

OGC® DOCUMENT: 22-041

External identifier of this OGC® document: <http://www.opengis.net/doc/PER/T18-D012>



Open
Geospatial
Consortium

TESTBED-18: BUILDING ENERGY DATA INTEROPERABILITY ENGINEERING REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2023-02-09

Approval Date: 2023-06-08

Publication Date: 2023-08-16

Editor: Leigh St. Hilaire, Aidan Brookson

Notice: This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is *not* an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

License Agreement

>Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Copyright notice

Copyright © 2023 Open Geospatial Consortium
To obtain additional rights of use, visit <https://www.ogc.org/legal>

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I.	EXECUTIVE SUMMARY	ix
I.A.	Testbed background	ix
I.B.	Testbed goals	ix
I.C.	Testbed tasks description	x
I.D.	Testbed locations	x
I.E.	Testbed data sources	x
I.F.	Testbed activities	xi
I.G.	Initial Processing Track Summary	xiii
I.H.	Extended Processing Track Summary	xv
I.I.	Data Visualization Track Summary	xvi
I.J.	Testbed challenges, lessons learned, and recommendations.	xxi
I.K.	Conclusions	xxiii
II.	KEYWORDS	xxiv
III.	PREFACE	xxv
IV.	SECURITY CONSIDERATIONS	xxvii
V.	SUBMITTING ORGANIZATIONS	xxviii
VI.	SUBMITTERS	xxviii
VII.	ACKNOWLEDGEMENTS	xxix
1.	SCOPE	2
2.	NORMATIVE REFERENCES	4
3.	TERMS, DEFINITIONS AND ABBREVIATED TERMS	8
3.23.	Abbreviated terms	13
4.	INTRODUCTION	16
4.1.	Report overview	16
4.2.	Testbed background and use cases	16
4.3.	Testbed goals	17
4.4.	Testbed and task description	20
5.	TESTBED-18 EXPERIMENTAL ARCHITECTURE	22
5.1.	Architecture background	22

5.2. Participant roles	22
5.3. Testbed-18 investigative tracks	24
5.4. Testbed data inputs and parameters	25
6. TESTBED EXPERIMENTS – INITIAL PROCESSING TRACK	30
6.1. Description	30
6.2. Test Locations	30
6.3. Test Data	30
6.4. Initial processing track summary	30
6.5. Building Energy Processing Service (Steinbeis Consortium)	31
6.6. Building Energy Data Service (interactive instruments)	38
6.7. Geospatial and Building Energy Data Service (Ecere)	40
7. TESTBED EXPERIMENTS – EXTENDED PROCESSING TRACK	55
7.1. Description	55
7.2. Test Locations	55
7.3. Test Data	55
7.4. Extended processing track summary	56
7.5. Building Energy Processing Service (Steinbeis Consortium)	57
7.6. Building Energy Data Service (interactive instruments)	61
8. TESTBED EXPERIMENTS – DATA VISUALIZATION TRACK	79
8.1. Description	79
8.2. Relevance to test bed goals	79
8.3. Data Visualization Track Summary	79
8.4. Visualization in the Browser as part of OGC Web APIs (interactive instruments)	84
8.5. Building Energy Client as 3D Web Application (Steinbeis Consortium)	88
8.6. Building energy client (Climative)	91
8.7. Building energy client (Properate)	93
8.8. Building energy client as 3D web app map viewer D128 (GeoSolutions)	95
8.9. Unity-Based Augmented Reality Visualization (Ethar Inc.)	100
9. TESTBED CHALLENGES, LESSONS LEARNED, AND RECOMMENDATIONS	135
9.1. Data, standards, and interoperability	135
9.2. Practices, collaboration, and communication	141
9.3. Examples of integration potential and selected data models/standards	144
10. FUTURE WORK	148
10.1. Automated workflow	148
10.2. Commercial building datasets and archetypes	149
10.3. Acquiring new data sources	149
10.4. Processing of datasets	150
10.5. Connection to other existing tools	150
11. CONCLUSIONS	152
11.1. General goals	152
11.2. Data model goals	152

11.3. Detailed use case goals	153
11.4. Final Notes	153
ANNEX A (INFORMATIVE) REVISION HISTORY	156
BIBLIOGRAPHY	158

LIST OF TABLES

Table 1	xi
Table 2	xii
Table 3	xiii
Table 4	xv
Table 5	xvi
Table 6	23
Table 7	24
Table 8	26
Table 9	31
Table 10	34
Table 11	35
Table 12 – Extended processing track summary	56
Table 13 – Comparison of data volumes in MB by format	66
Table 14	80

LIST OF FIGURES

Figure 1 – The data conflation and example processing workflows for the Initial Processing Track.	xiii
Figure 2 – Data conflation workflow for Extended Processing Track.	xv
Figure 3 – Map with the buildings color-coded based on the specific space heating demand	xvii
Figure 4 – Result visualization in Building Energy Client D125 Steinbeis Consortium	xviii
Figure 5 – Style classification based on monthly energy demand property applied to the interactive instruments’s Nun’s Island collection visualized in 3D inside a MapStore custom application	xix
Figure 6 – Ethar WebGL Client Desktop Visualization	xx
Figure 7 – Ethar WebGL Client Mobile Android Visualization	xxi
Figure 8 – Testbed-18 participant roles	23
Figure 9 – Overall Workflow Diagram of Steinbeis Consortium Contribution.	32

Figure 10 – Web 3D Visualization of Enriched Nuns’ Island CityGML Dataset.	33
Figure 11 – A high-level workflow of building energy processing service developed by the Steinbeis Consortium.	37
Figure 12 – The initial processing track workflow relevant to interactive instruments.	38
Figure 13 – The same building in Cesium JS using 3D Tiles	40
Figure 14 – Montreal 3D Buildings dataset imported from CityGML visualized in Ecere’s GNOSIS Cartographer	41
Figure 15 – Nuns’ Island subset Montreal 3D Buildings dataset imported from CityGML visualized in Ecere’s GNOSIS Cartographer	42
Figure 16 – Cesium JS client visualizing Montreal 3D Tiles generated from Ecere’s GNOSIS Map Server (downtown)	43
Figure 17 – Cesium JS client visualizing Montreal 3D Tiles generated from Ecere’s GNOSIS Map Server (Habitat 67)	44
Figure 18 – Cesium JS client visualizing Montreal 3D Tiles generated from Ecere’s GNOSIS Map Server (close up view)	44
Figure 19 – An example tile batching multiple buildings in a single 3D model tile	45
Figure 20 – Vector tile containing reference points, each referencing an individual model	46
Figure 21 – Single building accessed from Montreal 3D Buildings collection	46
Figure 22 – Node IDs within glTF model allowing to link to OGC API - Features item to retrieve energy attributes (shown in Blender)	47
Figure 23 – Single feature accessed from Montreal 3D Buildings collection (OGC API - Features)	48
Figure 24 – Resulting feature collection augmented with energy attributes calculated on-demand from SimStadt API	49
Figure 25 – Flow diagram illustrating use of Proceses - Part 3 to combine multiple nested processes in workflow to create feature collection augmented with energy attributes	49
Figure 26 – GNOSIS Cartographer’s Workflow Editor configuring AddAttributes process to take as input output from SimStadt API (through MOAW Adapter helper process), joining attributes based on UUID	51
Figure 27 – GNOSIS Cartographer’s Workflow Editor configuring SimStadt API process, selecting overall region of interest and energy calculation attributes of interest	52
Figure 28 – Data conflation process behind production of Montreal wide CityGML building stock dataset.	58
Figure 29 – Montreal web 3D viewer.	60
Figure 30 – The extended processing track workflow for accessing the City of Montreal data.	62
Figure 31 – The tileset visualized in Cesium JS	64
Figure 32 – Sample map with the Mapbox Vector Tiles in MapLibre JS	65
Figure 33 – Buildings in Ville-Marie build before 1870 (HTML, glTF-Binary, CityJSON)	67
Figure 34 – 500 buildings in Hampstead (HTML, glTF-Binary, CityJSON)	68
Figure 35 – Buildings in a small bounding box around Percival Molson Stadium (HTML, glTF-Binary, CityJSON)	69

Figure 36 – Buildings taller than 150 meters in Ville-Marie (HTML, gITF-Binary, CityJSON)	70
Figure 37 – Notre-Dame (HTML, gITF-Binary, CityJSON)	71
Figure 38 – A single building feature in the dataset, that is actually a collection of buildings including a church and an office building (HTML, gITF-Binary, CityJSON)	72
Figure 39 – Example where many buildings are bundled as a single building	74
Figure 40 – The extended processing track workflow for accessing the Quebec Housing Archetypes.	76
Figure 41 – Map with the buildings color-coded based on the specific space heating demand	80
Figure 42 – Result visualization in Building Energy Client D125 Steinbeis Consortium	81
Figure 43 – Style classification based on monthly energy demand property applied to the interactive instruments’s Nun’s Island collection visualized in 3D inside a MapStore custom application	82
Figure 44 – Ethar WebGL Client Desktop Visualization	83
Figure 45 – Ethar WebGL Client Mobile Android Visualization	84
Figure 46 – Map with the buildings color-coded based on space heating demand	86
Figure 47 – Overview of Building Energy Client D125 Steinbeis Consortium	88
Figure 48 – Simulation configuration in Building Energy Client D125 Steinbeis Consortium.	89
Figure 49 – Result visualization in Building Energy Client D125 Steinbeis Consortium	90
Figure 50 – Visualizing building energy data service D123 on the Steinbeis Consortium client.	91
Figure 51 – Climative data processing workflow	93
Figure 52 – Montreal 3D Tiles layer retrieved from an OGC API features endpoint rendered inside a MapStore WebGIS client	97
Figure 53 – The property structure of energyDemands	98
Figure 54 – Style classification based on monthly energy demand property applied to the interactive instruments’s Nun’s Island collection visualized in 3D inside a MapStore custom application	99
Figure 55 – Style classification based on monthly energy demand property applied to the interactive instruments’ Nuns’ Island collection visualized in 3D inside a MapStore custom application	99
Figure 56 – Testbed 18 Building Energy Deliverable Relationships (Credit: OGC)	101
Figure 57 – AR Viewing Mode Process Options (Credit: Ethar)	102
Figure 58 – GeoAR Conceptual Rendering Process (Credit: Ethar)	103
Figure 59 – ETSI AR Framework incorporating OGC Standards (Credit: ETSI)	104
Figure 60 – Ethar WebGL Client Desktop Visualization (Credit: Ethar)	112
Figure 61 – Ethar WebGL Client Mobile Android Visualization (Credit: Ethar)	113
Figure 62 – Ethar AR Fiducial Marker (Credit: Ethar)	114
Figure 63 – Ethar Mobile Application Screenshots (Credit: Ethar)	115
Figure 64 – Ethar Mobile Application Mapped Data (Credit: Ethar)	115
Figure 65 – Bird’s Eye rendering of AR over a real-world scene (Credit: Ethar)	116

Figure 66 – Bird’s Eye rendering of AR over a real-world scene (Credit: Ethar)	117
Figure 67 – Ethar AR Application Architecture Fundamentals (Credit: Ethar)	122
Figure 68 – NASA Unity plugin rendered through updated CESIUM_RTC code (Credit: Ethar)	124
Figure 69 – NASA Unity Plugin rendering with building vertical offset anomalies (Credit: Ethar)	125
Figure 70	145
Figure 71	146



EXECUTIVE SUMMARY

I.A. Testbed background

Within the context of annual global energy consumption, buildings are one of the most significant consumers. Regionally focused, building related analyses are important in arriving at data driven climate change policy that reduces the overall energy consumption of buildings and provides other benefits to society. Policymakers need simple tools that enable them to conduct rapid visualization-driven assessments of how new or existing building related climate change policies will impact the energy and emissions profiles of the building stock within specific geographic regions (i.e. a city, or a province). The component geospatial and building energy datasets needed to build these tools are available globally from many sources, but the datasets exist in different political, data, and expertise domains. The lack of dataset interoperability and lack of communication between each dataset's stakeholders results in duplication of effort, lost potential for energy savings, and lost opportunities for effective policy tools. Standards that enable interoperability of building energy and geospatial datasets would therefore address key barriers to integrated planning activities and tools for policymakers.

As the transformation to spatial computing continues, a key potential use case for data interoperability is the Canadian Geospatial Data Infrastructure (CGDI). The CGDI is an infrastructure for geospatial information. The CGDI concept is similar to the traditional physical infrastructure that helps Canadians with their everyday lives (e.g., roads, utilities, and telecommunications). The development of RESTful Web APIs, enhancements in the representation of 3D building data, and the growing availability of modeled energy data have allowed the rudimentary extension of some CGDI capabilities to be used within the domain of buildings and energy. Providing a fully functioning standards based energy and utilities component of the CGDI, enabled by interoperability and organizational communication, would go a long way to enabling municipal and utility entities to implement better energy and emissions policy and programs and work towards implementing Urban Digital Twins.

I.B. Testbed goals

The goal of the OGC Testbed-18: Building Energy Data Interoperability task was to address interoperability between geospatial and building energy datasets using existing OGC standards and where appropriate, fostering the development of new OGC standards. The task participants showed examples of the types of visualization tools that can benefit stakeholders and are enabled by Web API services that conform to OGC standards.

To accomplish those aims, the participants prototyped the components of an Energy Spatial Data Infrastructure (ESDI) that could become part of the CGDI and support the execution of building energy experiments and analysis – ideally using existing OGC Web API standards. Testbed-18

workflows attempted to use those components that implemented OGC Web APIs to build and access interoperable generalized data models containing geospatial and building energy data. The Testbed-18 Building Energy workflows focused on enabling the following stakeholder use cases.

- Provide the geospatial tools governments and other entities need to be able to quickly look at the existing building stock within their jurisdictions and assess a variety of different building retrofit scenarios that support climate change related retrofit policy and program development.
- Enable utilities to perform hosting capacity analysis, which is an analytical approach that can help utilities and the jurisdictions they serve understand how the electric grid is geospatially-impacted by the connection of new or modification of existing energy sources and sinks.

I.C. Testbed tasks description

Testbed-18 tasked participants with exploring existing datasets available from the building and energy domains in Canada. The participants developed mapping and integration approaches that aimed to combine the provided and available data sets into a single data model. The intent of the single model was to enable the use of services and/or interfaces (components) compliant with the model data for the direct and simple analysis, simulation, and visualization of building energy and greenhouse gas emissions metrics without further integration and mapping efforts. The data were served and accessed via standard-based Web services that aimed to be compliant with the latest OGC Web API standards.

I.D. Testbed locations

The locations chosen by the Testbed-18 participants and Natural Resources Canada (NRCan), the Testbed-18 sponsor, were the City of Montreal, Quebec, Canada, and the Nuns' Island neighborhood within it. These locations were chosen because of their publicly available geospatial buildings datasets.

I.E. Testbed data sources

The Testbed-18 participants had access to the following geospatial and building (energy) datasets relevant to the chosen locations:

- Montreal agglomeration property assessment roll data (City of Montreal);

- City of Montreal 3D buildings 2016 LoD 2 model with textures (City of Montreal);
- geoindex shared platform (Province of Quebec);
- City of Montreal digital terrain model (City of Montreal);
- automatically Extracted Buildings (NRCan);
- building, alteration and demolition permits (City of Montreal);
- EnerGuide Rating System (ERS) database entries (NRCan);
- Housing Technology Assessment Platform (HTAP) archetype files for HOT2000 software (NRCan CanmetENERGY); and
- reduced HOT2000 File Representation (NRCan CanmetENERGY).

It is important to note here that although these datasets were provided, they were not all used within the Testbed activities.

I.F. Testbed activities

During Testbed-18 all data models and OGC standards compliant API implementations were demonstrated as prototypes that included several services and clients – the participant roles. Short descriptions of these participant roles and the Testbed-18 participants assigned to them follow in the table below.

Table 1

PARTICIPANT ROLE	PARTICIPANTS	DESCRIPTION
Building Energy Data Service (D122 and D123)	<ul style="list-style-type: none"> • Ecere • interactive instruments 	The Building Energy Data Service was a Web API instance that served building energy data according to the data models investigated in this ER.
External Geospatial Data Service (D127 and D128)	<ul style="list-style-type: none"> • Ecere • GeoSolutions 	The External Geospatial Data Service was a Web API instance that provided access to geospatial data that was required for the applications offered by the Building Energy Processing Service (D126).
Building Energy Processing Service (D126)	<ul style="list-style-type: none"> • Steinbeis Consortium 	The Building Energy Processing Service was a Web API instance providing access to applications that make use of the components D122, D123, D127, and D128.
Building Energy Client (D124 and D125)	<ul style="list-style-type: none"> • Ethar • Properate • Steinbeis Consortium • Climative • interactive instruments 	The Building Energy Client applications accessed the other Testbed-18 web services and used them to provide stakeholder access to building characteristics and energy data through web and augmented reality interfaces.

PARTICIPANT ROLE	PARTICIPANTS	DESCRIPTION
	<ul style="list-style-type: none"> GeoSolutions 	

Different cases involving the prototype services and clients were explored in Testbed-18 through tracks of experimentation. These tracks both allowed parallelization of the tasks within Testbed-18 and the case-by-case analysis of different interactions between the services and clients. A summary of the participants, participant roles, and activities performed within each track follows in the table below.

Table 2

TRACK	PARTICIPANTS (PARTICIPANT ROLE)	DESCRIPTION
Initial Processing Track	<ul style="list-style-type: none"> Steinbeis Consortium (Building Energy Processing Service D126) interactive instruments (Building Energy Data Service D123) Ecere (External Geospatial Data Service D127) 	The initial processing track primarily focused on the development and implementation of different OGC compliant Web API services that attributed energy consumption to buildings given the provided source geospatial building characteristics data. As part of the track, a minimal viable product for building energy simulation using a given geospatial dataset as input was implemented for Nuns' Island in Montreal, Canada. Energy simulation results were attributed to buildings in the Nuns' Island dataset and offered to clients as a data model through Web API implementations.
Extended Processing Track	<ul style="list-style-type: none"> Steinbeis Consortium (Building Energy Processing Service D126) interactive instruments (Building Energy Data Service D123) 	The extended processing track focused on a more generalized geospatial processing workflow using data from Montreal, Canada. While track one aimed to stand up services using only the available datasets (without dataset conflation), track two aimed to generalize that workflow to the entire Montreal geographic area. The conflation of the provided geospatial building and building characteristics datasets into a common data model and its provision to clients using Web API implementations was a key activity within this track. The track also provided building archetypes for the Province of Quebec through a standards compliant Web API implementation. Unlike the initial processing track, the extended processing track did not attribute energy consumption to buildings in Montreal.
Data Visualization Track	<p>(all Building Energy Client D124 and D125)</p> <ul style="list-style-type: none"> interactive instruments Steinbeis Consortium Climative Properate GeoSolutions 	The data visualization track documented the workflows towards consuming the outputs of the initial and extended processing tracks and displaying them through novel and innovative visualization tools. These visualization tools were a demonstration of how the utility and municipal planning use cases could be achieved when supported by underlying Web APIs.

