as stairwell offsets or variations in apartment sizes, which are not fully captured by the heuristic partitioning rules.

Table 5.4.: Matching results between predicted and reference units

	Total	Good matched	Partial matched	Not matched
Count	514	410	74	30
Percentage	–	79.76%	14.40%	5.84%

In the evaluation, a total of 514 apartment units were processed. The total correct prediction is 68.67%. Based on the proportion of correctly predicted boundaries, three outcome categories were defined: *good matched* (correct prediction $> 50\%$), *partial matched* (correct prediction $< 50\%$), and *not matched* (correct prediction $= 0\%$). The overall results of the classification are summarised in Table 5.4.

L-shaped building. 3 L-shaped residential buildings (Building ID are 0599100000613750, 0599100000655189 and 0599100000763765) are chosen as the test cases, in which the result of building ID 0599100000613750 are presented in Table 5.5. It accommodates 176 apartment units across 9 floors. In this case, the accuracy values show a much larger variance between floors. The ground floor (Floor 0) achieved only 24.82% correct prediction due to a markedly irregular layout with commercial and service functions, which the generic partitioning algorithm is not designed to model. From the first floor upwards, however, the correct prediction stabilises around 66.3%, reflecting the algorithm's ability to approximate regularised apartment layouts despite the geometric complexity introduced by the L-shape. The overall building correct prediction is 61.75%, which, while lower than the Long-shaped case, still demonstrates the feasibility of applying the algorithm to more complex building footprints. Nevertheless, the reduced performance highlights the need for additional heuristics to handle corner effects, asymmetries, and non-residential ground floors.

In the evaluation, 406 apartment units were processed. The total correct prediction is 63.57%. The matching results between predicted and reference units is presented in the Table 5.6.

Discussion. The evaluation confirms that the algorithm performs well for Long-shaped buildings, achieving an overall accuracy above 80%, which is sufficient for approximating apartment layouts in valuation workflows. However, the results for the L-shaped building underline the challenges posed by irregular geometries. In particular, mixed-use ground floors and asymmetric wings reduce the reliability of the heuristic subdivision. These findings suggest that while the current method provides a promising first step towards automated partitioning, further refinements—such as corner-specific rules, floor-type differen-

Table 5.5.: Accuracy evaluation of 2D partitioning (L-shaped building)

Floor	Ground Truth Size (m^2)	Overlapped Size (m^2)	Correct Prediction (%)
0	1801.88	447.16	24.82
1	1810.42	1200.09	66.29
2	1810.42	1201.36	66.36
3	1810.42	1201.36	66.36
4	1810.42	1201.36	66.36
5	1810.42	1201.36	66.36
6	1810.42	1201.36	66.36
7	1810.42	1201.36	66.36
8	1810.42	1201.50	66.37
Total	**16285.20**	**10056.90**	**61.75**

Table 5.6.: Matching results between predicted and reference units

	Total	Good matched	Partial matched	Not matched
Count	406	3	84	39
Percentage	–	76.07%	16.34%	7.59%

tiation, or machine learning approaches—would be necessary to generalise the approach across more complex building typologies.

A Square-shaped building floor plans are not available, so unfortunately the accuracy test cannot be conducted.

6. Conclusion and Future Work

This chapter summarises the main outcomes of the research and reflects on how the developed methods address the research objectives. It revisits the central and sub-research questions, providing concise answers supported by the findings from the methodology and implementation chapters. Furthermore, it outlines potential directions for future work that could extend the system's functionality and applicability, and concludes with recommendations for key stakeholders such as the Waarderingskamer, Rotterdam Municipality, and the ISO/LADM working group. Together, these reflections highlight both the scientific contributions and the practical implications of establishing a 3D valuation information framework based on LADM Part 4.

6.1. Answers of Research Questions

The central research question of this thesis was: *How can WOZ valuation information—including the boundaries of residential properties and their corresponding dates and values—be effectively modelled and visualised in three dimensions across spatial scales and time, within an LADM Part 4–compliant framework?*

The findings of this research collectively demonstrate that such integration is both technically feasible and conceptually consistent with the data structures and semantics defined in LADM Part 4. By developing a modular workflow that includes 2D partitioning, data integration, temporal querying, and interactive 3D visualisation, the research provides a complete proof-of-concept pipeline for representing and disseminating valuation information across multiple spatial and temporal scales. The following subsections address each sub-question in turn.

