

Open Source 3D Cadastre Visualisation Pipeline

Pavan Sai Goud Goddu¹, Sisi Zlatanova², Mohsen Kalantari^{1*}

¹ School of Civil and Environmental Engineering, University of New South Wales (UNSW), Sydney, Australia - goddupavansaigoud@gmail.com; mohsen.kalantari@unsw.edu.au

² School of Built Environment, University of New South Wales (UNSW), Sydney, Australia - s.zlatanova@unsw.edu.au

Keywords: 3D cadastre; strata; web visualisation; JSON; JSON-LD; 3D CSDM

Abstract

Interpreting multi-storey property rights is difficult when information is scattered across 2D plans and text or locked inside desktop projects. We present a web-based pathway that communicates strata lots and common property consistently across levels in a standard browser. Aligned with the 3D Cadastral Survey Data Model and Exchange (3D CSDM) of Australia, we propose an open-source, web-first approach. The method couples a lightweight browser viewer (level/tenure filters, plan overlay, search, readable legend) with an explicit conversion step that standardises common GIS inputs into a fixed core JSON profile, with limited official CSDM-aligned JSON-LD hooks applied only to selected keys that have exact matches in the published vocabularies, while all remaining source attributes are preserved as standard JSON fields. Using a New South Wales case study, we evaluated the viewer against ISO 9241-11 criteria (effectiveness, efficiency). Across repeated trials (cache disabled/enabled), mean page-open times were 0.60 s (Chrome) and 1.48 s (Edge); interaction averaged 50–60 FPS; level filters applied in 40–55 ms; all five tasks succeeded. Practically, this delivers fast, consistent 3D communication of lots and common property without installs, lowering access barriers for agencies and owners while aligning with 3D CSDM's web-first direction. Next, we will finalise viewer parity between Upload-and-View and the Reference Viewer and add a light in-viewer validation panel.

1. Introduction

Understanding multi-storey property rights is difficult when the information is split across 2D plans and text. Practitioners and owners need a clear way to see which volumes are privately owned and which are shared, and how these spaces relate across levels (van Oosterom et al., 2018). Such distinctions are also important for understanding circulation, access, and evacuation within buildings (Alattas et al., 2017). Recent reviews group current 3D cadastral visualisation into three options: interactive viewers, 3D PDF, and augmented reality (AR) (Dudhee and Vukovic, 2021; Pouliot et al., 2016; Shojaei et al., 2017).

In New South Wales, the strata plan is the legal reference that shows how ownership is set out. It includes floor plans for each level with lot numbers and areas marked as common property, a short stratum statement that records any vertical limits (for example, basement levels or the height/depth of outdoor areas), and notes that say whether boundaries follow the inside surfaces of walls, floors and ceilings or the building structure. Within this framework, a strata scheme is a building (or a group of buildings) divided into strata lots. A strata lot is the space of a unit as defined by those stated boundaries. Common property is everything not assigned to a lot and is owned by all owners (e.g., lifts, stairs, corridors) (NSW Land Registry Services, 2025). Because many schemes have several storeys and basements, clear level labels and a readable legend are essential for interpretation.

Prior work highlights the strengths and weaknesses of each option. Interactive systems are widely used in practice but vary in consistency and usability (Shojaei et al., 2017). 3D PDF is easy to distribute but can be heavy and limited for exploration (Pouliot et al., 2016). AR has been trialled with promising engagement but higher setup complexity (Dudhee and Vukovic, 2021). Visual variables also matter: position, size, colour, texture, and orientation influence how people read 3D

representations, and transparency or cross-sections must be used carefully to avoid confusion (Ying et al., 2020).

Cheng (2024) conducted a structured user survey to assess the usability and visual clarity of alternative 3D cadastre display models. It defined usability using ISO 9241-11 concepts of effectiveness, efficiency, and satisfaction, and used task-based Likert ratings of key visual elements. The survey reports several practical problems that block everyday use: participants noted setup and access friction with the interactive PDF (requiring specific software), slow or awkward navigation, large files that opened slowly, operation errors that blocked tasks, and visual issues such as colours that were hard to read and overlapping geometry effects that made boundaries unclear. Respondents recommended delivering the final product in a web browser to avoid installs and settings, using lighter colours, adding clear outlines to improve edge readability, and keeping a consistent visual language with simple controls and a readable legend. This study relied on commercial CAD/BIM tools and 3D PDF outputs, so access depended on having the right software or a paid licence (Cheng, 2024). In contrast, we built an open-source, browser-based workflow that avoids installs, keeps artefacts lightweight, and provides consistent interaction across levels, which also aligns with respondents' recommendations.

The 3D Cadastral Survey Data Model and Exchange (3D CSDM) is an Australia–New Zealand initiative sponsored by the Intergovernmental Committee on Surveying and Mapping (ICSM) with support from ANZLIC (the Spatial Information Council). Its 2022 Final Report sets out a common direction for representing lots and common property in three dimensions so jurisdictions can produce consistent, interoperable, and validation-ready data. The report recommends a web-first, multi-encoding strategy that includes JSON and JSON-LD, alongside GeoPackage, GML, and IFC, and it calls for practical reference implementations plus a clear validation pathway from schema

checks to geometry checks. This guidance provides the policy and technical context that our work aligns to (SURROUND New Zealand Pty Ltd et al., 2022).