TRACK	PARTICIPANTS (PARTICIPANT ROLE)	DESCRIPTION
	<ul style="list-style-type: none"> Ethar Inc. 	

Expanding upon the above tracks of the experimentation summary table, the details of the activities performed by each participant within each track of experimentation are shown in the figures and tables below. The figures show the dataset (and processing) workflows performed within each track. The tables summarize, for each of the Testbed-18 participant activities (and their roles), the data used, the services provided (or used), and a general description of the activities performed within each track, as appropriate.

I.G. Initial Processing Track Summary

The initial processing track primarily focused on the development and implementation of different geospatial and building energy web services given the available data for Nuns' Island, Montreal, Canada as show in the following figure and table describing the activities.

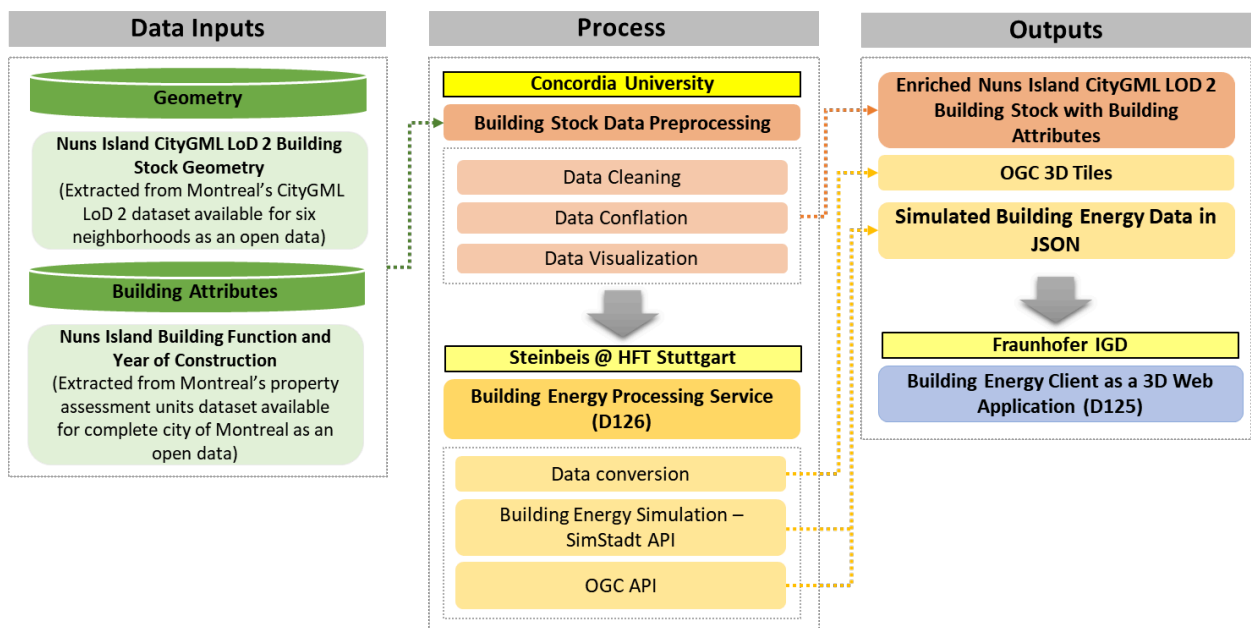


Figure 1 – The data conflation and example processing workflows for the Initial Processing Track.

Table 3

PARTICIPANTS (PARTICIPANT ROLE)	DATA AND SERVICES	DESCRIPTION
Steinbeis Consortium	<ul style="list-style-type: none"> City of Montreal 3D buildings 2016 	The Nuns' Island CityGML LoD 2 dataset from the City of Montreal

PARTICIPANTS (PARTICIPANT ROLE)	DATA AND SERVICES	DESCRIPTION
(Building Energy Processing Service – D126)	<ul style="list-style-type: none"> • Montreal agglomeration property assessment roll data • Cleaned and enriched City of Montreal 3D buildings 2016 CityGML LoD2 dataset for Nuns' Island • SimStadt API with OGC API – Processes API implementation 	<p>was cleaned and enriched with additional building characteristic information from the Montreal agglomeration property assessment roll dataset (building function, year of construction, building category, neighborhood name). A Simstadt energy simulation engine API implementing the OGC API – Processes interface was developed and deployed to attribute energy usage to buildings.</p>
interactive instruments (Building Energy Data Service – D122 and D123)	<ul style="list-style-type: none"> • Cleaned and enriched City of Montreal 3D buildings 2016 CityGML LoD2 dataset for Nuns' Island • SimStadt API with OGC API – Processes API implementation • Nuns' Island building energy provided by OGC API – Features compliant Web API. • Web API provided resources (3D Tiles, Mapbox Vector Tiles, and Features in a number of formats) 	<p>An OGC API – Features compliant API implementation instance was deployed that further enriched the Nuns' Island CityGML LoD 2 dataset with the building energy simulation results from the Building Energy Processing Service for Nuns' Island. The applicability of CityGML Energy ADE and other data formats such as Mapbox Vector Tiles, FlatGeobuf, and glTF for the storing of the produced energy attributes was explored.</p>
Ecere (External Geospatial Data Service – D127 and D128)	<ul style="list-style-type: none"> • Cleaned and enriched City of Montreal 3D buildings 2016 CityGML LoD2 dataset for Nuns' Island • OGC API – GeoVolumes compliant API implementation serving 3D Tiles bounding volume hierarchy, and implicit tiles (OGC Two-Dimensional Tile Matrix Set and Tile Set Metadata). • Virtual collections augmented with calculated attributes were provided with support for OGC API – Features. 	<p>3D geometry for Nuns' Island was made available through an implementation instance of the draft OGC API – GeoVolumes specification. The geometry was served as a 3D Tiles bounding volume hierarchy and implicit tiles. The attributes from the CityGML dataset were populated into a GNOSIS data store and made available through an implementation instance of the OGC API – Features specification. Collections were created using an implementation instance of the draft OGC API – Processes – Part 3: Workflows & Chaining specification and an adapter process to connect to an energy processing service that implements only the OGC API – Processes – Part 1: Core Standard.</p>

I.H. Extended Processing Track Summary

The extended processing track focused on a more general geospatial dataset processing workflow using data from Montreal, Canada. The following figure and table describe the activities.

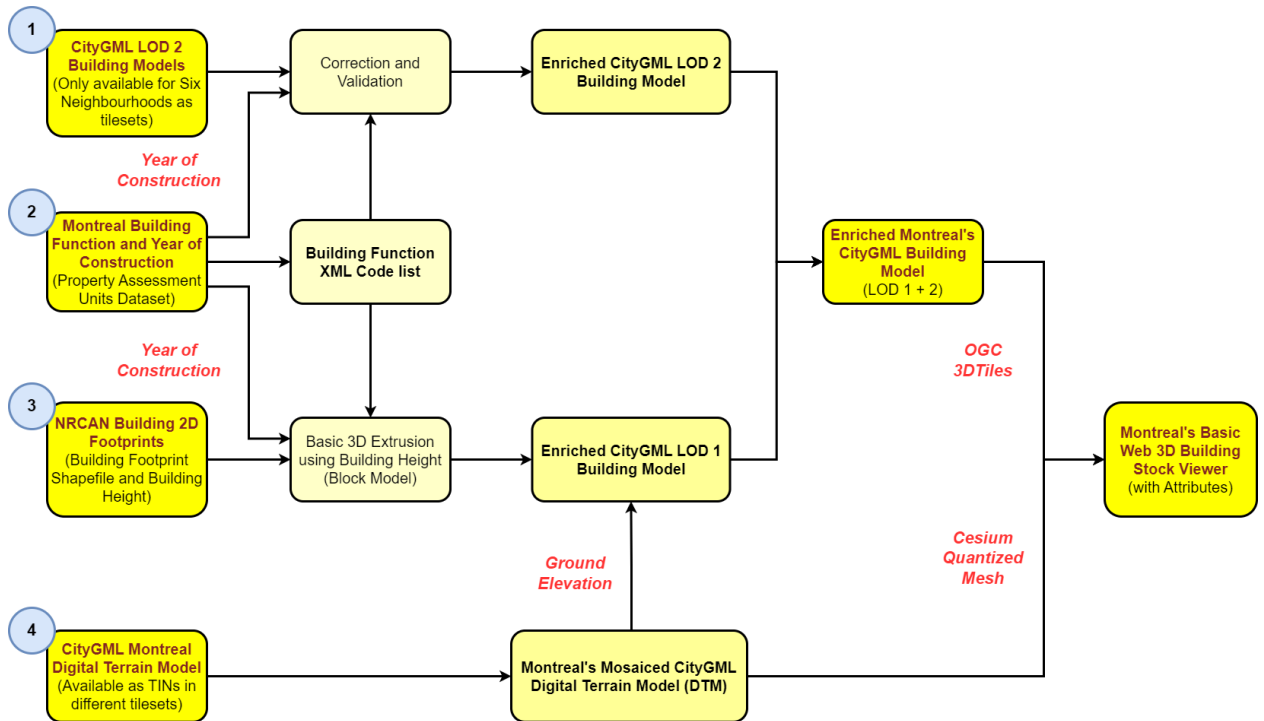


Figure 2 – Data conflation workflow for Extended Processing Track.

Table 4

PARTICIPANTS (PARTICIPANT ROLE)	DATA AND SERVICES	DESCRIPTION
Steinbeis Consortium (Building Energy Processing Service -D126)	<ul style="list-style-type: none"> City of Montreal 3D buildings 2016 (detailed textured building geometries in CityGML LoD2 for the Côte-des-Neiges-Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun neighborhoods) Automatically Extracted Buildings from NRCAN (2D footprint shapefile, Building Height) Montreal agglomeration property assessment roll data and codification of use (building function, year of construction) City of Montreal digital terrain model (DTM) Hybrid CityGML LoD 1 & 2 dataset for the entirety of Montreal 	<p>The activity built upon the Steinbeis Consortium's work on the Nuns' Island dataset during the Initial Processing Track but expanded it to cover the entire City of Montreal. Inputs from the City of Montreal CityGML LoD 2 (6 boroughs), Montreal agglomeration property assessment roll, Automatically Extracted Buildings from NRCAN, and LiDAR data (Montreal DTM) data were conflated to synthesize a hybrid</p>

PARTICIPANTS (PARTICIPANT ROLE)	DATA AND SERVICES	DESCRIPTION
interactive instruments (Building Energy Data Service – D122 and D123)	<ul style="list-style-type: none"> Hybrid CityGML LoD 1 & 2 dataset for the entirety of Montreal Reduced HOT2000 File Representation (derived from Housing Technology Assessment Platform (HTAP) archetype files) OGC API – Features compliant API implementation instance that provided resources (3D Tiles, Mapbox Vector Tiles, and Features in a number of data encodings) OGC API – Features compliant API implementation instance serving Reduced HOT2000 File Representation in GeoJSON and CSV encodings 	<p>CityGML LoD 1 & 2 dataset for the entirety of Montreal.</p> <p>The activity provided an OGC compliant Web API to access the City of Montreal hybrid CityGML LoD 1 & 2 dataset. An API implementation was provided for accessing City of Montreal building data as 3D Tiles, Vector Tiles, and Features with multiple data encodings. An OGC API – Features compliant Web API implementation also provided access to NRCan housing archetypes taken from the HTAP Archetype Database for the Province of Quebec and offered in Reduced HOT2000 File Representation data format.</p>

I.I. Data Visualization Track Summary

The data visualization track documented the workflows that consumed the outputs of initial and extended processing tracks, displaying them through novel and innovative visualization tools.

Table 5

PARTICIPANTS (PARTICIPANT ROLE)	DESCRIPTION
interactive instruments (Building Energy Client D124 and D125)	<p>The buildings in the Nuns' Island dataset were joined with the results of the energy simulation provided by the Building Energy Processing Service (yearly and monthly heating demand as well as the specific space heat demand). The information was visually presented as styled buildings rendered in Cesium JS and MapLibre GL JS. Cesium JS was used to render the feature data provided as 3D Tiles 1.1 and glTF 2.0; MapLibre GL JS was used to render feature data provided as Mapbox Vector Tiles or GeoJSON using the MapLibre Style Specification.</p>

**PARTICIPANTS
(PARTICIPANT
ROLE)**

DESCRIPTION

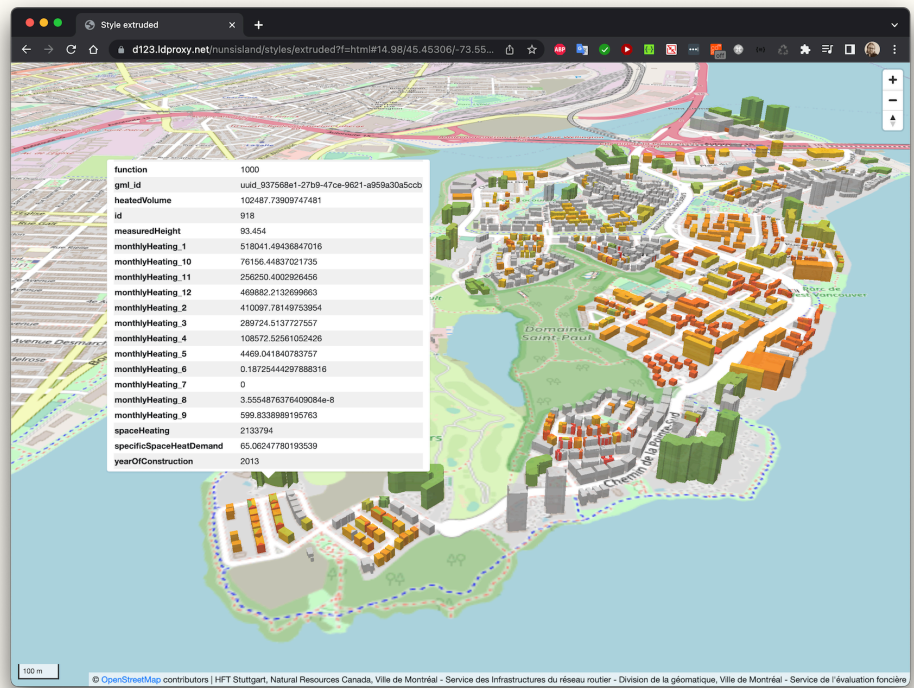


Figure 3 – Map with the buildings color-coded based on the specific space heating demand

Steinbeis Consortium
(Building Energy
Client D124 and
D125)

The Building Energy Client (D125) web application enabled exploration of existing geospatial and building datasets available through the Building Energy Data Services for Nuns' Island. The web application provided a geospatial visualization of 3D GeoVolumes in 3D Tiles format on a 3D globe based on Cesium JS. Furthermore, the client web application supported starting a building energy simulation using the Building Energy Processing Service (D126). Building energy attributes selected by the user within the application interface were simulated on the fly by the processing service and the result was returned to the client. The client conflated the results with the selected geospatial dataset containing the building geometry. Styling was then applied to the map visualization. Within the client application, clicking on an individual building displayed existing and simulated attributes for that building.

**PARTICIPANTS
(PARTICIPANT
ROLE)**

DESCRIPTION



Figure 4 – Result visualization in Building Energy Client D125 Steinbeis Consortium

Climative (Building Energy Client D124 and D125)

The goal for Testbed-18 client application development was to evaluate the capabilities of the Building Energy Data Service (D122) and External Geospatial Data Service (D127 and D128) to provide standardized access to both EnerGuide Rating system data and individual building characteristics. These results were not achieved within Testbed-18.

Properate (Building Energy Client D124 and D125)

The Client (D124) used the Building Energy Processing Service (D126) to acquire data. The input to the Client was a user defined geographic bounding box with four coordinates, entered through the client application interface. The client application then constructed a query for the Processing Service to perform an energy simulation for all buildings within the bounding box. The API response contained a list of buildings each with a nested attribute of spaceHeating. The client application displayed the results to the user within the interface as an average of the spaceHeating attributes for all buildings contained within the selected bounding box.

GeoSolutions (Building Energy Client D124 and D125)

The client application developed by GeoSolutions is a frontend-only single page application based on the MapStore framework with custom components that focus on the following aspects.

- Exploration and visualization of 3D data listed inside an OGC API Features collection.
- Style classification of features available in a collection mixing different content such as items, 3d tiles, and query-able data.

The client supported the inclusion of 3D Tiles data with a configured Catalog plugin provided out of the box by MapStore. The Catalog plugin User Interface supported adding new services including the tile set JSON href that is the entry point of a 3D Tiles layer.

**PARTICIPANTS
(PARTICIPANT
ROLE)**

DESCRIPTION

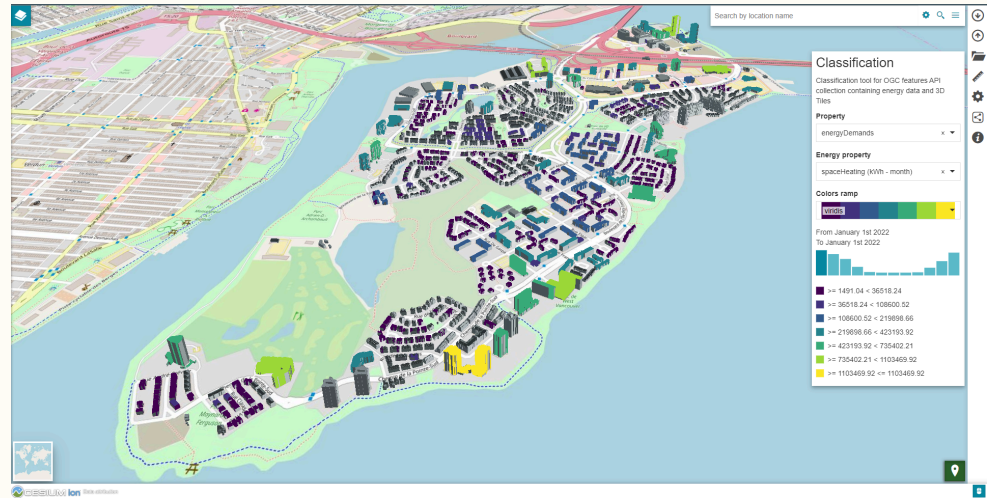


Figure 5 — Style classification based on monthly energy demand property applied to the interactive instruments's Nun's Island collection visualized in 3D inside a MapStore custom application

Ethar Inc. (Building Energy Client D124 and D125)

The objective of the Ethar augmented reality (AR) client application was to color-code 3D building polygons based on energy-use data associated with individual buildings, ideally offering several time-domain datasets filterable by building features, e.g., building height, year of construction, etc. In keeping with the OGC's reason for being, a key requirement for the project was to utilize or create open-source tools and code wherever possible. The following visualization modes were produced as part of the Testbed-18 work.