1. Approximating three-dimensional boundaries of WOZ units

Chapter 3 and Chapter 5 described the design and evaluation of a 2D partitioning algorithm that approximates the internal subdivision of multi-storey buildings into apartment units using minimal publicly available cadastral data. The method combines geometric heuristics based on building archetypes (Long-, L-, and Square-shaped) with a rule-based segmentation approach, producing closed 2D polygons that represent apartment footprints. These plans can subsequently be extruded into 3D solids for integration with CesiumJS as 3D Tiles. Accuracy assessments presented in Section 5.2 show that the algorithm achieves 82.75% correctness for long-shaped buildings and 61.75% for L-shaped buildings, confirming that the approach provides a reliable geometric approximation when detailed floor plans are unavailable.

2. Linking WOZ information with 3D building units

The integration strategy discussed in Section 4.3 and implemented in Section 4.4 demonstrates how WOZ object identifiers can be systematically linked to BAG and BGT geometries within a PostgreSQL database. Each valuation unit (WOZ-object) is assigned a unique identifier that connects it to its corresponding building, floor, and apartment levels through hierarchical relationships reflecting the LADM Part 4 class structure. This design enables semantic consistency across spatial units and supports extensibility towards future cadastral and valuation data models. The implemented numbering and linkage mechanism therefore ensures that valuation records can be accurately associated with their spatial representations in both 2D and 3D.

3. Extracting and populating WOZ values

Section 4.9 outlines the extraction and transformation process through which WOZ valuation data were cleaned, standardised, and populated into the 3D database environment. Temporal WOZ values from 2014–2024 were mapped to their respective object identifiers, allowing annual valuations to be queried dynamically. Derived indicators such as mean value, highest and lowest value, and annual increase were computed in the backend and visualised interactively in the web interface (Section 4.4). This systematic population and temporal structuring of valuation data enable longitudinal analysis and comparison across time.

4. Aggregation and interactive visualisation

The prototype visualisation platform described in Chapter **??** demonstrates that hierarchical and temporal visualisation of valuation information can be achieved within a web-based 3D environment. Built using Vue and CesiumJS, the interface allows users to explore valuation information at the building, floor, and apartment levels, switch between different aggregation modes, and view changes over time using an interactive timeline. The implementation further includes colour-coded thematic rendering, automated animation of valuation trends, and an address-based search function, all of which enhance usability and analytical potential. Together, these elements prove that dynamic, multi-scale visualisation of WOZ valuation information can be effectively realised through web-native 3D GIS technologies.

Summary

In summary, the developed approach demonstrates a coherent workflow that bridges cadastral geometry, valuation semantics, and temporal visualisation within an LADM Part 4–compliant framework. The integration of rule-based boundary approximation, structured data linking, temporal attribute management, and interactive 3D rendering collectively answers the central research question. The thesis thereby establishes a foundation for further development of 3D valuation cadastres in the Netherlands, where accurate spatial representation and transparent communication of property values are essential for public administration and spatial planning.

6.2. Future Work

The prototype intentionally prioritized minimal inputs and common plan archetypes to demonstrate feasibility. Several extensions would increase coverage and operational value:

1. **Inclusion of sub-surface residential property.** The current workflow targets above-ground residential space only. Future work should incorporate basements and other underground components (e.g., storage rooms and underground parking) into both geometry and valuation pipelines. Methodologically, this requires (i) explicit handling of vertical datums and below-terrain solids (NAP/ellipsoidal alignment), (ii) robust terrain intersection tests to avoid shell overlaps, and (iii) viewer support for underground rendering (clipping planes, occlusion control). Where open data are insufficient, permit archives or BIM/IFC models can supply depths and partition logic for sub-surface units.

2. **Composite (multi-space) properties.** At present, the system assumes a *single-space* valuation unit (e.g., an apartment). In practice, many dwellings comprise *accessory spaces*—such as an assigned parking bay or storage box—whose value should be linked to, and potentially apportioned with, the principal unit. Future work should model *composite valuation units* with explicit parent–child relations (apartment $\rightarrow$ {parking, storage}), define rules for value allocation (e.g., fixed shares, €/m^2 weighting, or assessor-provided coefficients), and expose these linkages in the UI (filtering and inclusion/exclusion toggles). On the data side, the persistence layer should support many-to-one associations between identifiers and geometry, and carry provenance for auditability.