Based on the survey findings and the 3D CSDM direction, the workflow was designed around several practical requirements. First, it should open in a standard web browser without plugins. Second, it should use a consistent colour and labelling scheme for levels and tenure, together with level and tenure filters, a readable legend, and basement handling. Third, it should support plan-based orientation and search for common tasks such as finding a lot, following it across levels, distinguishing lots from common property, and reading the 3D view against the 2D plan overlay. Finally, it should adopt a web-first fixed core JSON profile with selected official CSDM-aligned hooks while preserving all remaining source attributes as standard JSON fields, with validation aligned to the broader 3D CSDM pathway.

To meet these requirements, we built a static, client-side Reference Viewer (Design Baseline) with a fixed core JSON profile and locked its look and interaction using a pre-converted reference JSON dataset produced by the Python CLI converter. We then developed a Browser Converter so that users can prepare their own files without writing code, followed by an Upload-and-View component that renders those converted files in the same visual language. Figure 1 summarises this workflow, and full implementation details are provided in Section 3 (Data & Methods). Validation is planned as future work and will be aligned with recent model and exchange recommendations (SURROUND New Zealand Pty Ltd et al., 2022).

We implemented and tested the pipeline on a New South Wales strata scheme; results show fast load and smooth interaction on Windows in Chrome and Edge, with details in Section 4. Results & evaluation.

2. Related Work

2.1 Landscape and options

3D cadastral information spans several data standards and tool ecosystems. CityGML targets city-scale models; IFC is the BIM exchange format for building elements; LADM models parties, rights, and spatial units (Atazadeh et al., 2017; Stoter et al., 2019). Geometry formats such as OBJ, X3D, KMZ, and COLLADA appear in some workflows, and WebGL enables interactive graphics in the browser (Cheng, 2024). In practice, graphics/game tools (e.g., Blender), CAD/BIM tools (e.g., Revit, Civil 3D), and GIS tools (e.g., QGIS, ArcGIS) are commonly combined; 3D PDF and general 3D viewers are also used (Cheng, 2024). Prior reviews group display choices into three families: interactive viewers, 3D PDF, and augmented reality (AR) (Dudhee and Vukovic, 2021; Pouliot et al., 2016; Shojaei et al., 2017). Interactive viewers are flexible but vary in consistency and usability (Shojaei et al., 2017); 3D PDF is simple to distribute but can be heavy and limited for multi-level exploration (Pouliot et al., 2016); AR and VR can aid in-situ understanding but raise setup and environment constraints (Dudhee and Vukovic, 2021). These trade-offs motivated a lightweight, browser-based path with a fixed core JSON profile and consistent level/tenure cues.

2.2 Visual variables and perception

The way people read 3D cadastral scenes depends on visual variables such as position, size, shape, value, colour, texture, and orientation. Prior work reports that size and colour are effective

for distinguishing units, while transparency and cross-sections require careful use because they can help depth perception but also confuse boundaries if overused or oriented poorly (Ying et al., 2020). We apply these findings by using consistent colour by tenure, providing level filtering instead of heavy transparency, and keeping legends and labels readable.

2.3 Usability lenses and survey practice

Evaluation commonly follows ISO 9241-11 criteria: effectiveness, efficiency, and satisfaction. Task-based ratings of visual elements are used to judge how well viewers support common tasks, and short instruments such as Likert-style blocks are often sufficient for comparative assessments (Cheng, 2024). This lens informed our planned lightweight evaluation of interaction speed and clarity without running a large respondent study.

2.4 Examples of 3D visualisations

Case studies from New South Wales and Victoria in Australia, and international examples from China, the Netherlands, Poland, and Austria illustrate different balances of deployment simplicity, file size, interaction depth, and cost (Cheng, 2024). Government-hosted viewers demonstrate scalability, while research prototypes show specific features such as level filtering, cross-sections, plan overlays and web-based dissemination. Best-practice work in 3D cadastres has also emphasised the importance of clear visual communication of legal spaces and their spatial relationships for different user groups (Pouliot et al., 2018). More specific examples show different technical pathways: Alattas et al. (2021) demonstrate how private, common, and exclusive common spaces in multi-owner buildings can be mapped from BIM/IFC to LADM, while web-based prototypes have used browser-based environments such as Cesium to support interactive 3D cadastral dissemination (Kara et al., 2021; Višnjevac et al., 2019). A recent example is CadaSPACE, a cloud-based, low-cost platform for visualising 3D property rights in multi-storey buildings by combining cadastral/legal information, approximate BIMs, linked online tools, and semantic filtering (Andritsou and Potsiou, 2024). Together, these examples reinforce the value of browser-based access while also illustrating implementation paths that differ from the workflow proposed here. These exemplars set expectations for capability and helped scope our feature set.

2.5 Standards direction: 3D CSDM

The 3D CSDM is an ICSM and ANZLIC initiative in Australia and New Zealand that seeks consistent, interoperable, validation-ready 3D cadastre. The report recommends a web-first, multi-encoding approach that includes JSON and JSON-LD alongside GeoPackage, GML, and IFC, and calls for reference implementations and transformation or validation services bound to a canonical model (SURROUND New Zealand Pty Ltd et al., 2022).