- A WebGL client implementation was produced to illustrate the Nuns' Island structures and allow manipulation of various energy parameters and render styling.
- Variables available through the drop-down in the upper right corner are listed below and as each variable option is selected, the 3D buildings on the map are rendered in the appropriate color.
 - Heated Volume
 - Monthly Heating
 - Space Heating
 - Specific Space Heat Demand
 - Year of Construction

PARTICIPANTS
(PARTICIPANT
ROLE)

DESCRIPTION

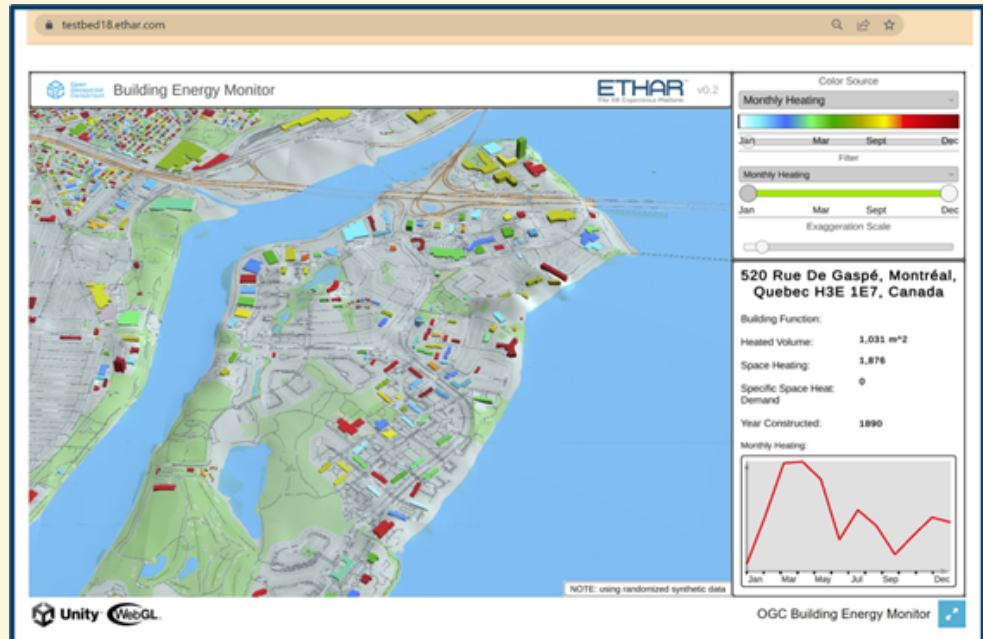


Figure 6 – Ethar WebGL Client Desktop Visualization

- The WebGL functionality from the desktop client was migrated into an AR device client application. The rendering was designed to spawn a 3D colorized map of terrain, natural and created features, basemap roads and waterways, and color-coded building models based on selected building energy variables. A 2D printed icon served as a “trackable” anchor for the rendering.
- The 3D AR data to be visualized was designed to be accessed using a handheld Android smartphone or tablet and a prototype .apk was produced.



Figure 7 – Ethar WebGL Client Mobile Android Visualization

Because building energy data was not available via servers for the majority of the Testbed-18 period of performance, the visualizations were mockups using simulated buildings and energy data created by Ethar. While the applications were not mature enough to work with live data, the mockups demonstrated the value of using GeoAR visualization for building energy analysis.

I.J. Testbed challenges, lessons learned, and recommendations.

Through the Testbed-18 activities that worked towards the Testbed goals there were challenges, lessons learned, and recommendations identified. Several of these key items are detailed in the following subsections.

I.J.1. Data, standards, and interoperability

Data quality, standards compliance, useability, and availability could be described as key issues within the Testbed. Data provided for use in the Testbed experiments did not meet the requirements of making location information and services FAIR – Findable, Accessible, Interoperable, and Reusable. **A key recommendation from the Testbed activities is to make buildings and energy datasets, whether explicitly or implicitly geospatial, strive to meet FAIR (Findable Accessible Interoperable, Reusable) requirements.**

The 3D volumes and building characteristics datasets, like the CityGML and property roll assessment data from the City of Montreal, were provided in a useable format but generally did not contain enough data for the type of energy analysis the Testbed goals aimed to address. They also did not completely comply with the standards formats in which they were provided and their availability varied by geographic region as well. These factors all led to the datasets requiring extensive cleaning and refactoring before they could be conflated together. **A key recommendation is to ensure provided datasets are offered by maintainers or owners in formats that comply with open and well-documented standards, such as those offered by OGC.**

Between different datasets, sometimes the technical nuance of parameter or attribute matching presented a barrier to dataset usage for those who may not be an expert in the dataset domains (i.e., geospatial software development, building science, and energy performance). This was especially observed within the Testbed when geospatial software developers tried to use Canadian building characteristics and building energy data. Compounding this challenge, building characteristic datasets were not formatted to easily allow conflation with other related datasets and did not contain enough data to be useful in a broader geographic context (provincially or nationally in Canada). Inconsistent naming of attributes, schemas, and data ontologies that connect the datasets and terms within them to each other presented significant technical and administrative barriers to data conflation. Therefore, there is a strong need for coherency in the provision of building stock data and for accelerating efforts to provide such data in a standardized and portable format. **A key recommendation within Canada is that a national push towards semantic data interoperability would remove barriers to 3D building volumes and building characteristics data conflation and enabling interoperability with energy data, accelerating progress towards emission reduction targets set by all levels of government and socio-economic policy objectives.**

Generally, the Testbed participants found low-rise housing related building characteristics and energy datasets hard to use and conflate with the other (geospatial) Testbed data. In the case of some datasets these barriers completely prevented the use of a dataset by the geospatial experts within the Testbed. These comments applied to both EnerGuide for housing and the HTAP Housing Archetypes datasets. **A key recommendation is that access to documentation for these datasets should be made publicly available in the form of data dictionaries, diagrams, and technical guidance providing information on how the data could be used beyond a building energy simulation context – ideally referencing or working towards adhering to geospatial standards concepts.**

I.J.2. Practices, collaboration, and communication

The practices, collaboration, and communication used both internally and externally at organizations providing datasets and/or creating guidelines and standards was also an indirect challenge within the Testbed. The indirect challenge was that the often-siloed nature of these organizations' activities led to semantic barriers in using the datasets and standards together for the purpose of accomplishing the Testbed's goals. In most cases, different organizations and governments were unaware of standards or practices in other domains that would have helped reach the Testbed goals. More work can be done to bridge silos in government, standards organizations, and industry to enable building related data interoperability and reduce duplication of effort. **Key recommendations in this area are to:**

- **conduct stakeholder consultation activities to present available datasets and identify missing data within relevant datasets;**
- **share information about relevant standards formats that could increase the potential for data interoperability; and**
- **governments in particular and all organizations should demonstrate leadership by requiring or encouraging database maintainers and owners to provide datasets in well-documented, standardized, and interoperable formats.**

I.K. Conclusions

Regarding the Testbed-18 goals, participants were able to accomplish the following.

- Leveraging existing building energy datasets to evaluate specific energy and geospatial interoperability requirements.
- Exploring draft data models and associated schemas for key building energy datasets.
- Extracting available attributes from a map for individual buildings to feed into building energy simulations.
- Demonstrating the potential of existing geospatial standards to support access and use of building energy data and their interoperability with other geospatial information.
- Demonstrating the potential of existing or new OGC standards to support building energy data interoperability and recommending future standards development activities to fully implement an Energy SDI.

However, these goals were not completed in full for every provided data source. It was clear from the Testbed activities that OGC standards compliance enabled use of some datasets (such

as the CityGML data from Montreal) but lack of compliance and documentation hindered the use of others (NRCan provided building characteristics and energy datasets).

Regarding Testbed-18 use case goals, through the adaptation of existing OGC Standards to the available energy and geospatial data, Testbed participants were able to explore the Housing Retrofit Program Planning and Provision of building stock statistics use case goals in support of contributing towards a Canadian Energy SDI. The Testbed participants were not able to address the Utility Hosting Capacity Analysis use case within the Testbed due to time and data constraints.

Testbed-18 was able to show promise in the application of standards to the problem of geospatial, building characteristics, and building energy data interoperability for the use cases put forward in the Testbed. The Testbed also demonstrated that standardized Web APIs foster an innovative and open environment where software providers from different domains (e.g., geospatial, building energy) developed modular software services and client applications that exchanged data and formed parts of large and feature-rich applications. This standardized, modular application format encourages economic development and innovation by allowing smaller, often subject matter expert developers, to interact with a vast ecosystem of other stakeholders to create innovative and useful software platforms and grow their technical and organizational capacities. This was shown in the Testbed where organizations drew upon their strengths and contributed modularized components to the full stack of services and clients needed to reach the Testbed-18 goals. The standardized, modular application format also provides the ability to isolate and verify the components of larger software platforms allowing easier process validation and leading to more accuracy and transparency within the vertical geospatial and building energy software stack. The Testbed therefore formed a foundation for future investigations into the converging fields of geospatial and building energy analyses.



KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, building energy modeling, urban building energy modeling, building energy mapping, building energy data interoperability, energy spatial data infrastructure



PREFACE

Within the context of annual global energy consumption, buildings are one of the most significant consumers. The 2022 Global Status Report for Buildings and Construction[1] found that the building and construction sector accounted for more than 34% of energy demand and around 37% of energy and process-related CO₂ emissions in 2021[2]. According to the Canada Green Buildings Strategy, the built environment represents Canada's third-highest source of emissions[3]. Direct fossil fuel combustion in and energy consumption of buildings (taking into account the GHG emissions from the generation of electricity) accounted for 13% and 18%, respectively, of Canada's greenhouse gas (GHG) emissions in 2020[4]. In the United States, total energy consumption by residential and commercial buildings, including end-use consumption and electricity, was 28% of total U.S. energy consumption in 2021 and rose to 39% if electrical system energy losses were included[5]. To address climate change goals, building energy usage and the associated emissions must be reduced globally.

Regionally focused building related analyses are important in arriving at data driven climate change policy. Policymakers need simple tools that enable them to rapidly assess how new or existing building related climate change policies will impact the energy and emissions profiles of the building stock within specific geographic regions (i.e. a city, or a province). These tools need to accurately conflate geospatial and building energy datasets within a single interface to provide the information policymakers need to arrive at effective building related climate change policies. The geospatial and building energy datasets needed to build these tools are available globally, but the datasets exist in different political, data, and expertise domains. Without tools to help them, policymakers are left to the onerous task of manually conflating the datasets for analyses. In many jurisdictions, to help quantify the energy usage of individual buildings, databases of building energy ratings have been collected. They often form part of wider regional and national rating systems and strategies to reduce energy usage, GHG emissions, and achieve economic outcomes, but are generally not often integrated or interoperable with geospatial datasets and tools. The lack of interoperability and integration between geospatial and energy rating datasets prevents the evaluation of metrics that require data from both sources, such as the evaluation of the relationships between energy poverty and regional socio-demographics. The same lack of interoperability amongst geospatial and building energy datasets also exists across other energy datasets collected by government, utilities, and industry. Lack of interoperability among these organizations' data formats results in duplication of effort, lost potential for energy savings, and lost opportunities for effective policy tools that are needed to help address energy use and subsequently climate change mitigation and resilience.

Both the OGC and NRCan have worked in the past to characterize gaps in data integration and interoperability of geospatial and building energy data through studies and standards development. The [Building Energy Mapping and Analytics Concept Development Study \(BEMA CDS\)](#) was undertaken by the OGC, NRCan, and stakeholders from 2020 to 2021 is an example of that work.

The purpose of BEMA CDS was to:

- characterize the state of development of energy mapping and analytics for building stock broadly; and

- inform and propose IT architectural practices and standards to enable mapping and analytics specifically of residential energy use and efficiency.

The outcomes of the study include challenges that building energy data and applications face, as well as opportunities to address those challenges with improved technologies, data sharing practices, and improved understanding of the benefits that could result.

Another example of an activity that aimed to bridge the interoperability gap between geospatial and energy data was the development of the Energy ADE for CityGML. The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations [article](#) discusses the providence and use cases of the ADE and is an earlier example of how the pathway towards geospatial and building energy data interoperability has developed. The Energy ADE also serves as a data format upon which policymakers can base current and future energy data related initiatives to enable native geospatial interoperability.

The work of the Building Energy Interoperability task within Testbed-18 builds upon the Energy ADE for CityGML and the outcomes of the BEMA-CDS. Of the challenges identified in the BEMA-CDS, the Testbed task aimed to address the following.

- Repetitive non-standardized methods are applied to collection, exchange, and integration of datasets.
- Data source methods and confidence are wide ranging and poorly documented, variously measured, modeled, inferred, estimated, assumed, etc.
- Lack of an overall data framework prevents connecting the scale and resolution of spatial data to particular use scenarios.
- It remains a challenge to connect archotyping methods (clustering / classification) with different use case scenarios.

Testbed-18 work addressed the following opportunities identified within the BEMA-CDS.

- National building data layer for comprehensive analysis of building types, energy performance, retrofit / upgrade technologies, costs, and benefits.
- National systems for consistent energy data at multiple scales.

Additionally, the activities conducted within Testbed-18 aimed to build on the development of the Energy ADE and the findings of the BEMA-CDS to continue working towards the creation of an Energy Spatial Data Infrastructure (E-SDI) architecture. Finally, the Testbed considered future work recommendations from the BEMA-CDS and addresses the following.

- Sandbox activities, such as interoperability pilots, modeling the mutual benefits of information sharing and data interoperability.
- Prototypes for an Energy SDI, demonstrating common availability of such technologies as cloud-based energy modeling, model-driven building archetypes, and enclave protocols for addressing data privacy and propriety constraints.



SECURITY CONSIDERATIONS

No security considerations have been made for this document.



SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Volta Research Inc
- Properate
- Ecere
- Climative
- Ethar Inc
- GeoSolutions
- interactive instruments
- Steinbeis Consortium



SUBMITTERS

All questions regarding this submission should be directed to the editor or the submitters:

NAME	AFFILIATION
Leigh St. Hilaire	Volta Research Inc
Aidan Brookson	Volta Research Inc
Jérôme Jacovella-St-Louis	Ecere
Patrick Dion	Ecere
Volker Coors	Steinbeis Consortium, HFT Stuttgart
Thunyathep Santhanavanich (Joe)	Steinbeis Consortium, HFT Stuttgart
Rushikesh Padsala	Steinbeis Consortium, Concordia University
Sanam Dabirian	Steinbeis Consortium, Concordia University

NAME	AFFILIATION
Mostafa M. Saad	Steinbeis Consortium, Concordia University
Ursula Eicker	Steinbeis Consortium, Concordia University
Maxim Rossknecht	Steinbeis Consortium, Fraunhofer IGD
Tony Hodgson	Ethar
Stefano Bovio	GeoSolutions
Clemens Portele	interactive instruments
Winston Morton	Climative (formally SimpTek Technologies)
Arman Mottaghi	Properate



ACKNOWLEDGEMENTS

The editors would like to acknowledge the invaluable support, advice, contributions, and encouragement from Jessica Webster and Ryan Ahola of Natural Resources Canada.

1

SCOPE



SCOPE

This OGC Testbed-18 Engineering Report (ER) represents deliverable D012 and D013 for the Building Energy Data Interoperability task.



2

NORMATIVE REFERENCES

NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Open API Initiative: **OpenAPI Specification 3.0.2**, 2018 <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.2.md>

van den Brink, L., Portele, C., Vretanos, P.: OGC 10-100r3, **Geography Markup Language (GML) Simple Features Profile**, 2012 http://portal.opengeospatial.org/files/?artifact_id=42729

W3C: **HTML5**, W3C Recommendation, 2019 <http://www.w3.org/TR/html5/>

Schema.org: <http://schema.org/docs/schemas.html>

R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee: IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*. RFC Publisher (1999). <https://www.rfc-editor.org/info/rfc2616>.

E. Rescorla: IETF RFC 2818, *HTTP Over TLS*. RFC Publisher (2000). <https://www.rfc-editor.org/info/rfc2818>.

G. Klyne, C. Newman: IETF RFC 3339, *Date and Time on the Internet: Timestamps*. RFC Publisher (2002). <https://www.rfc-editor.org/info/rfc3339>.

M. Nottingham: IETF RFC 8288, *Web Linking*. RFC Publisher (2017). <https://www.rfc-editor.org/info/rfc8288>.

H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub: IETF RFC 7946, *The GeoJSON Format*. RFC Publisher (2016). <https://www.rfc-editor.org/info/rfc7946>.

Clemens Portele, Panagiotis (Peter) A. Vretanos, Charles Heazel: OGC 17-069r3, *OGC API – Features – Part 1: Core*. Open Geospatial Consortium (2019). <https://docs.ogc.org/is/17-069r3/17-069r3.html>.

Clemens Portele, Panagiotis (Peter) A. Vretanos, Charles Heazel: OGC 17-069r4, *OGC API – Features – Part 1: Core corrigendum*. Open Geospatial Consortium (2022). <https://docs.ogc.org/is/17-069r4/17-069r4.html>.

Thomas H. Kolbe, Tatjana Kutzner, Carl Stephen Smyth, Claus Nagel, Carsten Roensdorf, Charles Heazel: OGC 20-010, *OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard*. Open Geospatial Consortium (2021). <https://docs.ogc.org/is/20-010/20-010.html>.

Steve Liang, Tania Khalafbeigi, Hylke van der Schaaf: OGC 18-088, *OGC SensorThings API Part 1: Sensing Version 1.1*. Open Geospatial Consortium (2021). <https://docs.ogc.org/is/18-088/18-088.html>.