3. **Generalising the 2D partition algorithm beyond three archetypes.** The partitioner currently handles *Long*, *L*, and *Square* footprints. To cope with *courtyard blocks, U-/C-shapes, tower cores, atria, and non-orthogonal/curvilinear plans*, the algorithm should be extended with more generic primitives: corridor inference via straight skeletons or medial axes; constrained Delaunay or rectilinear decomposition for complex shells; and rule-based *guide-graph* construction for multi-arm floor plates. A fallback learning component (e.g., explainable boosted trees) could rank candidate partitions under weak supervision, while quality controls (closure, minimum width, aspect ratio, corridor continuity) gate outputs before export.

4. **Automated detection of building types.** The current implementation of the 2D partitioning algorithm requires users to manually select the building archetype (*Long-Shaped*, *L-shaped*, or *Square-Shaped*) prior to subdivision. Future development could incorporate an automated classification module capable of detecting building types directly from the input footprint geometry. By applying geometric feature extraction techniques—such as shape descriptors, convexity ratios, and orientation analysis—or using supervised machine learning on labelled building samples, the system could automatically infer the most appropriate archetype. This enhancement would eliminate manual intervention, improve workflow efficiency, and increase the robustness of the partitioning process for large-scale applications.

6.3. Recommendations

Throughout the course of this research, several recommendations have emerged that may assist both institutional stakeholders and standardisation bodies in improving the management, dissemination, and modelling of property valuation information.

To the Waarderingskamer

The research highlights the potential of three-dimensional and temporal visualisation for enhancing the transparency and accessibility of WOZ valuation information. It is therefore recommended that the Waarderingskamer consider promoting the publication of 3D-ready valuation data, including geometry references and time series attributes, through open data channels. Standardised access to valuation information in structured, machine-readable formats (e.g., via APIs) would significantly facilitate research, validation, and reuse across municipalities. Furthermore, the inclusion of explicit identifiers linking WOZ objects to BAG and cadastral units would improve semantic consistency and reduce redundancy in valuation workflows.

To the Rotterdam Municipality

For the Rotterdam Municipality, the study recommends continued integration of cadastral, building, and valuation datasets within a unified 3D spatial data infrastructure. The current municipal 3D viewer already provides a valuable foundation for public communication of spatial information, yet further enhancements—such as finer subdivision of multi-storey buildings and apartment-level WOZ integration—would enable more detailed urban and fiscal analyses. Automating the exchange between cadastral and valuation registers, and embedding the 3D visualisation component within the city's digital twin initiative, would also improve internal efficiency and public engagement. Lastly, establishing a feedback channel for citizens to report valuation inconsistencies directly through the 3D interface could strengthen transparency and trust in municipal assessments.

Summary

In summary, the recommendations emphasise the importance of aligning data structures, visualisation tools, and legal frameworks towards a more integrated and transparent valuation ecosystem. By strengthening the connection between national valuation practices and international standards, both public authorities and citizens stand to benefit from improved data consistency, analytical capability, and clarity in property valuation.

A. appendix

Table A.1.: Evaluation of the visualization systems, required features and their corresponding visualization-workflow layer [Shojaei et al., 2013]

Visualization workflow layer	**Features**	**Visualization systems**			
		Google Earth	ArcGlobe	NASA World Wind	TerraExplorer Viewer
Presentation layer	*Cadastral features*				
	Handling massive data	Yes (network links)	Yes (caching)	Yes	Yes
	Result of functions and queries	Yes (only search)	Yes	Yes (only search)	Yes
	Underground view	No	Yes	No	Yes
	Cross-section view	No	No	No	No
	Measurements (3D)	Yes	Yes	Yes	Yes
	Non-spatial data visualization	Yes	Yes	Yes	Yes
	Visualization features				
	Interactivity	Yes	Yes	Yes	Yes
	Levels of detail	Yes	Yes	Yes	Yes
	Symbols	Yes	Yes	Yes	Yes
	Color, thickness, line-style	Yes	Yes	Yes	Yes
	Labeling	Yes	Yes	Yes	Yes
	Transparency	Yes	Yes	Yes	Yes
	Tooltips	Yes	Yes	Yes	Yes
Application layer	*Non-functional features*				
	Technical diversity	Yes	Yes	Yes	Yes
	System integration & interoperability	Yes (extendable)	Yes (extendable)	Yes (extendable)	Yes (extendable)
	Usability	High (no expertise)	Low (expertise)	Medium	Medium
	Platform independence	Windows, Mac OS X, Linux, Android	Windows	Platform-independent (Java-based)	Windows
	Cost	Freeware	Proprietary	Open-source	Proprietary
	Web-based 3D visualization	Yes	No	Yes	Yes
Presentation layer	*Other features*				
	Profiling	No	Yes	Yes	Yes
	Shadow analysis	No (just shadow)	Yes	No (just shadow)	Yes
	Animation creation	Yes	Yes	Yes	Yes
	Line of sight & visibility analysis	No	Yes	Yes	Yes
	Skyline creation	No	Yes	No	No
	Texture mapping	Yes	Yes	Yes	Yes
	Aerial and satellite images	Yes	Yes	Yes	Yes
	3D updating and manipulating	No	Yes (rotate, scale, shift)	Yes	Yes