Our pipeline maps to these directions as follows. We adopt a fixed core JSON profile for browser-based visualisation, with limited official CSDM-aligned JSON-LD hooks applied only to selected keys that have exact matches in the published vocabularies, namely object type, floor, zmin, zmax, and horizontalCRS, while all remaining source attributes are preserved as standard JSON fields. Browser Converter provides the transformation step that standardises inputs to this fixed core JSON profile. The Reference Viewer (Design Baseline) and

Upload-and-View function as reproducible, browser-based exemplars.

Current conversions already separate blocking errors from warnings, and planned work adds schema checks and basic 3D geometric checks consistent with the report’s validation pathway (SURROUND New Zealand Pty Ltd et al., 2022).

3. Data & Methods

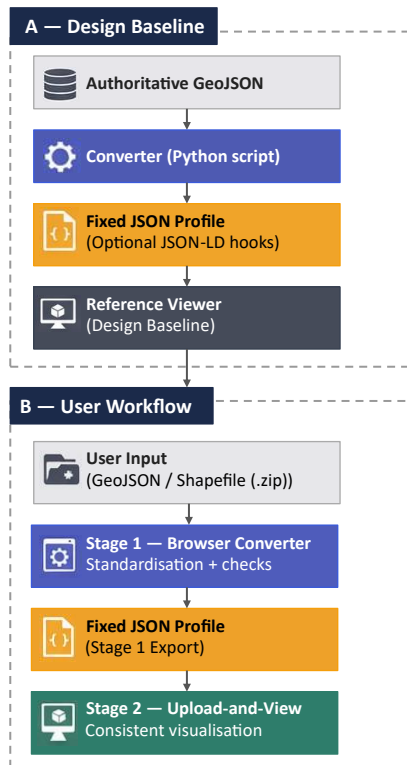


Figure 1: Pipeline overview: (A) design-baseline path and (B) user workflow; both produce the same fixed core JSON profile for web visualisation, with the same limited CSDM-aligned hook strategy and minor workflow-specific header and metadata differences.

3.1 Study area

The study area is a multi-storey strata scheme in New South Wales, Australia. Source material was supplied by state cadastral agencies through the project partners as plan-derived polygons representing lots and common property across levels, including cores such as lifts and stairs and basement areas where applicable. Data were provided in standard GIS formats, specifically GeoJSON and an ESRI Shapefile set; a GeoPackage version was also available upstream and, when used, was exported to GeoJSON or Shapefile before processing. Attribute fields include a lot or unit identifier, a tenure indicator distinguishing lots and common property, and a level reference; where available, base elevation and height are also present. Coordinate reference systems in the sources follow Australian map grid practice, with GDA94 MGA Zone 56 commonly encountered; the generated JSON outputs record CRS metadata in their headers for traceability. Figure 1 summarises the end-to-end flow from sources to the fixed core JSON profile and the two viewers.

3.2 Artifacts overview

The workflow is implemented through four main components: a fixed-design Reference Viewer, a Python CLI converter for preparing the reference dataset, a Browser Converter for user-side standardisation and export, and an Upload-and-View component for rendering converted files in the same visual language. The Reference Viewer provides the fixed baseline for look, controls, colours, and interactions, including level and tenure controls, plan overlay, search, legend, and counts.

In the current implementation, the web pages operate as static client-side applications served from a local or web static host. No server-side processing or API-based data retrieval is required. The reference dataset is prepared in advance by the Python CLI converter, while user datasets are manually supplied to the Browser Converter as GeoJSON or zipped Shapefile and exported as fixed core JSON profile files, so users do not need to write code. These files can then be opened in the Upload-and-View component, and the overall workflow remains simple and reproducible, with the main filenames and steps described in the Methods.

Artifact	Input → Output	Role in the workflow
reference-viewer.html	data-leylands-reference.json → browser rendering	Reference Viewer (Design Baseline): fixed UI baseline used to lock look, controls, colours, and interactions
convert_geojson_to_json.py	GeoJSON → fixed core JSON profile	Pre-Stage — CLI Converter: converts authoritative GeoJSON to the fixed core JSON profile used by the Reference Viewer
stage1-browser-converter.html	GeoJSON / zipped Shapefile → fixed core JSON profile	Browser Converter: client-side converter for standardising GeoJSON or zipped Shapefile to the fixed core JSON profile
stage2-upload-and-view.html	Browser Converter export → browser rendering	Upload-and-View: uploadable viewer that renders any Browser Converter JSON in the same visual language as the Reference Viewer
data-leylands-reference.json	—	Python-converted example displayed by the Reference Viewer
data-leylands-authoritative.geojson	—	Authoritative input for the Pre-Stage — CLI Converter
data-leylands-shapefile.zip	—	Alternative input bundle for Browser Converter

Table 1: Lists the software and datasets used in the workflow.

3.3 Fixed core JSON profile and attribute handling

To make the transformation step explicit, we use a fixed core JSON profile that keeps the source data in a browser-ready structure while applying only a small number of official CSDM-aligned JSON-LD hooks. The file remains standard JSON.



Figure 2: Reference Viewer (Design Baseline) - Web view of extruded lots and common property with level/tenure filters, core highlight, plan overlay, find tool, legend, and counts.

Official hooks are attached only where an exact match exists in the published vocabularies; in the current implementation, these are limited to object type, floor, zmin, zmax, and horizontalCRS, as shown in Table 2. All remaining source attributes are preserved as standard JSON fields so that the profile remains lightweight, readable, and directly usable by the browser workflow without introducing invented terms or unnecessary remodelling of the source data.