- Carl Reed, Tamrat Belayneh: OGC 17-014r9, *OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package (*.slpk) Format Community Standard Version 1.3*. Open Geospatial Consortium (2023). <https://docs.ogc.org/cs/17-014r9/17-014r9.html>.
- Patrick Cozzi, Sean Lilley: OGC 22-025r4, *3D Tiles Specification*. Open Geospatial Consortium (2023). <https://docs.ogc.org/cs/22-025r4/22-025r4.html>.
- ISO/IEC: ISO/IEC 12113:2022, *Information technology – Runtime 3D asset delivery format – Khronos glTF™ 2.0*. International Organization for Standardization, International Electrotechnical Commission, Geneva (2022). <https://www.iso.org/standard/83990.html>.
- Hugo Ledoux: OGC 20-072r2, *OGC CityJSON Community Standard 1.0*. Open Geospatial Consortium (2021). <https://docs.ogc.org/cs/20-072r2/20-072r2.html>.
- Benjamin Pross, Panagiotis (Peter) A. Vretanos: OGC 18-062r2, *OGC API – Processes – Part 1: Core*. Open Geospatial Consortium (2021). <https://docs.ogc.org/is/18-062r2/18-062r2.html>.
- Aguiaro, G., Benner, J., Cipriano, P. et al. The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations. *Open geospatial data, softw. stand.* 3, 2 (2018). <https://doi.org/10.1186/s40965-018-0042-y>
- ISO: ISO 16739-1:2018, *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema*. International Organization for Standardization, Geneva (2018). <https://www.iso.org/standard/70303.html>.
- Jeff Harrison, Ignacio Correias, Jerome Jacovella-St-Louis: OGC 22-029, *DRAFT OGC API – 3D GeoVolumes*. Open Geospatial Consortium (2023). <https://opengeospatial.github.io/ogcna-auto-review/22-029.html>
- Joan Masó, Jérôme Jacovella-St-Louis: OGC 17-083r4, *OGC Two Dimensional Tile Matrix Set and Tile Set Metadata*. Open Geospatial Consortium (2022). <https://docs.ogc.org/is/17-083r4/17-083r4.html>.
- Joan Masó, Jérôme Jacovella-St-Louis: OGC 20-057, *OGC API – Tiles – Part 1: Core*. Open Geospatial Consortium (2022). <https://docs.ogc.org/is/20-057/20-057.html>.
- Panagiotis (Peter) A. Vretanos, Clemens Portele: OGC 19-079, *DRAFT OGC API – Features – Part 3: Filtering and the Common Query Language (CQL)*. Open Geospatial Consortium (2020). <https://portal.ogc.org/files/96288>
- Panagiotis (Peter) A. Vretanos, Clemens Portele: OGC 21-065, *DRAFT Common Query Language (CQL2)*. Open Geospatial Consortium (2022). <https://docs.ogc.org/DRAFTS/21-065.html>
- Clemens Portele, Panagiotis (Peter) A. Vretanos: OGC 21-045, *OGC Features and Geometries JSON – Part 1: Core*. Open Geospatial Consortium (2022). <https://docs.ogc.org/DRAFTS/21-045.html>

Panagiotis (Peter) A. Vretanos, Tom Kralidis, Charles Heazel, Angelos Tzotsos: OGC 20-004, DRAFT OGC API – Records – Part 1: Core. Open Geospatial Consortium (2022). <http://docs.ogc.org/DRAFTS/20-004.html>

Clemens Portele, Panagiotis (Peter) A. Vretanos: OGC 18-058r1, OGC API – Features – Part 2: Coordinate Reference Systems by Reference corrigendum. Open Geospatial Consortium (2022). <https://docs.opengeospatial.org/is/18-058r1/18-058r1.html>.

3

TERMS, DEFINITIONS AND ABBREVIATED TERMS

TERMS, DEFINITIONS AND ABBREVIATED TERMS

This document uses the terms defined in [OGC Policy Directive 49](#), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications ([OGC 08-131r3](#)), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

3.1. OGC Web API

A Web API that implements one or more Conformance Classes from an OGC API Standard

3.2. Building archetype

A building archetype is a sample or virtual building that is characteristic for buildings of the same kind. The use of building archetypes for building related analyses allows the analytical activity of interest to be completed faster by using building archetypes as representations of a larger dataset of specific building characteristics. Building archetypes are often provided in file formats that allow their use within one or more software platforms. Building characteristics that are represented through the parameters contained within the archetype file and are appropriate for their analytical end use.

Example The concept of archetypes as the basis for modelling thermodynamic building performance, the ability to assess energy end-use, and undertake performance optimization that represent regional building stock. The Canadian bungalow building archetype in HOT2000 file format represents the characteristics of all single-story detached residential buildings in Canada and can be used for energy simulation exercises using HOT2000 that are applicable to all buildings that conform to the bungalow archetype.

3.3. Building archetype library

A building archetype library is a representative set of **building archetypes** stored in a database or common storage medium and collated or intended for specific use.

Example NRCan CanmetENERGY's **Housing Technology Assessment Platform** includes a library of building archetypes in HOT2000 file format intended that represent the Canadian low-rise building stock.

3.4. BTAP: Building Technology Assessment Platform

Building Technology Assessment Platform (BTAP) is a framework being developed by Natural Resources Canada (NRCan) to assist in the analysis of the energy performance of technologies used in commercial or other buildings that are governed by Part 3 of the National Building Code of Canada.

3.5. HOT2000 Software suite

HOT2000 is an energy simulation modeling software developed and maintained by Natural Resources Canada (NRCan) to support the EnerGuide Rating System, ENERGY STAR for New Homes, and R-2000 residential energy efficiency initiatives. The software suite can be obtained from Natural Resources Canada HOT2000 Energy simulation software.

3.6. HTAP: Housing Technology Assessment Platform

Housing Technology Assessment Platform (HTAP) is a building archetype library and a collection of data and tools that automate and extend the HOT2000 residential energy simulation modeling software to enable parametric modeling of residential buildings in Canada.

3.7. CityGML: City Geography Markup Language

The CityGML standard defines a conceptual model and exchange format for the representation, storage, and exchange of virtual 3D city models. It facilitates the integration of urban geodata for a variety of applications for Smart Cities and Urban Digital Twins. The Application Domain

Extension (ADE) is a built-in mechanism of CityGML to augment its data model with additional concepts required by particular use cases. The [CityGML Energy ADE](#) extends the CityGML Standard by features and properties, which are necessary to perform an energy simulation and to store the corresponding results.

3.8. ADE: Application Domain Extension

The Application Domain Extension (ADE) is a built-in mechanism of CityGML to augment its data model with additional concepts required by particular use cases.

3.9. ERS: EnerGuide Rating System

EnerGuide is the official mark of the Government of Canada for its energy performance rating and labeling program for key consumer items – houses, light-duty vehicles, and certain energy-using products. When used in the context of this report, ERS refers to both [EnerGuide rating system for homes](#) and [EnerGuide for home evaluations](#).

3.10. EA: Energy Advisors

Energy Advisors (EA) are professionals in Canada who are trained to evaluate low-rise buildings in accordance with the ERS and are registered with Natural Resources Canada (NRCan) to perform this service. Each evaluation performed by an EA is submitted to NRCan and is recorded in the EnerGuide for Housing (EGH) database as a row entry in the form of a tab separated value (TSV). The TSV entries in the EGH database are available through the [EnerGuide API](#).

3.11. IFC: Industry Foundation Classes

The Industry Foundation Classes (IFC) are an open international standard for Building Information Model (BIM) data that are exchanged among software applications used by the building sector.

3.12. Dataset

A collection of data that has been published or curated in a coordinated way that has been made available for access or download in one or more formats.

3.13. Cesium

Cesium is an open platform for creating 3D geospatial applications. Cesium ion's 3D global content can be combined with point clouds, photogrammetry, BIM, or other 3D data to facilitate app development. A variety of app development environments are supported including JavaScript (CesiumJS), the Unity ecosystem (Cesium for Unity), the Unreal game engine (Cesium for Unreal), the NVIDIA Omniverse (Cesium for Omniverse), and the open source Open 3D Engine (Cesium for O3DE).

3.14. 3D Tiles

3D Tiles is an OGC standard and is designed for streaming and rendering massive 3D geospatial content such as photogrammetry, 3D Buildings, BIM/CAD, instanced features, and point clouds. It defines a hierarchical data structure and a set of tile formats which deliver content.

3.15. Open Asset Import Library (assimp)

A library to import and export various 3D data model formats including scene post processing to generate missing render data.

3.16. Feature collection; Collection

A set of features from a dataset.

Note 1 to entry: In this ER, 'collection' is used as a synonym for 'feature collection.'

3.17. Augmented Reality

Human perception is enhanced with additional computer-generated sensorial input to create a new user experience, including, but not restricted to, enhancing human vision by combining natural with digital offers. (IEEE 1589-2020)

3.18. Mixed reality system

System that uses a mixture of representations of physical world data and virtual world data as its presentation medium. Syn: mixed and augmented reality system. (ISO/IEC 18039:2019).

3.19. GNOSIS

The GNOSIS SDK is an object-oriented SDK written by Ecere in the eC programming language for use in visualizing geospatial data. The GNOSIS Map Server is an experimental API implementation of OGC compliant services.

3.20. USD

Universal Scene Description (USD) is an efficient, scalable system for authoring, reading, and streaming time-sampled scene description for interchange between graphics applications.

3.21. Digital Twin

A digital twin is a virtual representation of real-world entities and processes, synchronized at a specified frequency and fidelity.

3.22. Style

A style is a sequence of rules of symbolizing instructions to be applied by a rendering engine on one or more features and/or coverages.

3.23. Abbreviated terms

API	Application Programming Interface
AR	Augmented Reality
BIM	Building Information Modelling
CAD	Computer Aided Design
CDM	Central Data Model
CGDI	Canadian Geospatial Data Infrastructure
CRS	Coordinate Reference System
DTM	Digital Terrain Model
ER	Engineering Report
ESDI	Energy Spatial Data Infrastructure
ETSI ARF	European Telecommunications Standards Institute (ETSI) Augmented Reality Framework
EUI	Energy Use Intensity
FME	Feature Manipulation Engine
GBXML	Green Building Extensible Markup Language
GHG	Greenhouse Gas
GML	Geography Markup Language
HPXML	Home Performance XML
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers

ISO	International Standards Organization
LoD	Level of Detail
MEUI	Mechanical Energy Use Intensity
MOAW	Modular OGC API Workflows
NRCan	Natural Resources Canada
OGC	Open Geospatial Consortium
PTZR	pan / tilt / zoom / rotate
TEDI	Thermal Energy Demand Intensity
TIN	Triangulated Irregular Network
UBEM	Urban Building Energy Modelling
XML	Extensible Markup Language
XR	Extended Reality
6DOF	Six degrees of freedom



4

INTRODUCTION

4.1. Report overview

This publicly available OGC Engineering Report (ER) documents the OGC Testbed-18 activities and subsequent results and summarizes how geospatial standard-based APIs can enable progress towards building energy and geospatial data interoperability. The ER is organized into sections for each of the two tracks of experimental prototyping activities investigated and the client interfaces presented. The ER concludes with a summary of lessons learned and suggestions for future work.

4.2. Testbed background and use cases

The Canadian Geospatial Data Infrastructure (CGDI) represents Canada's national Spatial Data Infrastructure (SDI). Similar to traditional physical infrastructures that help Canadians with their everyday lives (e.g., roads, utilities, telecommunications), the CGDI is an infrastructure for geospatial (i.e., location) information. In short, the CGDI helps Canadians find, access, use, and share geospatial information to support their social economic activities. The CGDI is built on a foundation of geospatial standards, which allow geographic information to be accessed and used consistently. By using the CGDI, individuals with diverse expertise, working in different domains, with independent software suites, can leverage each other's geographic information in a consistent way.

The development of RESTful Web APIs, enhancements in the representation of 3D building data, and the growing availability of modeled energy data have allowed the extension of some CGDI capabilities to be used within the domain of buildings and energy. The rudimentary extension of energy data and services into the CGDI has highlighted the further potential of the CGDI to serve as a management and analysis tool for Greenhouse Gas (GHG) emissions and energy usage reduction – climate change related – initiatives and to contribute to the development of a sustainable future in Canada. These incremental additions to the CGDI also show some of the benefits possible through the full implementation of Urban Digital Twins in Canada.

For the CGDI to serve as that management and analysis tool, the roadblocks that still exist that have prevented the widespread use of CGDI capabilities relating to building energy in Canada will have to be overcome. A non-exhaustive list of examples of these roadblocks includes a lack of consensus on the need for spatial data integration, lack of understanding of the CGDI concept, and lack of awareness of spatial data interoperability standards, their function, and benefits. The intent of the sponsor for this testbed task was to support activities leading to the removal of barriers limiting the use of SDI in conjunction with Building Energy data in Canada. The task prioritized two potential and important Canadian use cases for the CGDI where barriers have prevented effective use of building energy and geospatial data: municipal planning for

climate change related building retrofit programs and policies; and utility conservation planning and the offsetting of capital costs associated with the maintenance and development of energy transmission and distribution networks. These two cases were determined to be high value because, in Canada, they represent two core players who plan, design, and implement housing policy with regard to energy consumption and emissions, and those that are tasked with then planning how to and supplying energy to those buildings and communities. Providing municipal and utility entities with better planning tools, namely a fully functioning energy and utilities component of the CGDI, would go a long way to enabling them to implement better energy and emissions policy and programs.

4.3. Testbed goals

4.3.1. General goals

Per the NRCan guidance and the [OGC Testbed-18: Call for Participation](#), the goal of Testbed-18 documented in this ER task was to prototype an Energy Spatial Data Infrastructure (ESDI) that could become part of the CGDI and allow for execution of building energy experiments and analysis – ideally using existing OGC Web API standards. To that end, this task undertook initial work towards the development of an Energy SDI. The activities anticipated to take place within the task included the following.

- Leveraging existing building energy datasets to evaluate specific energy and geospatial interoperability requirements.
- Exploring draft data models and associated schemas for key building energy datasets.
- Demonstrating the potential of existing geospatial standards to support access and use of building energy data and their interoperability with other geospatial information.
- Demonstrating the potential of existing or new OGC standards to support building energy data interoperability and recommending future standards development activities to fully implement an Energy SDI.

4.3.2. Data model goals

At the outset of the Testbed, the development of a generalized data model was anticipated based on the following potential scenarios:

- extracting information from building energy models for map attribution; or
- extracting available attributes from a map for dwellings in the stock to feed into building energy simulations.

The goals related to data models within Testbed-18 included documenting the input data structures – both geospatial and energy – to the processing and data visualization tracks. The Testbed-18 workflows that attempted to use OGC Web APIs towards the creation of an interoperable, generalized data model containing geospatial and building energy data were also documented.

4.3.3. Detailed use case goals

At the time of the execution of Testbed-18, mapping and analysis of the energy consumption of buildings was undertaken in Canada by different jurisdictions including local municipalities, energy utilities, and provincial and federal agencies. These analyses, conducted across different geospatial and temporal scales, were often independently performed by each jurisdiction using their own datasets and to address their relevant use cases. Fundamentally, however, the intention of the independent work was the same: understanding of the building stock – the age, number of stories, floor areas, and other characteristics of various building archetypes – and how climate change related programs and policies might impact the energy consumption and emissions of those buildings. Despite this commonality, there was little coordination between these groups resulting in differing methodologies, duplicated effort, lost energy savings, and lost opportunities for decarbonization, climate change mitigation, and climate resilience.

Of the previously mentioned needs, the data models worked on within Testbed-18 focused on enabling the following use cases.

- Housing Retrofit Program Planning
- Electric Utility Hosting Capacity Analysis

Through the adaptation of existing OGC Standards, in particular OGC Web APIs, to the available energy and geospatial data, Testbed-18 participants explored their application to the above use cases in support of contributing towards a Canadian Energy SDI. The above data model use cases are expanded in the following sections.

4.3.3.1. Housing retrofit program planning

Governments and other entities using geospatial tools need to be able to quickly look at the existing building stock within their jurisdiction and assess a variety of different building retrofit scenarios to support climate change related retrofit policy and program development. To do so, they need to collect and parse many different data inputs such as existing energy models, building permits, building characteristics, socio-economic indicators, socio-demographic indicators, and utility consumption. From these inputs they need simple, clear, and accurate metrics of interest presented to them showing the impact of policies and programs at the level of an entire city down to an individual building. Through these types of analyses, entities deciding on land planning and policy would be able to trial a variety of retrofit scenarios on the stock of buildings in their jurisdiction, estimate the results across a suite of metrics, and make informed planning and policy decisions based on those outcomes.

The temporal needs of emissions, energy, or demographic datasets to be considered in house retrofit program analyses varies from decades to hourly intervals depending on the output

metrics required. In most cases, annual data points are deemed sufficient for policy planning with seasonal or monthly data being appropriate for climates with large seasonal variations in the needs of their populations.

The geographic scaling needs of this use case vary from the national to the individual building level with the ability to select known and arbitrary geographic areas existing between those two scales of use. With regard to planning in Canada, those known bounding boxes consisted of established geospatial divisions used by Statistics Canada (administrative boundaries, statistical boundaries, and non-standard boundaries), Natural Resources Canada, Environment and Climate Change Canada, and mixtures of those with or without other bespoke datasets.

Early examples of tools where geospatial and building energy analyses converge are those that are working to enable building energy labeling for low-rise residential buildings in Canada. These tools are often referred to as virtual energy assessment or audit tools and they are a key subset of the housing retrofit planning use case that would be further enabled through data interoperability and standards.

4.3.3.2. Utility hosting capacity analysis

Utility hosting capacity analysis (HCA) is an analytical approach that can help utilities and the jurisdictions they serve understand how the electric grid is geospatially impacted by the connection of new or modification of existing energy sources and sinks. One of the goals of utility HCA is to determine when the connection of or change of use of existing energy sources and sinks will result in the need for asset upgrade and capital expenditures to the electrical distribution system, or conversely, how much extra capacity will be available for use when asset upgrades are completed[6].

By seeking to enable the standards based interoperability of geospatial datasets with utility planning datasets, the activities of Testbed-18 could help utilities and jurisdictions increase their ability to communicate through the exchange of interoperable data. This increase in communication would give jurisdictions and utilities the agility to coordinate the planning and evaluation policies and programs that target building related energy consumption and emissions. Part of this coordination could include the potential to perform policy and program specific HSAs to manage their impact on electrical distribution system or for jurisdictions to deploy regional policies or programs that are aware of the loading characteristics (previous HSA results) on the jurisdictional electrical distribution system.

Some examples of areas in which this type of coordination would be effective include policies or programs that encourage photovoltaic systems, energy storage, residential electrification, or electric vehicle charging stations because their climate change and electrical distribution system impacts are often intertwined at a geospatial level (i.e., jurisdictional programs that would increase load are designed to be deployed within geospatial areas that could receive those new loads, facilitated by interoperable planning-utility geospatial data sharing).