Bibliography

Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4 (4):2842–2889, 2015. doi: 10.3390/ijgi4042842.

Filip Biljecki, Hugo Ledoux, and Jantien Stoter. An improved lod specification for 3d building models. *Computers, Environment and Urban Systems*, 59:25–37, 2016. ISSN 0198-9715. doi: https://doi.org/10.1016/j.compenvurbsys.2016.04.005. URL https://www.sciencedirect.com/science/article/pii/S0198971516300436.

EPSG: Geodetic Parameter Registry. Amersfoort / rd new – netherlands (epsg:28992). https://epsg.io/28992, 2023a. Accessed: 2025-10-20.

EPSG: Geodetic Parameter Registry. Wgs 84 – world geodetic system (epsg:4326). https://epsg.io/4326, 2023b. Accessed: 2025-10-20.

Martin Goetz. Towards generating highly detailed 3d citygml models from openstreetmap. *International Journal of Geographical Information Science*, 27(5):845–865, 2013. doi: 10.1080/13658816.2012.721552.

Government of the Netherlands. General Administrative Law Act (Algemene wet bestuursrecht, Awb). https://wetten.overheid.nl/BWBR0005537/, 1994. Act of 4 June 1992, in force since 1 January 1994. Published by the Government of the Netherlands.

Gerhard Gröger and Lutz Plümer. Citygml – interoperable semantic 3d city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 2012. doi: 10.1016/j.isprsjprs.2012.04.004.

Dieter Hildebrandt and Jürgen Döllner. Service-oriented, standards-based 3d geovisualization: Potential and challenges. *Computers, Environment and Urban Systems*, 34(6):484–495, 2010. ISSN 0198-9715. doi: https://doi.org/10.1016/j.compenvurbsys.2010.05.003. URL https://www.sciencedirect.com/science/article/pii/S0198971510000426. GeoVisualization and the Digital City.

ISO/TC 211. Geographic information — land administration domain model (ladm) — part 4: Valuation information. Draft International Standard ISO/DIS 19152-4, International Organization for Standardization, 2024. Draft International Standard.

Abdullah Kara, M. van Oosterom, Peter J. Ruud Kathmann, Azer Ilgar, and Christiaan Lemmen. LADM Valuation Information Model Compliant Prototype for Visualisation and Dissemination of 3D Valuation Units and Groups. In *Proceedings of the 10th International FIG Workshop on the Land Administration Domain Model (LADM)*, pages 97–111, Dubrovnik, Croatia. International Federation of Surveyors (FIG). ISBN 978-87-92853-62-2. doi: 10.4233/uuid:606f4a8f-88c5-456a-b709-7f8262a59271. URL http://resolver.tudelft.nl/uuid:606f4a8f-88c5-456a-b709-7f8262a59271. Workshop held 31 March – 2 April 2022.

Abdullah Kara, Ruud Kathmann, Peter van Oosterom, Christiaan Lemmen, and Ümit Işıkdağ. Towards the netherlands LADM valuation information model country profile. In *Proceedings of the 8th International FIG Workshop on the Land Administration Domain Model (LADM 2019)*, Kuching, Malaysia, 2019. FIG – International Federation of Surveyors. URL https://repository.tudelft.nl/islandora/object/uuid:8c6ec7a3-1b6e-4d0b-92c4-4d2165fa4f40. Conference Paper.