GeoJSON source field	JSON element in the profile	Official CSDM-aligned hook
each exported parcel object	@type	https://linked.data.gov.au/def/csdm/parcels/Parcel
FLOOR_NUMBER	floor	https://linked.data.gov.au/def/csdm/parcels/floor
BASE_Z	geometry.zmin	https://linked.data.gov.au/def/csdm/parcels/zmin
BASE_Z + HEIGHT	geometry.zmax	https://linked.data.gov.au/def/csdm/parcels/zmax
working file CRS / header metadata	horizontalCRS	https://linked.data.gov.au/def/csdm/container/horizontalCRS

Table 2: Shows the source-to-profile mapping used for the limited official hooks.

A simplified example is shown below.

Example 1. Simplified structure of the fixed core JSON profile

```
{
  "@context": {
    "horizontalCRS":
      "https://linked.data.gov.au/def/csdm/container/horizontalCRS",
    "floor":
      "https://linked.data.gov.au/def/csdm/parcels/floor",
```

```
    "zmin":
      "https://linked.data.gov.au/def/csdm/parcels/zmin",
    "zmax":
      "https://linked.data.gov.au/def/csdm/parcels/zmax"
  },
  "horizontalCRS": "EPSG:3857",
  "objects": [
    {
      "@type":
        "https://linked.data.gov.au/def/csdm/parcels/Parcel",
      "floor": 0,
      "geometry": {
        "zmin": 0.0,
        "zmax": 3.0
      }
    }
  ]
}
```

Additional and extended attributes are handled conservatively. Source fields are inherited directly and retained under properties.attrs. They are not automatically promoted to official hooks and are not forced into approximate mappings. Promotion occurs only where an exact published match exists in the official vocabularies. This keeps source-specific identifiers and attributes available to the viewer while avoiding unsupported claims of full semantic alignment.

Validation is treated as a staged process. In the current implementation, both the CLI converter and the Browser Converter already separate blocking errors from warnings during conversion and export preparation. The next stage is profile-level validation of required fields, header metadata, and hooked elements in the fixed core JSON profile, followed by basic geometric checks on polygon validity and extrusion parameters.

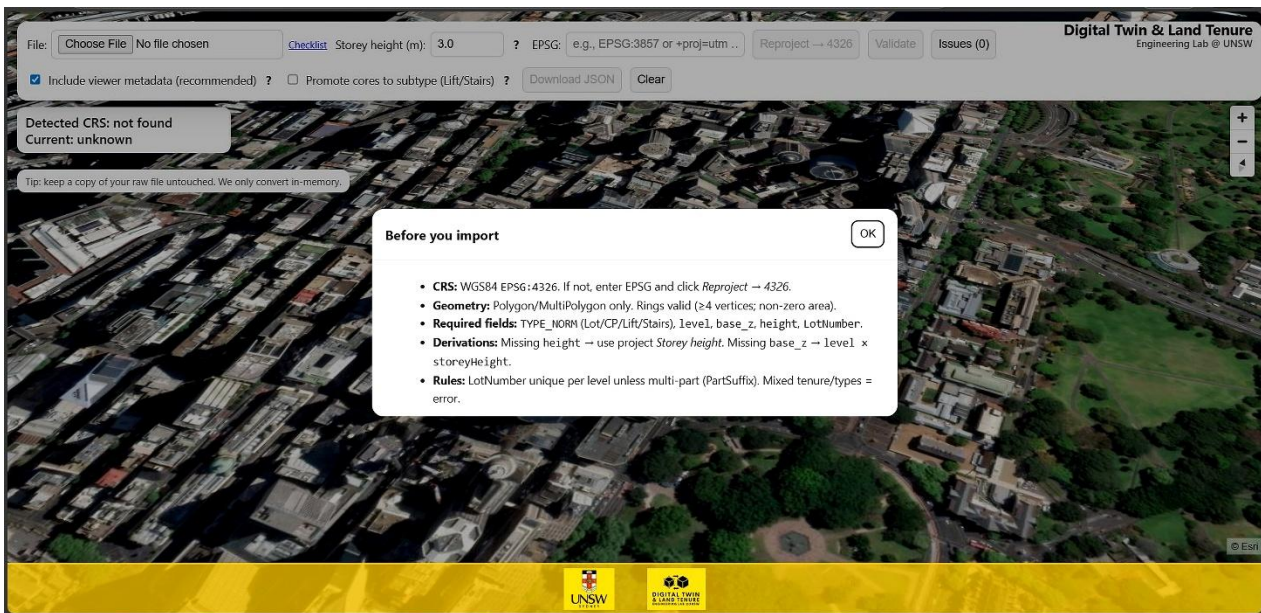


Figure 3: Stage 1 Browser Converter - Client-side page that standardises GeoJSON/Shapefile to the fixed core JSON profile; dialogue lists CRS, geometry, required fields, derivations, and blocking rules.

A later step will add an optional machine-readable rule file so that the same checks can be reproduced outside the browser environment. Section 3.6 provides a more detailed description of the current Browser Converter pre-export checks.