4.4. Testbed and task description

The [OGC Testbed-18: Call for Participation \(CFP\) document](#) section 2.6 Building Energy Spatial Data Interoperability guided the activities conducted during the Testbed-18 task. The evolving needs of the sponsor expressed throughout the Testbed task fine-tuned the requirements defined in the CFP.

The Building Energy Spatial Data Interoperability task participants explored existing data sets available from the geospatial and energy domains in Canada. The participants then developed a mapping and integration approach that aimed to combine the provided and available data sets into a single data model. The intent of the single model was to enable the use of services and/or interfaces compliant with the model data for direct analysis, simulation, and visualization of building energy and GHG metrics without further integration and mapping efforts. The data were served as standard-based Web services that aimed to be compliant with the latest set of OGC Web API standards.

During the Testbed-18 activity, all data models and API functionalities were demonstrated as prototypes that included several services and clients. Each of these services and client roles are described in more detail within the Participant roles section of this report. The objective of this approach was to enable the client applications to integrate various data without additional conversion costs. Different cases involving the Testbed-18 prototype services and clients were explored in Testbed-18 through different tracks. These tracks both allowed parallelization of the tasks within Testbed-18 and the case-by-case analysis of different interactions between the services and clients.



5

TESTBED-18 EXPERIMENTAL ARCHITECTURE

5.1. Architecture background

The Testbed-18 Building Energy Spatial Data Interoperability task's experimental architecture can be described both through the Testbed-18 participant roles and the investigative "tracks" the participants contributed to. Participants were each assigned roles within this Testbed-18 task in order to create a cohesive group of participants with the collective expertise to perform the experiments required to complete the tasks. A brief description of each of the participant roles follows. The Testbed-18 task investigative "tracks" consisted of selected participant roles that were grouped together in order to perform parallel experimental activities and are described in a section of this ER.

5.2. Participant roles

The service and client rolls that each of the Testbed-18 participants were assigned to are described within this section. The service and client roles aimed to represent a vertical stack of modular software components communicating through a common data model. The services and clients could be combined to offer novel insights that addressed the Testbed-18 use cases. The API services offered by each of the participant roles implemented a current, emerging, or newly defined OGC Web API, as applicable. See the figure and tables below.

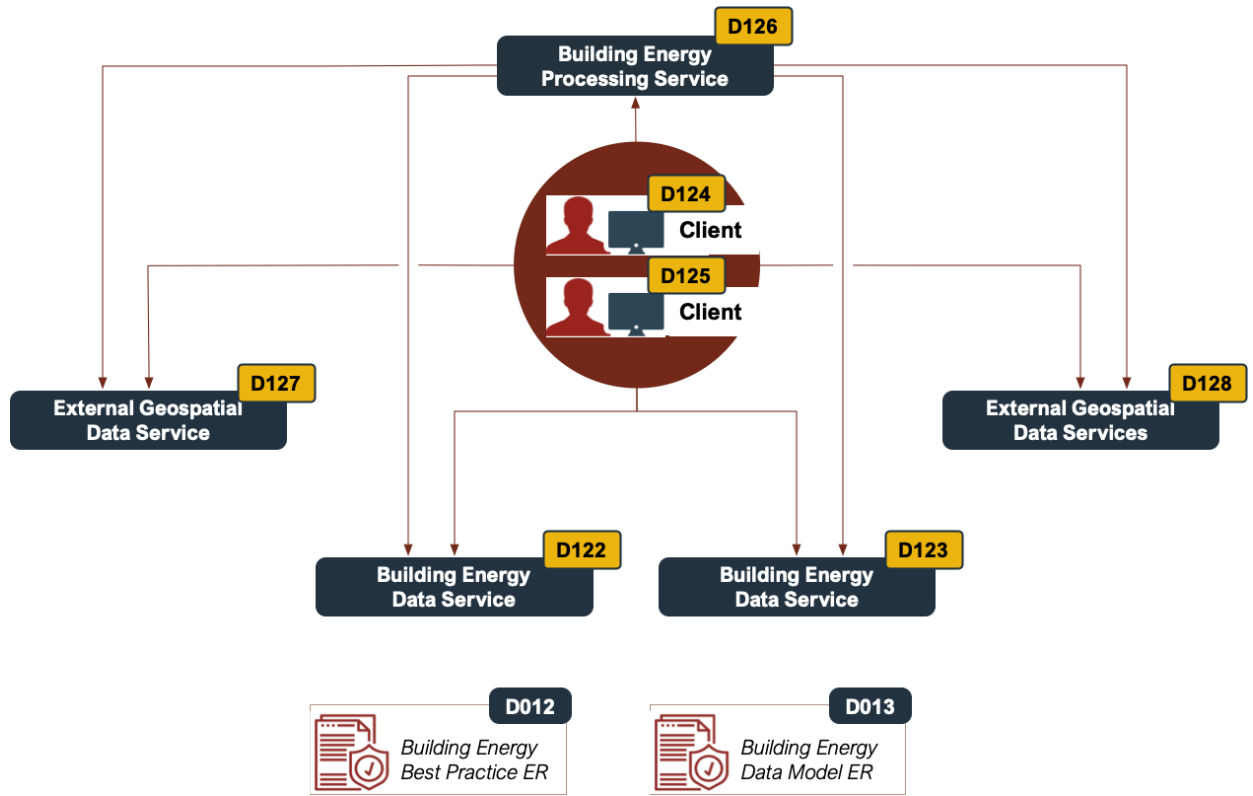


Figure 8 – Testbed-18 participant roles

Table 6

PARTICIPANT ROLE	PARTICIPANTS	DESCRIPTION
Building Energy Data Service (D122 and D123)	<ul style="list-style-type: none"> • Ecere • interactive instruments 	The Building Energy Data Service was a Web API instance that served building energy data according to the data models investigated in this ER.
External Geospatial Data Service (D127 and D128)	<ul style="list-style-type: none"> • Ecere • GeoSolutions 	The External Geospatial Data Service was a Web API instance that provided access to geospatial data that was required for the applications offered by the Building Energy Processing Service (D126).
Building Energy Processing Service (D126)	<ul style="list-style-type: none"> • Steinbeis Consortium 	The Building Energy Processing Service was a Web API instance providing access to applications that make use of the components D122, D123, D127, and D128.
Building Energy Client (D124 and D125)	<ul style="list-style-type: none"> • Ethar • Properate • Steinbeis Consortium • Climative • interactive instruments • GeoSolutions 	The Building Energy Client applications accesses the other Testbed-18 web services and used them to provide stakeholder access to building characteristics and energy data through web and augmented reality interfaces.

5.3. Testbed-18 investigative tracks

To efficiently work towards addressing the goals of the Testbed-18 Building Energy Spatial Data Interoperability task and the use cases put forward by NRCan, the activities conducted within Testbed-18 were broken into loosely parallel investigative tracks. Each track clustered data sources, participant roles, and tasks together so that the components or problems needed to build the prototype API services – through implementations of OGC standards – could be simultaneously worked on, tested, and iterated. The investigative tracks are summarized in the table below and are described in more detail within subsequent sections of this ER.

Table 7

TRACK	PARTICIPANTS (PARTICIPANT ROLE)	DESCRIPTION
Initial Processing Track	<ul style="list-style-type: none"> • Steinbeis Consortium (Building Energy Processing Service D126) • interactive instruments (Building Energy Data Service D123) • Ecere (External Geospatial Data Service D127) 	The Initial Processing Track focused on the development and implementation of different OGC compliant Web API services that attributed energy consumption to buildings given the provided source geospatial building characteristics data. As part of the track, a minimal viable product for building energy simulation using a given geospatial dataset as input was implemented for Nuns' Island in Montreal, Canada. The energy simulation results were attributed to buildings in the Nuns' Island dataset and offered to clients as a data model through Web API implementations.
Extended Processing Track	<ul style="list-style-type: none"> • Steinbeis Consortium (Building Energy Processing Service D126) • interactive instruments (Building Energy Data Service D123) 	The Extended Processing Track focused on a more generalized geospatial processing workflow using data from Montreal, Canada. While track one aimed to stand up services using only the available datasets (without dataset conflation), track two aimed to generalize that workflow to the entire Montreal geographic area. The conflation of the provided geospatial building and building characteristics datasets into a common data model and its provision to clients using Web API implementations was a key activity within this track. The track also provided building archetypes for the Province of Quebec through a standards compliant Web API implementation. Unlike the Initial Processing Track, the Extended Processing Track did not attribute energy consumption to buildings in Montreal.
Data Visualization Track	(all Building Energy Client D124 and D125) <ul style="list-style-type: none"> • interactive instruments • Steinbeis Consortium • Climative 	The Data Visualization Track documented the workflows towards consuming the outputs of the Initial and Extended Processing Tracks and displaying them through novel and innovative visualization tools. These visualization tools were a demonstration of how the

TRACK	PARTICIPANTS (PARTICIPANT ROLE)	DESCRIPTION
	<ul style="list-style-type: none"> • Properate • GeoSolutions • Ethar Inc. 	utility and municipal planning use cases could be achieved when supported by underlying Web APIs.

5.4. Testbed data inputs and parameters

5.4.1. Testbed locations

The locations chosen by the Testbed participants and NRCan were the City of Montreal Canada, and the Nuns' Island enclave within it. These locations were chosen because of their available sets of publicly available geospatial data.

5.4.2. Testbed data inputs and descriptions

The Testbed-18 participants had access to the following geospatial and building (energy) data sources relevant to the chosen locations:

- Montreal agglomeration property assessment roll data (City of Montreal);
- City of Montreal 3D buildings 2016 LoD 2 model with textures (City of Montreal);
- Geoindex shared platform (Province of Quebec);
- City of Montreal digital terrain model (City of Montreal);
- Automatically Extracted Buildings (NRCan);
- Building, alteration and demolition permits (City of Montreal);
- EnerGuide Rating System (ERS) database entries (NRCan);
- Housing Technology Assessment Platform (HTAP) archetype files for HOT2000 software (NRCan CanmetENERGY); and
- Reduced HOT2000 File Representation (NRCan CanmetENERGY).

Descriptions of the above datasets with web links providing access to the data and more information on the datasets (taken from the database maintainers, where publicly available) is provided in the table below. Details on the specific use cases for each dataset as well as links to databases follow in the Test Data section of each investigative track.

Table 8

DATABASE NAME (OWNER)	AUGMENTED DESCRIPTION FROM DATABASE MAINTAINER
Montreal agglomeration property assessment roll data (City of Montreal)	<p>The City of Montreal's property assessment roll, collected every 3 years, consists of an inventory of all the properties in the city. The property roll is also collected for each of the 16 related municipalities within the Montreal agglomeration. The roll includes geospatial vector data describing the property boundaries within the City of Montreal and Montreal agglomeration. It also contains general information on the assessed units, including the codification of use (CUBF) – the code that defines the type of activities (text use label) that a building is recorded as used for (building function), the evaluation unit category as either regular or condominium (building category), approximate dimensions, and additional parameters as described and available on the City of Montreal open data portal on the page for the property assessment roll.</p>
City of Montreal 3D buildings 2016 City GML LoD 2 model with textures (City of Montreal)	<p>The City of Montreal provides the 3D buildings 2016 (3D buildings 2016 (LOD2 model with textures)) dataset through its open data portal web page. The dataset contains digital models of 3D LoD2 buildings with textures in CityGML, 3DM and GDB formats and in DWG format, without textures, for the Montreal boroughs of Côte-des-Neiges, Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun (Nuns' Island). Within the dataset, building roofs are modeled by photogrammetry and walls are generated by negative extrusion of the roofs towards the ground. Aerial photographs (from MWC 2016) are applied to projected walls and roofs. The model is structured according to the CityGML standard. The digital terrain model (DTM) of some boroughs within the City of Montreal is also available (detailed in its own table entry).</p>
Geoindex shared platform (Province of Quebec)	<p>Geoindex is a shared platform of geospatial data and aerial photographs for the Province of Quebec. Access to the platform is granted to members of the Quebec university community or through a guest request. The platform includes many geospatial datasets and is not limited to buildings. Selected datasets are available for export in different geospatial (or data) formats such as .dwg, .shp, .geojson, .csv, .xls. Building related information used by Testbed-18 participants was created by the Québec – Ministère des Ressources naturelles et de la Faune and is available for dates back to 2008.</p>
City of Montreal digital terrain model (City of Montreal)	<p>The City of Montreal's Digital Terrain Model (DTM) is a simplified representation of ground altimetry. The available data is in the form of a triangulated irregular network (TIN). These are areal digital geographic data constructed by triangulating a set of points. The vertices are connected to a series of segments to form a mesh of triangles of different sizes. This representation can be used as a basis for the 3D buildings of the digital base model. The building elevation data has a planimetry of ± 20 cm and altimetry of ± 20 cm.</p>
Automatically Extracted Buildings (NRCan-CCMEO)	<p>The Automatically Extracted Buildings vector format dataset was created in 2021 by NRCan using high-resolution elevation data. It consists of a single topographical feature class that delineates polygonal building footprints automatically extracted from airborne LiDAR data, high-resolution optical imagery, or other sources. The data provided included minimum and maximum heights of a building. The data is available in .shp format and was geo-referenced. The Automatically Extracted Buildings product is available as an OGC Web Map Service (WMS).</p>
Building, alteration, and demolition	<p>The building, alteration and demolition permits dataset is publicly available from the City of Montreal. It includes information for all building permits issued by different boroughs within the City of Montreal. The dataset is available in multiple formats (.csv, GeoJSON). From</p>

DATABASE NAME (OWNER)	AUGMENTED DESCRIPTION FROM DATABASE MAINTAINER
permits (City of Montreal)	this dataset, valuable information regarding construction, modifications, and demolitions of buildings in the region can be obtained.
EnerGuide for Houses database entries (NRCan-OEE)	<p>NRCan delivers energy labeling programs for residential housing in Canada. These programs include offerings for both new and existing homes. These programs include on-site evaluations, in which an energy advisor (EA) undertakes numerous measurements of a home's energy-efficiency characteristics. Advisors enter this data into NRCan's HOT2000 energy analysis software, which a) calculates the home's estimated energy use, and b) saves all measurements in an XML formatted file (known as the .h2k file). The results of the HOT2000 energy analysis are used by EAs to identify effective energy efficiency measures and provide recommendations to homeowners and home builders. One of the XML tags within the .h2k tag structure is called the TSV tag. The TSV tag contains a set of parameters (~400) that both describe the characteristics of a building and record its energy simulation results (derived from running the file in HOT2000). The EnerGuide for Houses (EGH) database is a central repository for tracking the TSV tag data collected during residential energy evaluations. Information collected in the database can be used by internal program personnel to generate statistics such as the number of house evaluations performed across the country or technical house characteristics data and assist in making policy decisions.</p>
Housing Technology Assessment Platform (HTAP) archetype files for HOT2000 software (NRCan CanmetENERGY)	<p>NRCan developed the Housing Technology Assessment Platform (HTAP) to support research and program development. HTAP is a collection of data and tools that automate and extend the HOT2000 residential energy simulation tool. HTAP has been used to:</p> <ul style="list-style-type: none"> • optimize the design of Net-Zero Energy and Net-Zero-Ready Housing, and deep energy retrofits; • investigate the impact potential of different heating technologies in the Canadian housing stock; and • estimate costs and benefits associated with changes in the building code. <p>The HTAP Archetype Database is a collection of ~6800 residential archetypes generated using HOT2000 and EGH records. The database follows previous work that characterized and developed a statistically representative database of low-rise housing across Canada. The HTAP Archetype Database files are provided in the .h2k (HOT2000) file format.</p>
Reduced HOT2000 File Representation (NRCan Canmet ENERGY)	<p>Through internal and external feedback, NRCan recognized that the data stored in the EGH database (TSV format) and the HTAP (.h2k format) files are relevant to various energy conservation and carbon reduction research activities, including stock modeling and mapping and program design. While the EGH and HTAP data is valuable, its current format is cumbersome for non-experts to work with. In the case of HOT2000 files, which contain important building characteristics that are not contained in the TSV EGH format, each house audit exists in a separate .h2k file, and .h2k files comprise various versions and schemas. Researchers and developers must develop their own computer code to parse data from various .h2k formats to perform statistics and analysis. There is no way to search and extract audit data without such measures.</p> <p>Understanding that this data storage scheme is a barrier to accessing and using data from NRCan's programs, NRCan undertook a project to develop a prototype "Reduced HOT2000 File Representation" alternate data storage format for information contained in .h2k files. Key requirements for the format were:</p> <ul style="list-style-type: none"> • accommodate data from multiple homes (i.e., multiple .h2k files); • data from different versions of .h2k files;

DATABASE NAME (OWNER)	AUGMENTED DESCRIPTION FROM DATABASE MAINTAINER
	<ul style="list-style-type: none"> • be machine-readable and suitable for scripting, machine learning and statistics applications; and • be interoperable with geospatial analysis applications. <p>The work to create this prototype format was completed before Testbed-18 and therefore the format and associated documentation were provided to Testbed participants for use and evaluation within the Testbed activities. The data format and work leading up to its design can be obtained by contacting CanmetENERGY in Ottawa.</p>



6

TESTBED EXPERIMENTS – INITIAL PROCESSING TRACK

TESTBED EXPERIMENTS – INITIAL PROCESSING TRACK

6.1. Description

The initial processing investigative track primarily focused on the development and implementation of different building energy web services as APIs. These API-accessible services consisted of building energy data services, external geospatial data services, and building energy processing services. Due to this track's nature, less attention was paid to input data quality. This was due to the small spatial extent of the input dataset used and the prototype-focused nature of the activity. A CityGML Level of Detail (LoD) 2 model of Nuns' Island (Verdun) in Montreal was taken as a sample input dataset for this track.

6.2. Test Locations

This track focused primarily on Nuns' Island in Montreal although some data from the datasets covering the entire City of Montreal were also used.

6.3. Test Data

The following datasets were used by the participants in the initial processing track.

- City of Montreal 3D buildings 2016 (detailed textured building geometries in CityGML LoD2 for the Côte-des-Neiges–Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun neighborhoods)
- Montreal agglomeration property assessment roll data (attributes on building function, year of construction, building category, neighborhood name)

6.4. Initial processing track summary

The following sections detail the participants' roles and interactions within this work track and are divided by the type of web service being provided by the participant to the track task.

Each web service section describes the activities that occurred during the testbed to complete the deployment of the service as well as any relevant comments, conclusions, and results. A description of the actions performed by the participants within this track follows in the table below.