Abdullah Kara, Peter van Oosterom, Volkan Çağdaş, Ümit Işıkdağ, and Christiaan Lemmen. 3 dimensional data research for property valuation in the context of the ladm valuation information model. *Land Use Policy*, 98:104179, 2020. ISSN 0264-8377. doi: https://doi.org/10.1016/j.landusepol.2019.104179. URL https://www.sciencedirect.com/science/article/pii/S0264837719306635. 3D Land Administration for 3D Land Uses.

Abdullah Kara, M. van Oosterom, Peter J. Ruud Kathmann, and Christiaan Lemmen. Requirements and opportunities for web-based 3d visualization and dissemination of property valuation information. In Eftychia Kalogianni, Alias Abdul-Rahman, and Peter van Oosterom, editors, *Proceedings of the 7th International FIG Workshop on 3D Cadastres (Virtual/online event)*, pages 113–128, New York, United States, October 2021. International Federation of Surveyors (FIG). ISBN 978-87-92853-48-6. doi: 10.4233/uuid:7a207ba9-7c2d-4e6d-98e0-1a0cd2363c90. URL http://resolver.tudelft.nl/uuid:7a207ba9-7c2d-4e6d-98e0-1a0cd2363c90. Workshop held 11–13 Oct 2021.

Abdullah Kara, Peter van Oosterom, Ruud Kathmann, and Christiaan Lemmen. Visualisation and dissemination of 3d valuation units and groups – an ladm valuation information compliant prototype. *Land Use Policy*, 132:106829, 2023. ISSN 0264-8377. doi: https://doi.org/10.1016/j.landusepol.2023.106829. URL https://www.sciencedirect.com/science/article/pii/S0264837723002958.

Jürg Kaufmann and Daniel Steudler. Cadastre 2014: A vision for a future cadastral system. Fig publication, International Federation of Surveyors (FIG), 1998. URL https://www.fig.net/resources/publications/figpub/cadastre2014/index.asp. Retrieved from FIG Publication.

Tatjana Kutzner, Kanishk Chaturvedi, and Thomas H. Kolbe. Citygml 3.0: New functions open up new applications. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1):43–61, 2020. doi: 10.1007/s41064-020-00095-z.

Jiyeong Lee. 3d gis for geo-coding human activity in micro-scale urban environments. In Max J. Egenhofer, Christian Freksa, and Harvey J. Miller, editors, *Geographic Information Science*, pages 162–178, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

Jiyeong Lee. Gis-based geocoding methods for area-based addresses and 3d addresses in urban areas. *Environment and Planning B: Planning and Design*, 36:86–106, 01 2009. doi: 10.1068/b31169.

Bas Nottrot, Erwin Folmer, Debraj Roy, Bob Scheer, and Peter Merx. Multi-unit building address geocoding: An approach without indoor location reference data. *Transactions in GIS*, 27(1):57–83, 2023. doi: https://doi.org/10.1111/tgis.13017. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.13017.

Open Geospatial Consortium (OGC). Citygml standard – city geography markup language. https://www.ogc.org/standards/citygml/, 2021. Accessed: 2025-10-20.

Davood Shojaei, Mohsen Kalantari, Ian D. Bishop, Abbas Rajabifard, and Ali Aien. Visualization requirements for 3d cadastral systems. *Computers, Environment and Urban Systems*, 41: 39–54, 2013. ISSN 0198-9715. doi: https://doi.org/10.1016/j.compenvurbsys.2013.04.003. URL https://www.sciencedirect.com/science/article/pii/S0198971513000422.

Jantien Stoter, Serene Ho, and Filip Biljecki. Considerations for a contemporary 3d cadastre for our times. In *Proceedings of the 7th International FIG Workshop on 3D Cadastres*, pages 1–20, Seoul, Republic of Korea, 2019. International Federation of Surveyors (FIG). URL https://ual.sg/publication/2019-considerations-3-d-cadastre/2019-considerations-3-d-cadastre.pdf.

Waarderingskamer – Netherlands Council for Real Estate Assessment. Explanation of woz value. https://www.waarderingskamer.nl/en/for-residents/explanation-woz-value?utm_source=chatgpt.com, 2025. Accessed: 2025-10-20.

Colophon

This document was typeset using $\LaTeX$, using the KOMA-Script class `scrbook`. The main font is Palatino.

TU Delft