3.4 Reference Viewer (Design Baseline)

We first fixed the visual outcome. The Reference Viewer (Design Baseline) is a static, client-side page that runs in a standard web browser and loads a single known dataset (data-lylands-reference.json). It is termed “reference” because it serves as the fixed design baseline of the workflow, using one known dataset and a locked visual and interaction setup against which the later workflow components can be compared. It uses fixed colours, labels, legend, and camera, and allows no uploads or per-session overrides. The viewer renders unit volumes as extrusions and provides level and tenure filters, a basement handling note, a plan overlay, a search control, a legend, and an on-screen counter that reflects the current selection. Layout and styling are fixed so figures reproduced from the viewer are consistent across environments. This lightweight browser-based design was preferred over relying on external platforms or more general-purpose frameworks because it allowed the interface, styling, interaction logic, and figure output to be controlled consistently within a simple, reproducible static-file workflow. No server-side code is required. The page is served from a simple static host and opened in current versions of major browsers. An example scene is shown in Figure 2.

3.5 Pre-Stage: CLI Convert (Python) to fixed core JSON profile

To supply the Reference Viewer with a stable dataset, we prepared a small converter script, `convert_geojson_to_json.py`. The script reads a cadastral GeoJSON, maps the tenure and level fields to normalised names, assigns extrusion parameters (`base_z`, `height`) with sensible defaults where needed, and preserves all original fields under a dedicated `properties.attrs` block. It writes a compact header with units, CRS, generation time, and provenance, then outputs a fixed core JSON profile that the Reference Viewer loads consistently. A minimal run on the authoritative GeoJSON writes the file used in the figures.

The script is included with the submission and in the repository so that the same process can be reproduced on other data.

3.6 Stage 1: Browser Converter

Once the visual look and interactions were settled, we built a user pathway that reaches the same fixed core JSON profile without running code. The Browser Converter is a client-side converter page. It opens either a GeoJSON or JSON file, or a zipped ESRI Shapefile that contains the core parts (.shp, .shx, .dbf; .prj recommended). When inputs are not in EPSG:4326, the interface prompts for the source CRS and provides an in-browser reprojection step to 4326. Export is disabled until reprojection is complete and blocking errors are resolved. These pre-export checks form the first validation layer of the workflow and distinguish blocking errors from warnings before conversion to the fixed core JSON profile.

The converter standardises attributes into the fixed core JSON profile required by our viewers. It expects a tenure indicator, a level reference, and the two extrusion fields `base_z` and `height`. Field names are flexible, with alias matching for common variations in source data. Lot features must carry a lot identifier. When height is missing the converter uses a project “storey height” parameter with a default of 3.0 m. When `base_z` is missing it computes `level × storey height`. When level is missing but `base_z` is present it infers the level by dividing `base_z` by the storey height and rounding. Each derivation is recorded as a warning so users can review assumptions.

Geometry is restricted to polygons and multi-polygons. In the current workflow, point and line geometries are outside scope because the Browser Converter and fixed core JSON profile are designed around parcel extrusion from area-based footprints. The converter checks that the outer ring is valid with non-zero area and sufficient vertices. It flags self-intersections and duplicate consecutive vertices as warnings. Numeric sanity is also reviewed; heights less than or equal to zero and unusually large magnitudes are marked for attention. Rules that undermine correctness are treated as errors and prevent export. Examples include a Shapefile zip that is missing core parts, a lot feature with no lot identifier, or mixing lot and common-property records

within the same lot at the same level. On download, the converter writes a compact header with schema version, units in metres, EPSG:4326, a generation timestamp, and a global bounding box. Each object carries normalised properties for tenure, level, and identifiers; the original attributes under properties.attrs; and an extrusion geometry with base_z, height, and a closed 2D ring profile in longitude and latitude. An optional switch promotes attributes that indicate lift or stair cores into a subtype while normalising tenure to common property. If a multi-part lot lacks explicit part suffixes the exporter assigns A, B, and C labels in a consistent order and records that they were auto generated. The interface and pre-export checks are illustrated in Figure 3.

3.7 Stage 2: Upload-and-View

To support user-supplied datasets directly, we developed Stage 2: Upload-and-View. Unlike the Reference Viewer, which is tied to one fixed reference dataset, Upload-and-View accepts a Browser Converter export at runtime as a user-selected file and presents it in the same visual language. This separation is needed because the Browser Converter prepares user data, while Upload-and-View provides the corresponding viewer for those converted files without requiring users to replace the reference dataset or edit file paths manually. The current version is functional end-to-end and will receive small interface and behaviour refinements so that it fully matches the Reference Viewer. Once these refinements are complete, both viewers will expose identical controls and responses.

3.8 Deployment and reproducibility

The workflow is delivered as static HTML and JavaScript pages that run in a standard web browser. Here, "static" means that the pages are served as files without server-side application logic or database processing. A simple static file server is sufficient for both local use and web hosting. In practice, we served the directory from the QGIS OSGeo4W Shell by invoking Python's built-in HTTP server and opened the viewer in the default browser. The same command can be used from any terminal on Windows, macOS, or Linux. Equivalent options include Node's http-server, the VS Code Live Server extension, GitHub Pages, and other static hosts. Once opened, the Browser Converter and Upload-and-View run entirely in the browser.

For local reproduction, the Python CLI converter is first run once to create the fixed core JSON profile used by the Reference Viewer. A static server is then started, the viewer page is opened in the browser, and Upload-and-View can be used to open any JSON file exported by the Browser Converter. Exact filenames are listed in Table 1, and the repository includes a short README with the required commands.