Table 9

PARTICIPANT ROLE	PARTICIPANTS	DESCRIPTION
Building Energy Processing Service (D126)	Steinbeis Consortium	The Nuns' Island CityGML LoD 2 dataset from the City of Montreal was cleaned and enriched with additional building characteristic information from the Montreal agglomeration property assessment roll dataset. A Simstadt energy simulation engine API implementing the OGC API – Processes interface was developed and deployed.
Building Energy Data Service (D122 and D123)	interactive instruments	An OGC API – Features compliant API implementation instance was deployed that further enriched the Nuns' Island CityGML LoD 2 dataset with the energy simulation results from the Building Energy Processing Service for Nuns' Island buildings. The applicability of CityGML Energy ADE and other data formats such as Mapbox Vector Tiles, FlatGeobuf, and glTF for the storing of the produced energy data was explored.
External Geospatial Data Service (D127 and D128)	Ecere	3D Geometry was made available through an implementation instance of the draft OGC API – Geo Volumes specification. The geometry was served as a 3D Tiles bounding volume hierarchy and implicit tiles. The attributes from the CityGML dataset were populated into a GNOSIS data store and made available through an implementation instance of the OGC API – Features specification.

6.5. Building Energy Processing Service (Steinbeis Consortium)

An overall workflow diagram of various tasks and contributions made by partners within the Steinbeis Consortium within the Initial Processing Track is shown in the figure below.

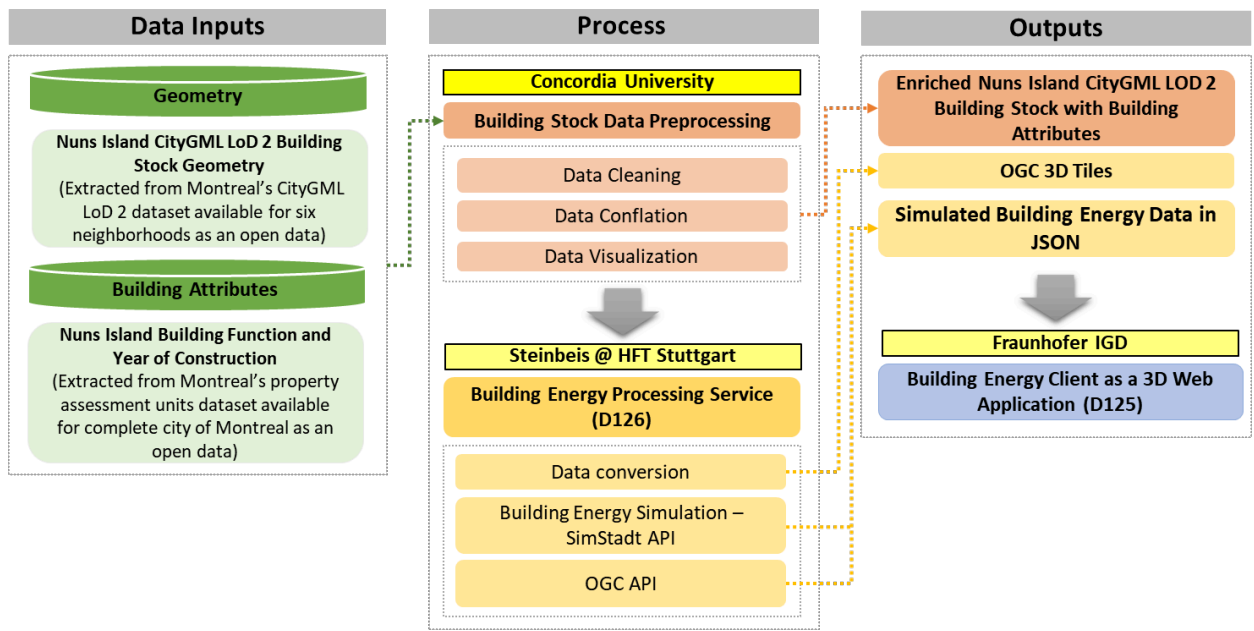


Figure 9 – Overall Workflow Diagram of Steinbeis Consortium Contribution.

6.5.1. Test data conflation

The original Nuns' Island CityGML LoD 2 geometry dataset was extracted from the City of Montreal 3D buildings 2016 dataset (detailed textured building geometries in CityGML LoD2 for the Côte-des-Neiges-Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun neighborhoods) held within the City of Montreal's open data portal. The dataset was further processed and enriched by Concordia University (Steinbeis Consortium). The textures from the original Nuns' Island CityGML data were removed to reduce file size. Moreover, the original CityGML data contained building GML IDs starting with numbers. This violates the constraints of XML/GML. Subsequently, new building GML IDs were produced for the entire Nuns' Island CityGML dataset. A copy of the original building GML IDs was kept for referring back to the original dataset whenever required. Additionally, in the original dataset the Coordinate Reference System (CRS) was also found to be missing. Thus, the processed Nuns' Island dataset was correctly referenced to its native CRS of the dataset: NAD83 (CSRS)/MTM zone 8 + CGVD28 height (a compound CRS of EPSG:2950 and EPSG:5713). The processed Nuns' Island CityGML dataset was then further enriched with attributes such as year of construction, building function, building category, and neighborhood name integrated from the Montreal agglomeration's property assessment roll dataset. The Nuns' Island (Verdun) CityGML building geometries and attributes for building function and year of construction were mandatory requirements for the Building Energy Client D125 from the Steinbeis Consortium.

In addition to its use as a data input for the development and implementation of different building energy web services within the Testbed, the enriched CityGML Nuns' Island dataset also formed the basis for the data conflation of the Montreal wide CityGML dataset produced as a part of the Extended Processing Track.

A basic web-based visualization of enriched Nuns' Island building stock dataset in OGC 3D Tiles standard using CesiumJS is shown in the figure below.



Figure 10 – Web 3D Visualization of Enriched Nuns' Island CityGML Dataset.

6.5.2. Energy simulation and building energy processing service

The building energy processing service provided by the Steinbeis Consortium was built based on SimStadt – an urban simulation environment developed initially as desktop software at HFT Stuttgart. This software was used to create workflows to simulate the dynamic energy-related attributes in the 3D city models provided to the service in CityGML format. Example generated attributes include annual and monthly specific space heating and domestic hot water demands. The building energy simulation workflow in SimStadt is supported by customizable modules (also called processors) of building preprocessing, building physics, building usage, weather, solar irradiance, and monthly energy balance. The building preprocessing module imports the CityGML building dataset, its validation, and creation of a SimStadt specific building model.

The building physics module is linked with a building physics library running in the backend to assign the building physics specific attributes such as building type, U-value, infiltration rate, and window ratio to every building surface based on the building's year of construction. The building usage module is also linked with the building usage library in the backend which defines usage zones according to different building functions and also defines the occupancy density, internal gains, set point temperatures, domestic hot water consumption, and ventilation according to usage types. Thus, as a prerequisite to SimStadt and subsequently D126, CityGML attributes of building function and year of construction are mandatory to calculate the building physics and building usage attributes from SimStadt.

If the required attributes are not available, the following default values are applied: (building function = residential building) and (year of construction = 1980). Both the building physics library and building usage libraries are developed based on the local building archetypes of the case study area under consideration. Originally, the building physics and building usage libraries of SimStadt were developed using German building archetypes. However, due to SimStadt's flexible and modular approach, building physics and usage libraries for any location can be easily

defined based on local building archetypes. For any new location, the building functions from CityGML dataset can be internally mapped to the existing building physics and usage libraries within Simstadt or new building physics and usage libraries could be developed.

In the last three modules, the weather processor retrieves the weather data based on location (coordinates and CRS) of the CityGML building dataset, the solar irradiance processor calculates monthly average solar irradiances on every surface – with or without shadows, and the monthly energy balance processor calculates a monthly energy balance to each building, in order to determine monthly heating (including domestic hot water demand) and cooling energy demands.

During Testbed-18, a basic implementation of the SimStadt simulation software was developed using Nuns’ Island CityGML which was called SimStadt-API. To fast track the development process, Canadian building archetypes were not implemented and the building functions from Nuns’ Island CityGML dataset were internally mapped with the SimStadt’s German archetype library to assign the building physics and usage values for running the building energy simulation. The SimStadt API allowed users to store the CityGML file in the SimStadt repository disk and then simulate the 3D building energy data on-the-fly. The service communicated over a RESTful protocol using JavaScript Object Notation (JSON) encodings.

Once the SimStadt API was tested within the Testbed to verify its functionality when called upon by the client, the OGC API – Processes interface, based on the OGC API – Processes – Part 1: Core specification, was implemented on top of the SimStadt API. The OGC API – Processes enabled the execution of computing processes and the retrieval of metadata describing their purpose and functionality. The SimStadt API with OGC API – Processes service endpoints were published online and are shown in the table below.

Table 10

#	URL	METHOD	DESCRIPTION
1	https://steinbeis-3dps.eu/ogc-api-processes	GET	Landing Page
2	https://steinbeis-3dps.eu/ogc-api-processes/processes	GET	Get a list of all available processes (currently: 1, “simstadt_api”)
3	https://steinbeis-3dps.eu/ogc-api-processes/processes/simstadt_api	GET	Get a description of one process by id “simstadt_api” e.g. input parameter.
4	https://steinbeis-3dps.eu/ogc-api-processes/processes/simstadt_api/execution	POST	Execute “simstadt_api” process (create a job) with an input bbox parameter
5	https://steinbeis-3dps.eu/ogc-api-processes/jobs	GET	Get a list of all jobs
6	https://steinbeis-3dps.eu/ogc-api-processes/jobs/{id}	GET	Get a job status by JobID
7	https://steinbeis-3dps.eu/ogc-api-processes/jobs/{id}/result	GET	Get a result of the job by JobID
8	https://steinbeis-3dps.eu/ogc-api-processes/jobs/{id}	DELETE	Delete a job by JobID

With this OGC API Processes interface, users or clients can request interaction with the SimStadt API computational tasks within the prepared test dataset of Nuns' Island. To interact with this API, users may input the bounding box (bbox) and interested attributes. The possible attributes that can be requested from the OGC API Processes based on SimStadt tool are:

Table 11

location, cityName, occupants, buildingFunction, yearOfConstruction, yearOfRefurbishment, exclusionReason, center, height, calculatedHeight, volume, surfaceToVolumeRatio, totalSurfaceArea, azimuth, surfaceBaseLength, sunWindExposedArea, roofType, averageStoreyHeight, atticHeatingType, storeysAboveGround, basementCeiling HeightAboveGround, basementHeatingType, heatedVolume, heatedArea, physicsLibraryName, physicsLibrary Source, effectiveThermalCapacity, infiltrationRate, indirectlyHeatedAreaRatio, constructionTypeIId, shortWave Reflectance, uValue, windowRatio, windowTypeIId, windowFrameRatio, usageLibraryName, usageLibrarySource, usageLabel, usageZoneArea, healthyAirChangeRate, averageInternGains, coolingSetpointTemperature, heating SetpointTemperature, heatingSetbackTemperature, usageHoursPerDay, usageDaysPerYear, occupantNumber, dhwDemand, usageType, usageTypeDescription, ambientTemperature, snow, windSpeed, windDirection, short WaveIrradiance, irradianceAzimuthResolution, irradianceDescription, irradianceCalculatedTime, irradianceMaxTilt, monthlyHeating, monthlyCooling, pvPotential, loadDurationCurve, loadProfile, shadowPercentage, timedShadow Percentage.

The request to execute SimStadt process is formatted according to the OGC API Processes official schema <https://schemas.opengis.net/ogcapi/processes/part1/1.0/openapi/schemas/execute.yaml>. The example below shows the SimStadt-OGC API request with a specified bbox and interested attributes:

- POST Request HTTP URL https://steinbeis-3dps.eu/ogc-api-processes/processes/simstadt_api/execution
- POST Request Body

Example 1

```
{
  "inputs": {
    "bbox": {
      "bbox": [
        -73.5517930984,
        45.4548326034,
        -73.5433816910,
        45.4600102336
      ]
    },
    "interestedAttributes": [
      "yearOfConstruction",
      "buildingFunction",
      "monthlyHeating",
      "specificSpaceHeatDemand",
      "spaceHeating"
    ]
  }
}
```

- Example Result from SimStadt Process

Example 2

```
{
  "attributes": {},
  "buildings": [
    {
      "id": "uuid_38081050-b528-4bf8-a724-e7283dceadff",
      "parts": [
        {
          "id": "uuid_38081050-b528-4bf8-a724-e7283dceadff",
          "attributes": {
            "buildingFunction": "1010",
            "spaceHeating": 35584,
            "yearOfConstruction": 1980,
            "specificSpaceHeatDemand": 127.51620831302965,
            "monthlyHeating": [
              7912.167770794624,
              6568.995639505456,
              5007.5388255877115,
              2364.773931418515,
              351.13249275333396,
              0.32753967937503603,
              0,
              0.000004998638218012275,
              144.71517862135374,
              1957.066680475804,
              4192.914879173882,
              7084.667482389498
            ]
          },
          "surfaces": null,
          "usageZones": [
            {
              "attributes": {}
            }
          ]
        }
      ]
    }
  ],
  "...": {}
}
```

- The result is also cached and listed to the server under the job entity in the Processes API. The list of all jobs can be accessed at <https://steinbeis-3dps.eu/ogc-api-processes/jobs>
- The job result can be accessed via <https://steinbeis-3dps.eu/ogc-api-processes/jobs/{job-name}/result>
- For example, the process result above can be accessed via https://steinbeis-3dps.eu/ogc-api-processes/jobs/simstadt_api_2023-01-15T11:28:27.604Z/result. It can be used to conflate or add each building to any client on the fly.

A building energy client as a 3D web-based client that supports the implementation of D126 is explained in D125 under the data visualization track. A high-level workflow of the above processes is shown in figure below:

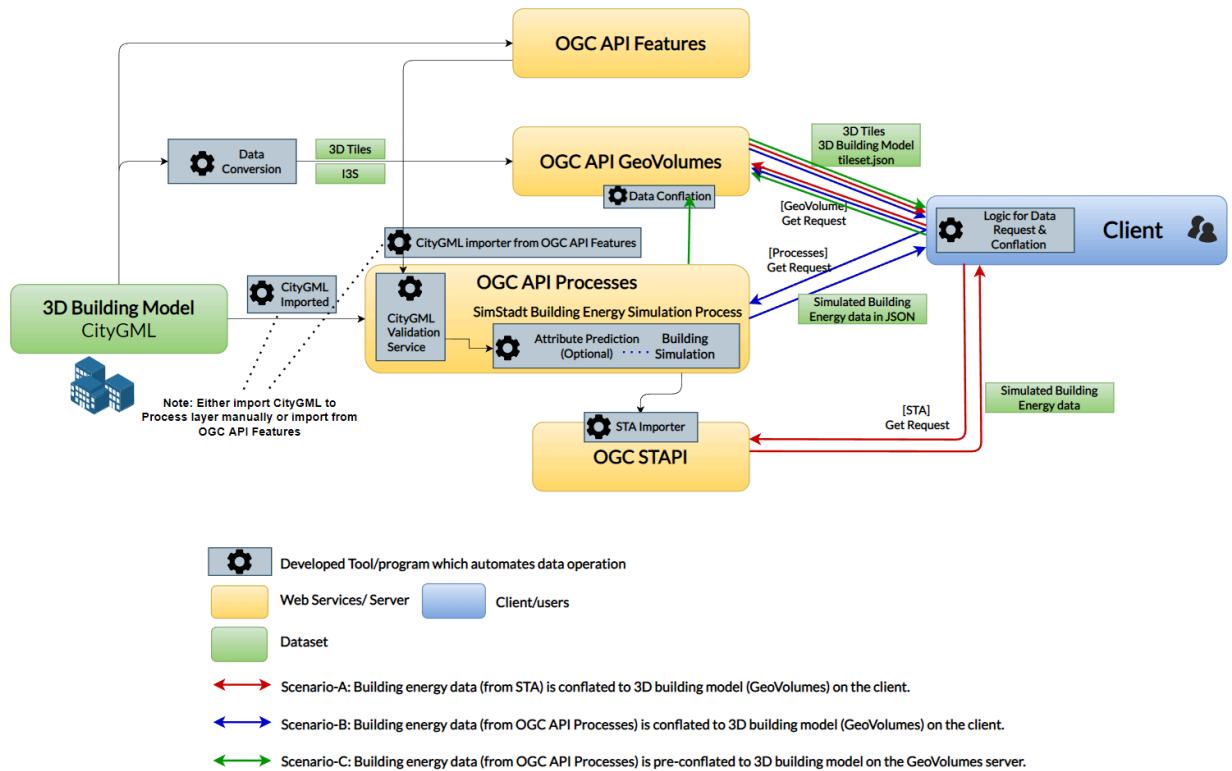


Figure 11 – A high-level workflow of building energy processing service developed by the Steinbeis Consortium.

6.5.3. Challenges Faced

In the development process of D126, challenges occurred due to the unsuitability of Canadian specific building archetypes libraries in their current form, which were missing CRS as well as lack of building function and year of construction building attributes in the original CityGML, which is available as open data from the city of Montreal. In addition, invalid GML IDs and geometrical issues within the original CityGML dataset also slowed the development process. The majority of these issues were identified and solved while preparing the Nuns' Island CityGML dataset, however issues with building geometries and the fact that the building functions and year of construction were actually conflated from Montreal's property assessment units which provides general information on the use of property, year of construction, category, approximate dimensions, and registration number on a land parcel level still exists. As open data, the City of Montreal's property assessment unit dataset is the closest dataset available as open data to identify a building's function and its year of construction. These issues are discussed in detail under the Testbed challenges and Lessons Learned section.

6.6. Building Energy Data Service (interactive instruments)

For the initial processing track, the Building Energy Data Services deployed an OGC API – Features-compliant API implementation instance that provided energy information for buildings on Nuns’ Island in the City of Montreal.

The OGC API – Features implementation instance described within this section provided buildings from the Nuns’ Island CityGML LoD 2 dataset with the same API provided resources (3D Tiles, Mapbox Vector Tiles, features in multiple data formats) as the API for the Montreal CityGML dataset from the Extended processing track.

The CityGML data was enriched with the energy simulation results from the Building Energy Processing Service for the Nuns’ Island buildings. These were provided by the Building Energy Processing Service as JSON objects for each building. The results included the ID of the building in the Nuns’ Island CityGML dataset. This information was used to join energy simulation results with the buildings in the CityGML dataset (see Example 1) and the following workflow diagram.