4. Results & evaluation

Figure 4 illustrates representative viewer interactions corresponding to the main tasks evaluated in the pilot, including feature selection, level filtering, search, basement handling, and floor-plan overlay.

Evaluation setup, we used the ISO 9241-11 usability lens (effectiveness, efficiency, satisfaction) and, for this pilot, explicitly evaluated the first two pillars (effectiveness and efficiency) in a stated context of use. The purpose of this pilot was to assess how well the browser workflow supported representative interaction tasks on the New South Wales strata case study shown in Figure 2, rather than to make broader claims about performance across different building types or datasets.

Context: Windows 11, two desktop browsers (Chrome, Edge), ASUS FX570UD (Intel Core i5-8250U, 16 GB RAM, 4 GB dGPU), single evaluator. One evaluator performed five representative tasks: filter to a specific level, identify tenure, find a lot by ID, toggle the plan overlay, and reset the view. Measures and instrumentation: we recorded (a) page open/load time (seconds, end-to-end as perceived by the user), (b) per-task completion time (seconds), (c) task success (Y/N), and (d) interaction smoothness via average FPS captured with DevTools → Performance during a short rotate/pan interaction window. We also noted network transfer size (Network panel) and working-set memory (steady/peak) as supportive performance indicators consistent with ISO/IEC 25010's "performance efficiency" characteristic (International Organization for Standardization, 2018). No satisfaction ratings were collected in this pilot. Reported timing and FPS figures are means over *N* repeats; cache state = [disabled/enabled].

Results are summarised in Table 3. Initial load completed in ~0.6 s in Chrome and ~1.48 s in Edge. During rotation and typical interaction, average frame rate was ~50 FPS in Chrome and ~60 FPS in Edge. Level and tenure filtering applied within ~40--55 ms and was perceived as instantaneous. All five tasks were completed successfully in both browsers. Network transfer was ~2.4--2.6 MB in total, with the dataset at ~0.214 MB. Observed memory refers to renderer working set: ~398/407 MB (steady/peak) in Chrome and ~93/100 MB in Edge.

Measure	Chrome	Edge
Page load time	~0.60 s	~1.48 s
Avg. interaction frame rate	~50 FPS	~60 FPS
Level / tenure filter latency	~40–55 ms	~40–55 ms
Task success (5 tasks)	5/5 ✓	5/5 ✓
Network transfer (total)	~2.4–2.6 MB	~2.4–2.6 MB
Dataset size	~0.214 MB	~0.214 MB
Memory — steady / peak	~398 / 407 MB	~93/100 MB

Table 3: Browser trial results summary (Windows 11, ASUS FX570UD, single evaluator, NSW strata dataset, cache disabled/enabled).

Results were broadly consistent across both browsers for task success, filter latency, and transfer size. In this small pilot, Chrome showed higher memory use, while Edge showed slightly higher frame rates.

Interpretation and scope, these pilot results are consistent with the ISO 9241-11 focus on effectiveness and efficiency because they report task success together with measures of interaction speed and responsiveness in a defined context of use. The reported measures are also relevant to ISO/IEC 25010 performance efficiency because they describe time behaviour and resource use through page-load time, filter latency, frame rate, network transfer, and working-set memory. For the operations tested, the initial results show promising performance, including ~0.6–1.48 s page opening, ~40–55 ms filter response, and ~50–60 FPS interaction (International Organization for Standardization and International Electrotechnical Commission, 2024; International Organization for Standardization, 2018).

This pilot evaluation was intentionally designed as a task-based user-experience baseline in a defined context of use, focusing on representative browser-based operations rather than claiming

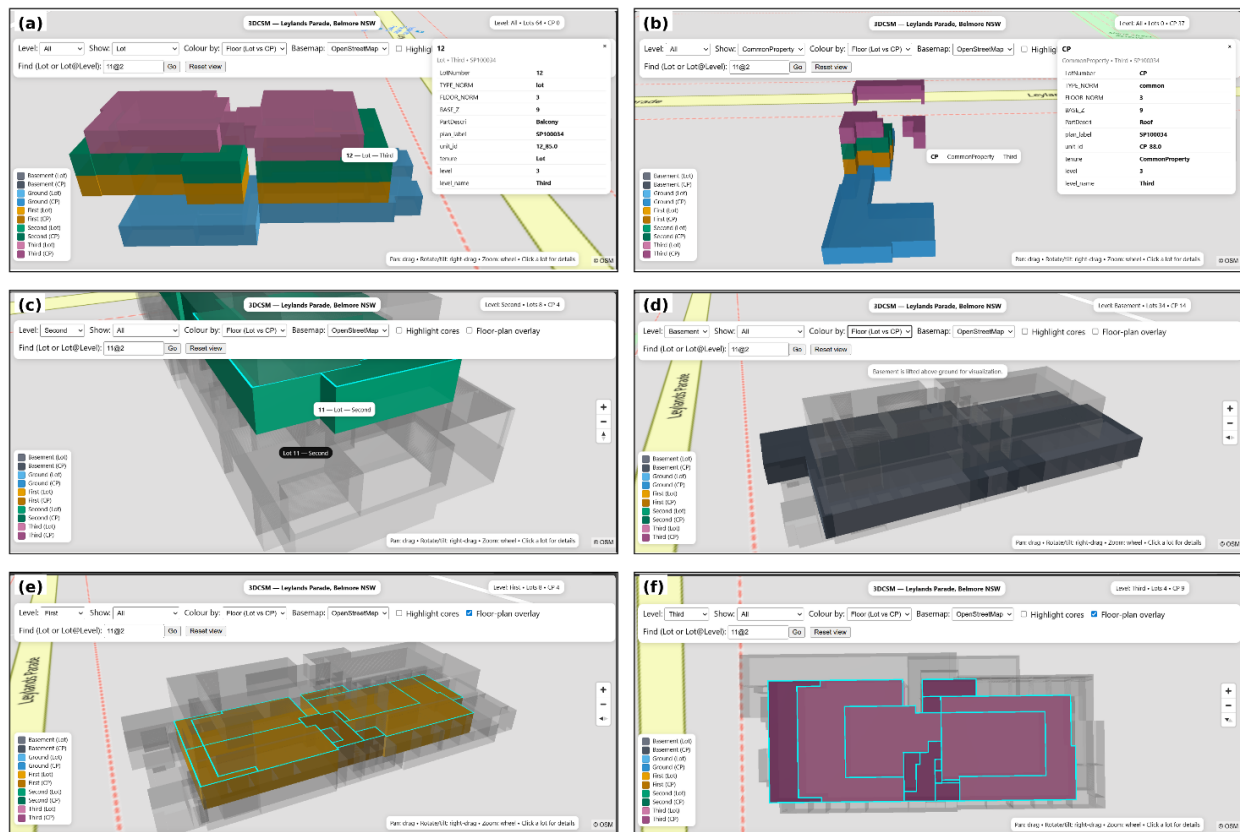


Figure 4. Examples of key viewer interactions in the Reference Viewer: (a) lot selection with attribute/details panel, (b) common-property selection with attribute/details panel, (c) find/search result, (d) basement filtering with lifted-basement note, (e) level filtering to a selected floor, and (f) floor-plan overlay for plan-to-volume comparison.

general usability across building types, datasets, or deployment settings. The context of use is explicitly scoped: one evaluator, one Windows machine, two modern browsers, a NSW strata dataset, and five fixed tasks. Broader participant and cross-platform testing on larger datasets are reserved for future work.

As this pilot runs locally, network timing components were out of scope; for future technical depth, once the viewer is deployed on the public web, we can complement end-to-end open time by reporting response-time components, DNS lookup, TCP connect, time-to-first-byte, and content download, following Kalantari et al. (2005).

5. Alignment and future work

Our pipeline follows the 3D Cadastral Survey Data Model and Exchange direction toward web-first, multi-encoding delivery. In the present workflow, GeoJSON and zipped Shapefile are accepted as source inputs and standardised into a fixed core JSON profile for browser-based visualisation. We use a fixed core JSON profile with the same limited official CSDM-aligned hook strategy described in Section 3.3, while preserving all remaining source attributes as standard JSON fields. Browser Converter standardises heterogeneous inputs to this profile, and both the Reference Viewer and the Upload-and-View act as browser-based exemplars for rendering and communication.

Next, we will finish making the Upload-and-View behave exactly like the Reference Viewer. The current Upload-and-View build renders Browser Converter exports correctly, but a few details still need parity, including legend spacing, search hinting,

and minor interface interactions. After parity, we will integrate conversion and viewing into a single-entry point so a user can upload a zipped Shapefile or GeoJSON, convert to the fixed core JSON profile, visualise in the same style, and export figures without switching tools.

We will also add a light validation panel inside the viewer. The initial scope will cover geometry checks, semantic checks, and an optional machine-readable rule file so the same checks can be reproduced outside the viewer. At this stage, we plan to focus on practical issues commonly reported in the literature and standards guidance, and we will refine the exact ruleset after a short review to ensure it is well-grounded. Issues will appear beside the scene with short messages and clickable feature IDs.

For deployment, we will move from a local static host to a public site so users can run conversion, visualisation, and validation directly in the browser. We will publish the viewer pages, the Browser Converter, the example dataset, and a short README with the few commands needed to serve the pages on any static host.

We will also broaden evaluation by repeating the same ISO 9241-11 based protocol on additional machines and browsers and with multiple participants while keeping the procedure brief and repeatable. This broader evaluation will extend the current pilot baseline to additional datasets and building types, so that the experience-focused findings reported here can be tested under a wider range of practical conditions. The public repository link will be provided in the paper's data and code availability note and on the project website.

6. Conclusion

This paper presented a lightweight, browser-based pathway for 3D cadastral communication built around a fixed core JSON profile, a Reference Viewer (Design Baseline), a Browser Converter, and an Upload-and-View component that renders converted files in the same visual language. The approach follows current 3D CSDM model and exchange guidance by making the transformation step explicit and supporting browser-based visualisation through a consistent fixed-core profile. Results reflect a single evaluator on one platform and two browsers, and broader testing and cross-platform runs remain future work. Next steps are to finalise parity between Upload-and-View and the Reference Viewer, add a light in-viewer validation panel, and integrate conversion, visualisation, and validation into one public website with open code and an example dataset.

Data and code availability

The Reference Viewer, Browser Converter, Upload-and-View, the example dataset, and a brief README are available at Figshare: <https://doi.org/10.6084/m9.figshare.30632786>. The package includes the fixed core JSON profile and the minimal steps required to serve the pages from a static host.