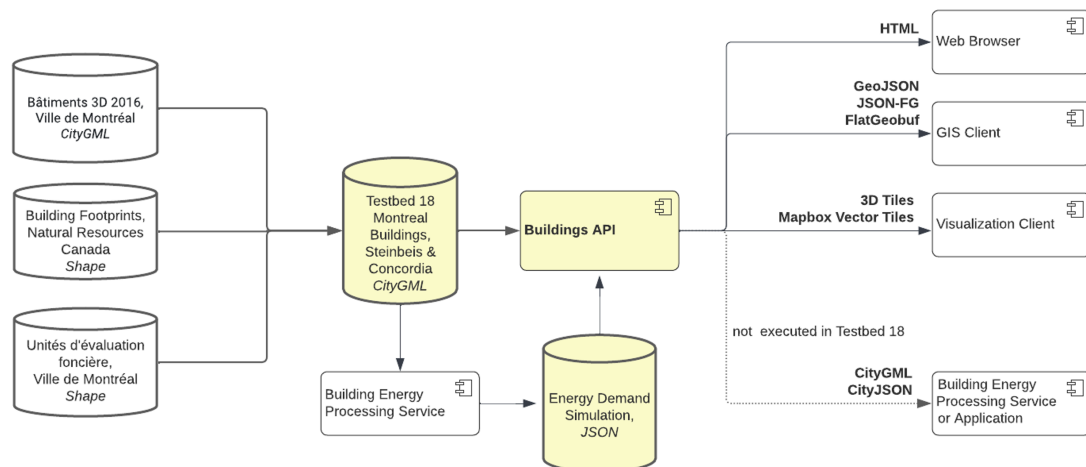


Figure 12 – The initial processing track workflow relevant to interactive instruments.

Example 1 – Simulation results for a single building

```
{
  "id": "uuid_3e0f963f-d185-4b73-8996-e7204ff80e25",
  "heatedVolume": 1974.3817841571754,
  "spaceHeating": 78401,
  "specificSpaceHeatDemand": 124.09105825730002,
  "monthlyHeating": [
    17492.251089563418645411730, 14744.943033879686481668614,
    11147.866684966062166495249,
    5044.366256928023176442366, 620.114847088773331051925,
    0.347244921370020165341,
    0.00000000000000000000, 0.000005360097304674127,
    318.355546060957749432418,
```

```

        4360.371482101460060221143, 9147.588205604211907484569,
15525.435466602464657626115
    ]
}

```

The energy data was loaded into the same database as the buildings data and the API configuration was updated to join the two datasets. The schema of the building energy data follows the CityGML Energy ADE format. Shown below (Example 2) are the attributes of the same building in the CityJSON representation, including the energy information.

Example 2 – Attributes of the building in the CityJSON representation

```

{
  "creationDate": "2022-06-29",
  "function": 1000,
  "yearOfConstruction": 2002,
  "measuredHeight": 20.024,
  "volume": {"type": "EnergyReferenceVolume", "value": 1974.3817841571754},
  "energyDemands": [
    {
      "endUse": "spaceHeating",
      "energyAmount": {
        "variableProperties": {
          "thematicDescription": "Space Heat Demand",
          "acquisitionMethod": "Simulation",
          "interpolationType": "SucceedingTotal",
          "source": "SimStadt"
        },
        "temporalExtent": {"begin": "2022-01-01", "end": "2022-12-31"},
        "timeInterval": {"value": 1.0, "unit": "year"},
        "values": [78401.0],
        "unit": "kWh"
      }
    },
    {
      "endUse": "spaceHeating",
      "energyAmount": {
        "variableProperties": {
          "thematicDescription": "Specific Space Heat Demand",
          "acquisitionMethod": "Simulation",
          "interpolationType": "SucceedingTotal",
          "source": "SimStadt"
        },
        "temporalExtent": {"begin": "2022-01-01", "end": "2022-12-31"},
        "timeInterval": {"value": 1.0, "unit": "year"},
        "values": [124.09105825730002],
        "unit": "kWh/m2"
      }
    },
    {
      "endUse": "spaceHeating",
      "energyAmount": {
        "variableProperties": {
          "thematicDescription": "Space Heat Demand",
          "acquisitionMethod": "Simulation",
          "interpolationType": "SucceedingTotal",
          "source": "SimStadt"
        },
        "temporalExtent": {"begin": "2022-01-01", "end": "2022-12-31"},
        "timeInterval": {"value": 1.0, "unit": "month"},
        "values": [
          17492.25108956342
        ]
      }
    }
  ]
}

```

```

    14744.943033879686    ,
    11147.866684966062    ,
    5044.366256928023    ,
    620.1148470887733    ,
    0.34724492137002017  ,
    0.0                    ,
    0.000005360097304674127,
    318.35554606095775    ,
    4360.37148210146     ,
    9147.588205604212     ,
    15525.435466602465    ,
  ],
  "unit": "kWh"
}
]
}
}

```

In comparison, the flexible storage capabilities of CityGML Energy ADE, other data formats such as Mapbox Vector Tiles, FlatGeobuf, and glTF are designed to only support attributes with simple values. In the 3D Tiles / glTF representation of the combined geospatial and energy datasets, the monthly data was therefore not included, as shown in the following screenshot.

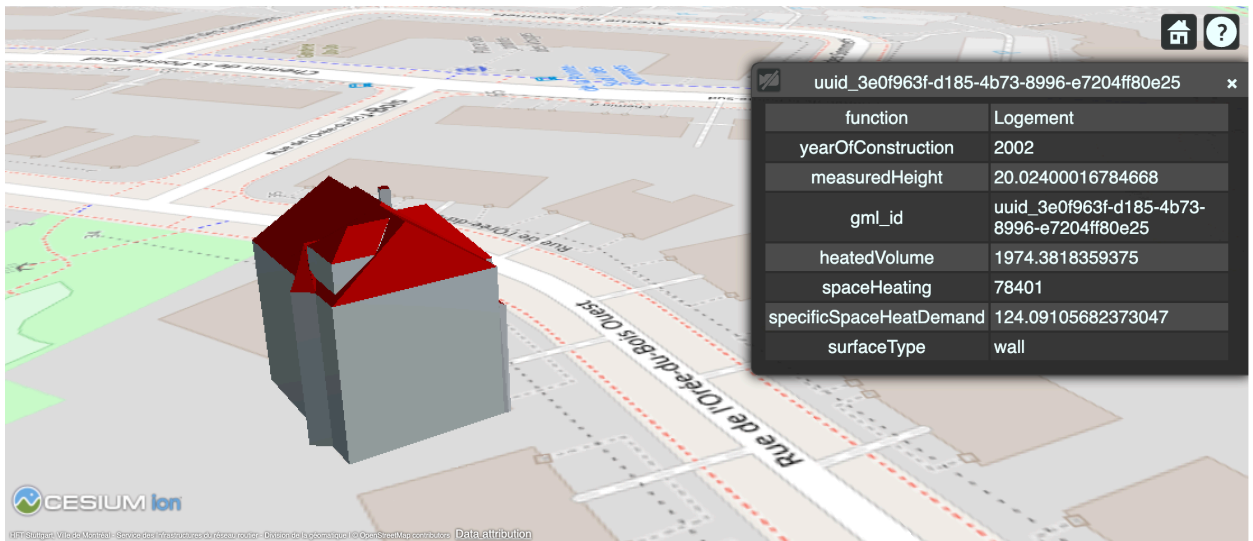


Figure 13 – The same building in Cesium JS using 3D Tiles

In the Mapbox Vector Tiles representation of the combined dataset, the monthly values were supported and included as shown in the screenshot from the data visualization track.

6.7. Geospatial and Building Energy Data Service (Ecere)

Ecere deployed a Web API using its GNOSIS Map Server product, implementing several OGC API Standards and draft Candidate Standards, to provide access to the Montreal 3D buildings dataset. The data provided included both the geometry and attributes from the full CityGML dataset, available from the following end-point

<https://maps.gnosis.earth/ogcapi/collections/montreal>

as well as a subset for *Nuns' Island*, available here:

<https://maps.gnosis.earth/ogcapi/collections/NunsIsland> .

The API also allowed dynamically augmenting the *Nuns' Island* subset with building energy attributes calculated on-demand using the SimStadt API, as demonstrated below.

<https://maps.gnosis.earth/ogcapi/processes/AddAttributes/execution?response=collection> .

6.7.1. Importing CityGML datasets

The first step in providing access to the Montreal 3D buildings dataset was to import the CityGML datasets prepared by Hochschule für Technik Stuttgart and the Steinbeis Consortium into the GNOSIS Map Server's native GNOSIS Data Store.

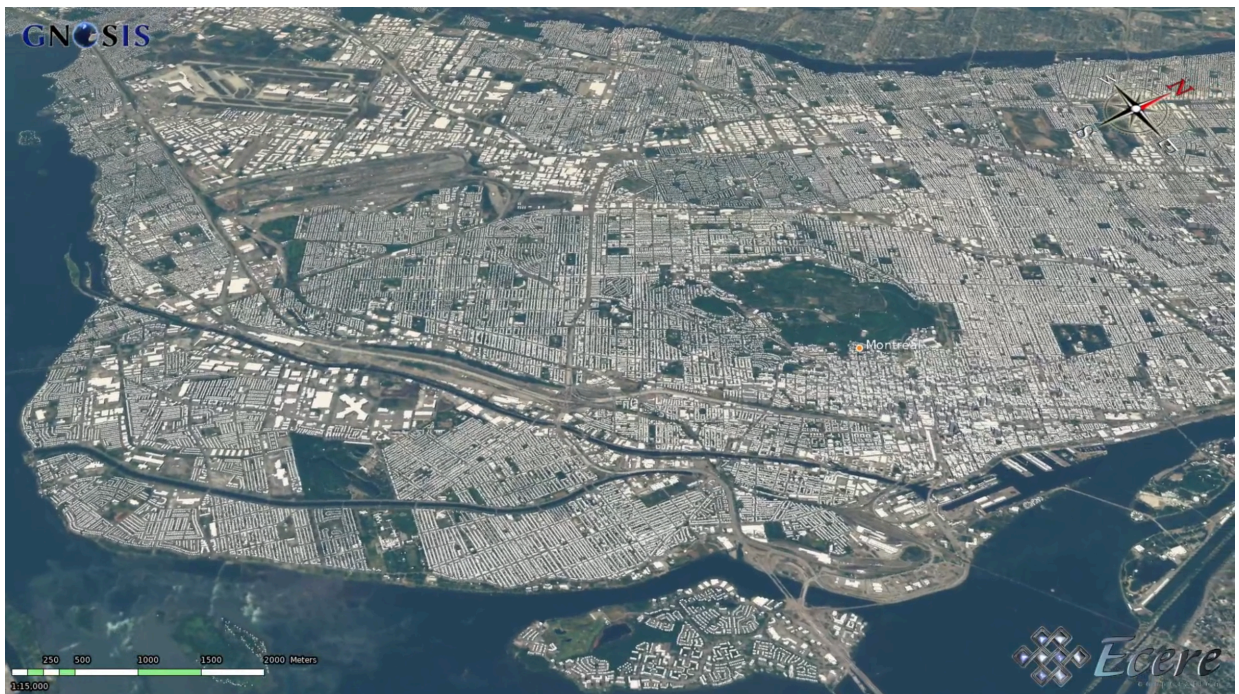


Figure 14 – Montreal 3D Buildings dataset imported from CityGML visualized in Ecere's GNOSIS Cartographer



Figure 15 – Nuns' Island subset Montreal 3D Buildings dataset imported from CityGML visualized in Ecere's GNOSIS Cartographer

The GNOSIS Data Store organizes geospatial information using a Discrete Global Grid System (DGGs) based on the *GNOSIS Global Grid 2D Tile Matrix Set*. The CityGML 3D building features are stored as tiled point features in *GNOSIS Map Tiles* format, each point feature referencing one 3D model. The 3D model geometry is stored using Ecere's compact *E3D* format, while the attributes are stored in a SQLite database. The attributes for the buildings imported from the Montreal CityGML dataset included the following.

- CATEGORIE (category)
- ID_UEV
- NOM (name of the neighborhood)
- function
- functionCode (based on an XML codespace)
- gmID (the ID from the source CityGML dataset)
- measuredHeight
- ori_bldgID (original building ID)
- yearOfConstruction

6.7.2. Access to 3D Buildings

The Montreal 3D buildings dataset could be accessed using the following OGC API Standards and candidate Standards from the deployed Web API.

- [OGC API – 3D GeoVolumes](#),
- [OGC API – Tiles](#), and
- [OGC API – Features](#).

The 3D GeoVolumes implementation in Ecere’s GNOSIS Map Server supports access to 3D geometry as a *3D Tiles 1.0* Bounding Volume Hierarchy tileset referencing batched 3D models (.b3dm).

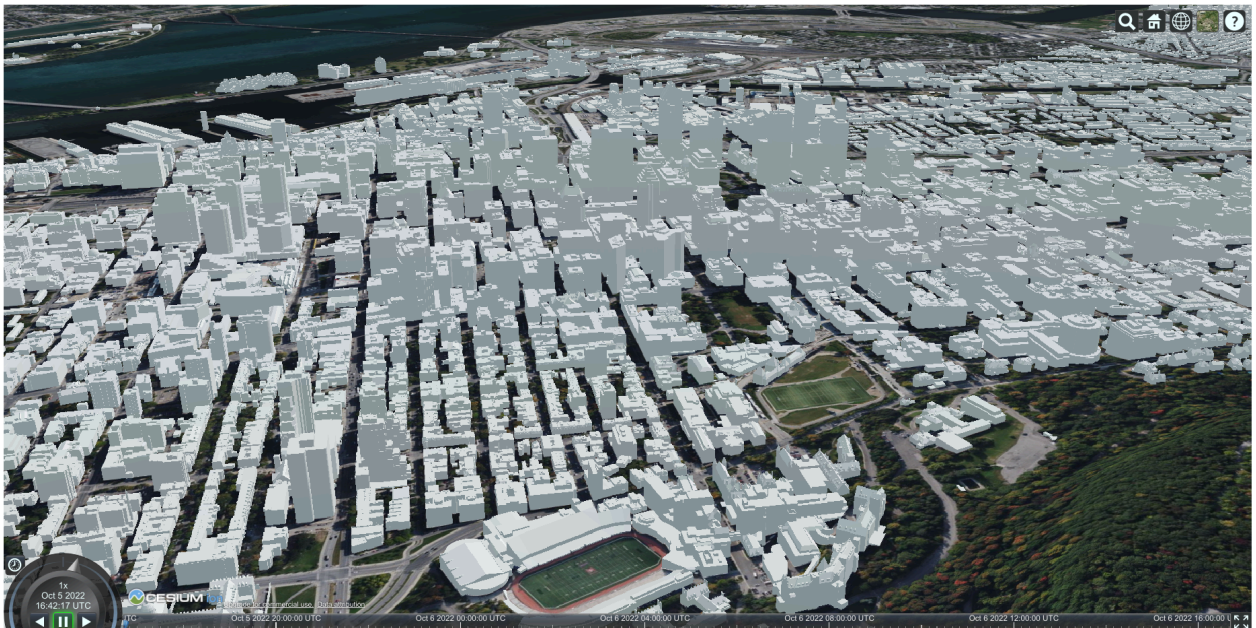


Figure 16 – Cesium JS client visualizing Montreal 3D Tiles generated from Ecere’s GNOSIS Map Server (downtown)

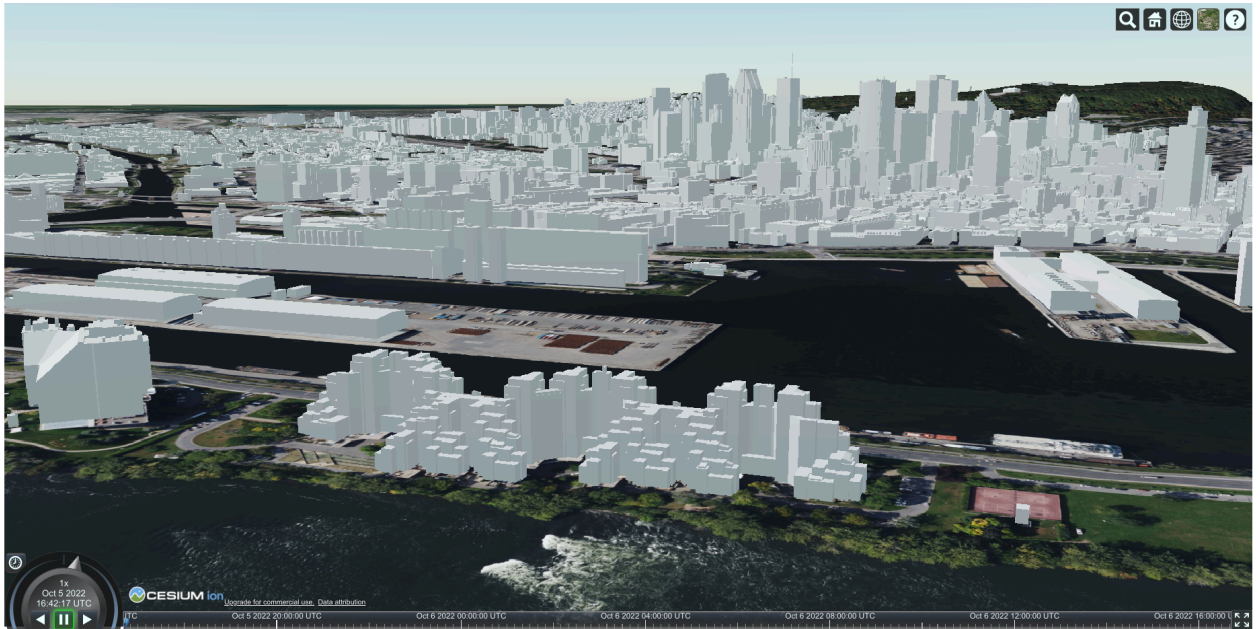


Figure 17 – Cesium JS client visualizing Montreal 3D Tiles generated from Ecere’s GNOSIS Map Server (Habitat 67)

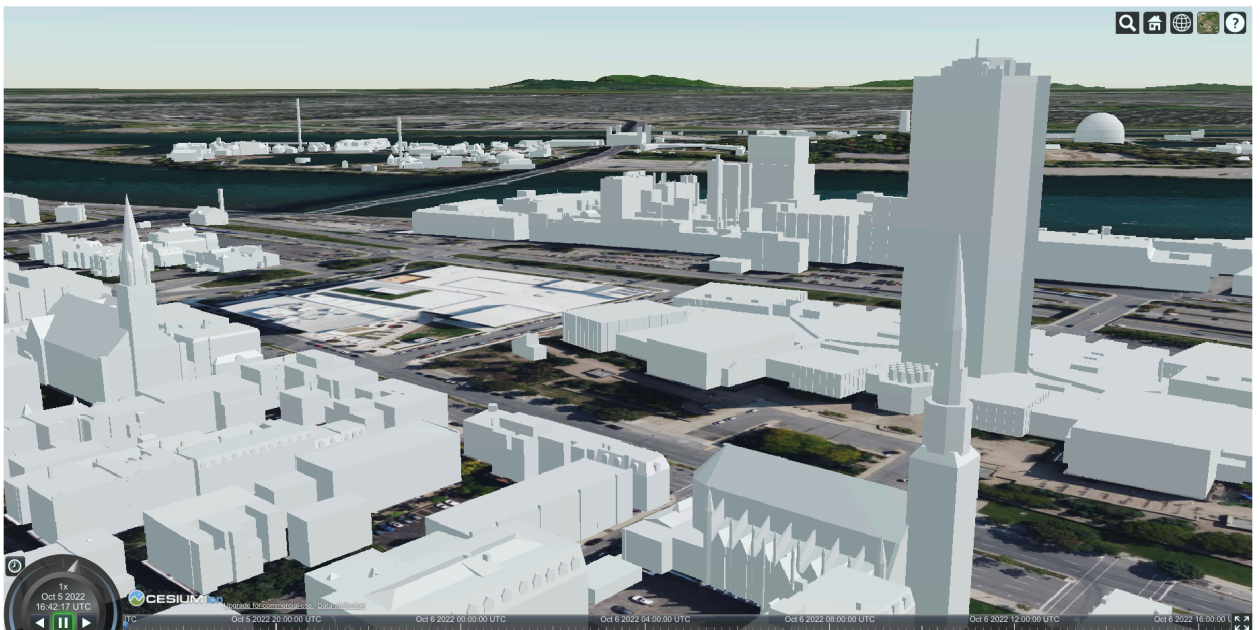


Figure 18 – Cesium JS client visualizing Montreal 3D Tiles generated from Ecere’s GNOSIS Map Server (close up view)

The implementation also supports two additional access modes to multi-resolution tiles of data based on the approach described in the appendix J of the [OGC Two Dimensional Tile Matrix Set and Tile Set Metadata](#) and [OGC API – Tiles](#).