Conflicts of interest

The authors declare no conflicts of interest.

Acknowledgements

This work was supported by the Australian Research Council under Grant LP22020090.

References

- Alattas, A., Kalogianni, E., Alzahrani, T., Zlatanova, S., van Oosterom, P., 2021. Mapping private, common, and exclusive common spaces in buildings from BIM/IFC to LADM. A case study from Saudi Arabia. *Land use policy* 104, 105355. <https://doi.org/10.1016/j.landusepol.2021.105355>
- Alattas, A., Zlatanova, S., Oosterom, P. Van, Chatzinikolaou, E., Lemmen, C., Li, K.J., 2017. Supporting indoor navigation using access rights to spaces based on combined use of IndoorGML and LADM models. *ISPRS Int. J. Geo-Information* 6. <https://doi.org/10.3390/ijgi6120384>
- Andritsou, D., Potsiou, C., 2024. CadaSPACE: A Cloud Based Platform for a low - cost 3D visualization of property rights available in a 2D cadastral registry. An example for urban multi - storey buildings. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* 10, 25–31. <https://doi.org/10.5194/isprs-annals-X-4-W5-2024-25-2024>
- Atazadeh, B., Kalantari, M., Rajabifard, A., Ho, S., Champion, T., 2017. Extending a BIM-based data model to support 3D digital management of complex ownership spaces. *Int. J. Geogr. Inf. Sci.* 31, 499–522. <https://doi.org/10.1080/13658816.2016.1207775>
- Cheng, J., 2024. *Developing Visualisation Options of Common Property in 3D*. University of New South Wales.
- Dudhee, V., Vukovic, V., 2021. Building information model visualisation in augmented reality. *Smart Sustain. Built Environ.* 12, 919–934. <https://doi.org/10.1108/SASBE-02-2021-0021>

International Organization for Standardization; International Electrotechnical Commission, 2024. *ISO/IEC 25010:2024 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuARE) — System and software quality models*. Geneva, Switzerland.

International Organization for Standardization, 2018. *ISO 9241-11:2018 Ergonomics of human–system interaction — Part 11: Usability: Definitions and concepts*. Geneva, Switzerland.

Kalantari, M., Rajabifard, A., Wallace, J., Williamson, I.P., 2005. Toward e-Land Administration: Australian Online Land Information Services, in: *Proceedings of SSC 2005 Spatial Intelligence, Innovation and Praxis: The National Biennial Conference of the Spatial Sciences Institute*. Spatial Sciences Institute, Melbourne, Australia.

Kara, A., van Oosterom, P., Kathmann, R., Lemmen, C., 2021. Requirements and Opportunities for Web-Based 3D Visualization and Dissemination of Property Valuation Information. *Proceedings of the 7th International FIG 3D Cadastre Workshop*. 113–128.

NSW Land Registry Services, 2025. Strata Schemes (Registrar General's Guidelines). URL https://rg-guidelines.nswlrs.com.au/strata_schemes (accessed 3.10.26).

Pouliot, J., Ellul, C., Hubert, F., Wang, C., Rajabifard, A., Kalantari, M., Shojaei, D., 2016. 3D Cadastre Visualization : Recent Progress and Future Directions, in: *Proceedings of the 5th International FIG 3D Cadastre Workshop*. Athens, pp. 28–37.

Pouliot, J., Ellul, C., Hubert, F., Wang, C., Rajabifard, A., Kalantari, M., Shojaei, D., Atazadeh, B., van Oosterom, P., de Vries, M., Ying, S., 2018. Chapter 5: Visualization and New Opportunities, in: van Oosterom, P. (Ed.), *3D Cadastres Best Practices*. International Federation of Surveyors (FIG), Istanbul, pp. 183–230.

Shojaei, D., Olfat, H., Briffa, M., Rajabifard, A., 2017. 3D digital cadastre journey in Victoria, Australia. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. IV-4/W5*, 26–27.

Stoter, J., Ho, S., Biljecki, F., 2019. Considerations for a contemporary 3D cadastre for our times. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.* 42, 81–88. <https://doi.org/10.5194/isprs-archives-XLII-4-W15-81-2019>

SURROUND New Zealand Pty Ltd; OpenWork Ltd; Pangaea Innovations Pty Ltd; Evolve & Amplify; McKenzie & Co, 2022. 3D Cadastral Survey Data Model and Exchange – Final Report. Auckland, New Zealand.

van Oosterom, P., Lemmen, C., Thompson, R., Janečka, K., Zlatanova, S., Kalantari, M., 2018. Chapter 3: 3D Cadastral Information Modelling, in: van Oosterom, P. (Ed.), *FIG Publication*. International Federation of Surveyors (FIG), Istanbul, pp. 95–132.

Višnjevac, N., Mihajlović, R., Šoškić, M., Cvijetinović, Ž., Bajat, B., 2019. Prototype of the 3D cadastral system based on a NoSQL database and a Javascript visualization application. *ISPRS Int. J. Geo-Information* 8. <https://doi.org/10.3390/ijgi8050227>

Ying, S., Chen, N., Li, W., Guo, R., He, B., Zhao, Z., Wang, S., 2020. Visualization Methods for the Coherent Set of 3D Building Property Units. *Geomatics Inf. Sci. Wuhan Univ.* 45, 81–88. <https://doi.org/10.13203/j.whugis20190242>