In the first of these approaches, the batched 3D models can be accessed directly using a tile matrix (zoom level), tile row, and column in either E3D, gTIF or batched 3D models (.b3dm).

Within the tile, the axes are oriented East/North/Up based on the local tangent plane at the center of the tile, with the 0 of the Up axis on the surface of the WGS84 ellipsoid.

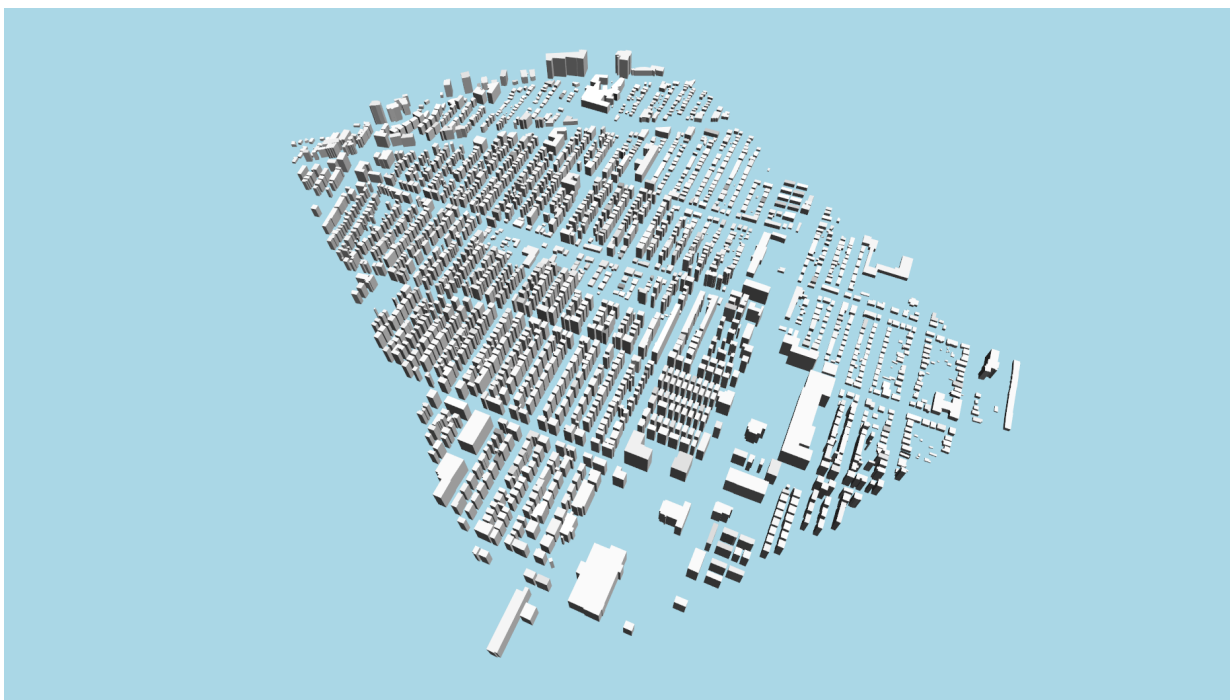


Figure 19 – An example tile batching multiple buildings in a single 3D model tile

In the second of these approaches, instead of containing geometry, the tiles contain points vector tiles, where each point references an individual model by ID and specifies a position, as well as an optional orientation and scaling, for that model.

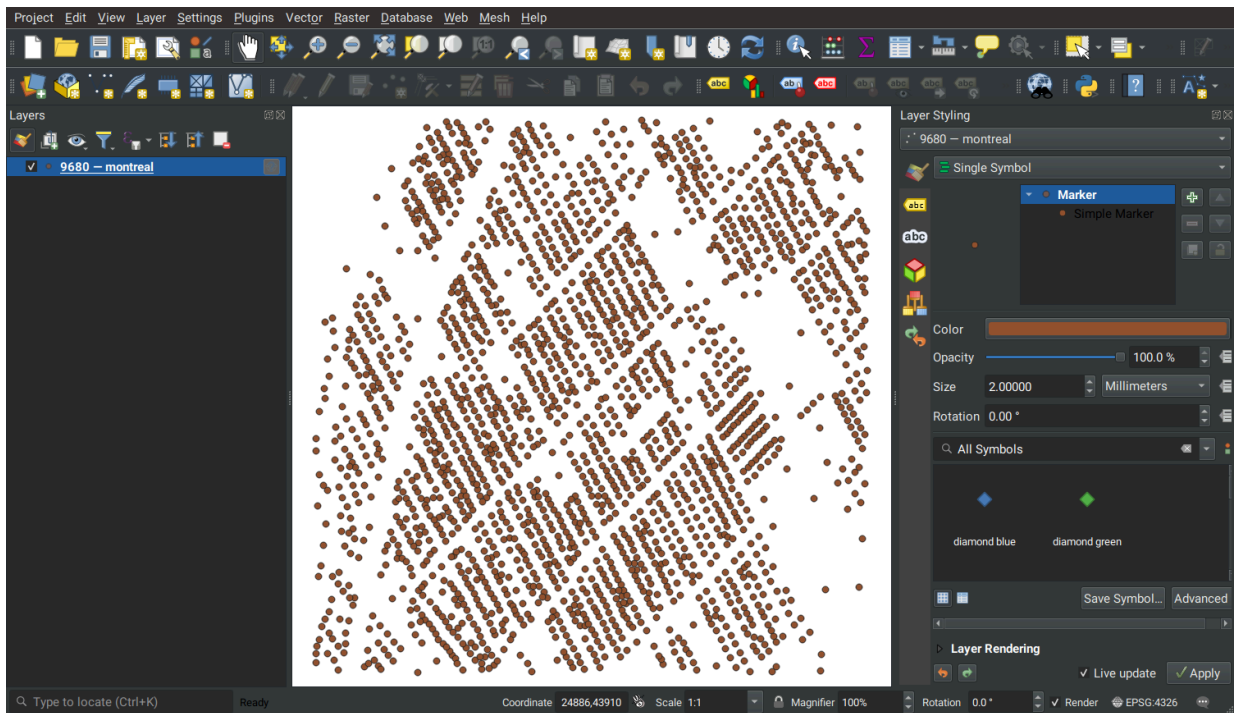


Figure 20 – Vector tile containing reference points, each referencing an individual model

Those individual models can be retrieved from a `/models/{modelId}` resource path as E3D or glTF.

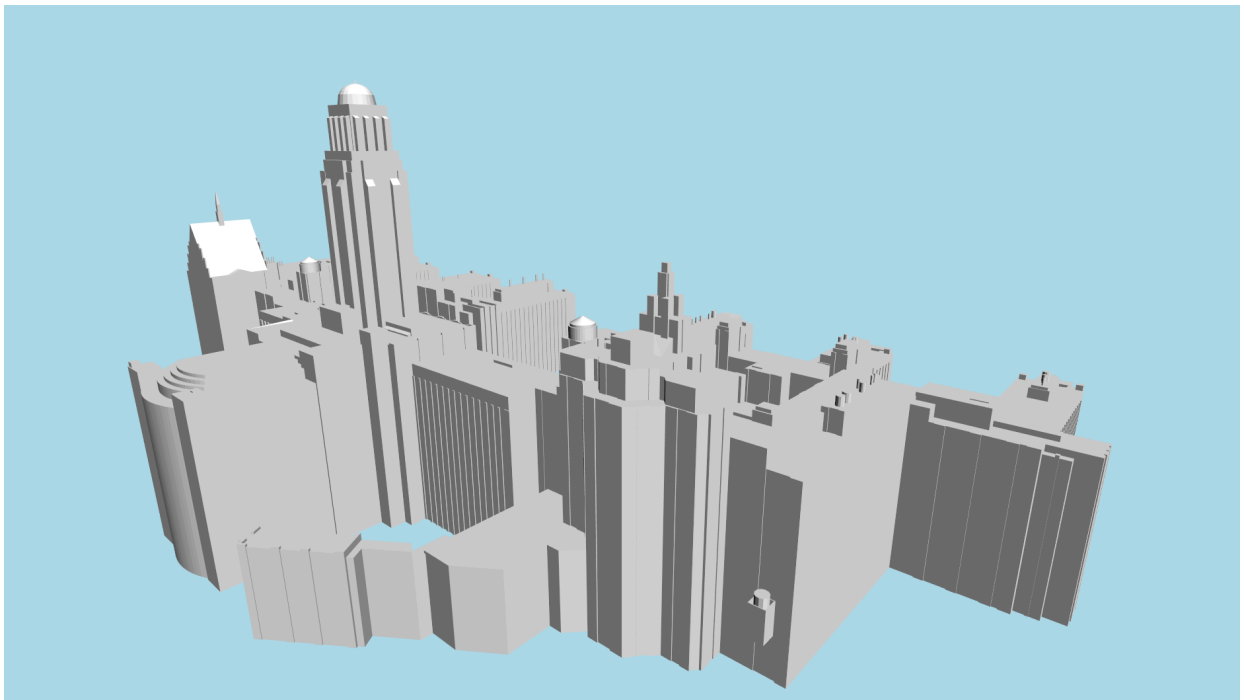


Figure 21 – Single building accessed from Montreal 3D Buildings collection

In order to allow visualization of the attributes associated with the 3D geometry, the UUID referencing individual buildings is stored as the gITF node identifier, whether retrieving batched 3D models or individual buildings.

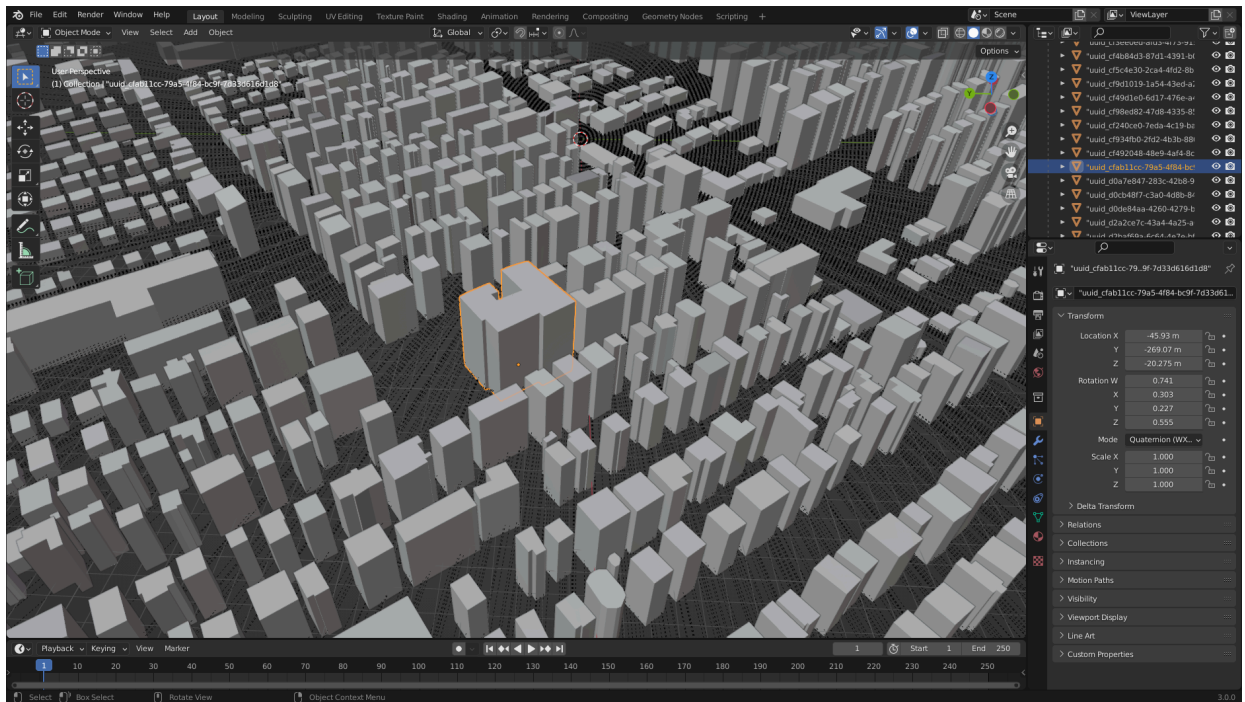
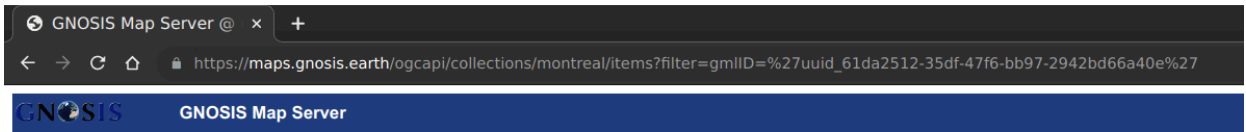


Figure 22 – Node IDs within gITF model allowing to link to OGC API - Features item to retrieve energy attributes (shown in Blender)

A client visualizing a particular tile could then make an OGC API – *Features* query for the relevant UUIDs with a CQL2 *filter* parameter referencing that UUID. Directly including the attributes in the gITF would have been significantly more difficult, since limited gITF extensions support currently exists in the Open Asset Import library used internally by the *GNOSIS Map Server* to generate the binary gITF 2.0 3D model files.



Feature 6497

from montreal

[Back to data collection](#)

(Download [GeoJSON](#) representation; view on [geojson.io](#))

Geospatial extent: { { lat: 45.5020385, lon: -73.8084064 } - { lat: 45.5020385, lon: -73.8084064 } }

Property	Value
CATEGORIE	Régulier
ID_UEV	04076714
NOM	Pierrefonds-Roxboro
function	Logement
functionCode	1000
gmlID	uuid_61da2512-35df-47f6-bb97-2942bd66a40e
maxAlt	33.1664999999956
maxLat	45.5021284000966
maxLon	-73.8082896993694
measuredHeight	5.34
minAlt	27.6885999999994
minLat	45.5019486993922
minLon	-73.8085231991489
ori_bldgID	61da2512-35df-47f6-bb97-2942bd66a40e
yearOfConstruction	1957

Figure 23 – Single feature accessed from Montreal 3D Buildings collection (OGC API - Features)

6.7.3. Integrating building energy attributes from SimStadt API

Ecere also experimented with augmenting a subset of the Montreal 3D buildings dataset for *Nuns' Island* with building energy attributes calculated on-the-fly by invoking the SimStadt API provided by the Steinbeiss Consortium.

Participants were able to successfully demonstrate the use of the *OGC API – Part 3: Workflows and Chaining* draft candidate Standard, including the *Collection Output* requirements class, that can be used to access a resulting virtual collection using *OGC API – Features*, including the additional energy attributes computed on-demand based on the area of interest (bbox parameter).

GNOSIS Map Server

GNOSIS Map Server

ecore.ca

Vector features

for NunsIsland

(Download GeoJSON representation)

Back to data collection

Feature ID	Geometry	Min Lat	Min Lon	Max Lat	Max Lon	function	functionCode	gnID	maxAlt	maxLat	maxLon	measurHeight	minAlt	minLat	minLon	yearOfConstruction	buildingFunction	heatedVolume	monthlyHeating	spaceHeating	specificSpaceHeatDemand
1	point	45.465978	-73.5399734	45.465978	-73.5399734	Legement	1000	mid_522017a1-548446018a2e15789649977	13.0232633091188	45.4660359094905	-73.539969748282	14.551	0	45.4659202081478	-73.540071789217	1985	1010	822.638523150881	[8274.7557261389562, 6896.9375447929779, 5081.0781855126597, 2396.018767243689, 237.0657129397721, 0.20685595662, 0, 0.000065487974, 331.472140258437, 2034.4258871895987, 4359.0078539655369, 7400.727624689423]	36502	139.003949700735
2	point	45.4617934	-73.5491034	45.4617934	-73.5491034	Legement	1000	mid_e1152878-050c4165-9502c869326c1f4f	13.1201599046048	45.4618886176602	-73.5489615417561	15.104	0	45.4617182616714	-73.5492453769724	1992	1010	2733.911536947841	[20548.11267141884859, 16979.7313227182531, 11280.586694143385, 4197.080282123697, 258.1578782838348, 0.891580399906, 0, 0.000065487974, 72.6572127646286, 3524.6243946867818, 10595.243661274515, 18739.4663293865229]	85027	97.1901875424375
3	point	45.4625221	-73.553908	45.4625221	-73.553908	Legement	1000	mid_and3861-b48c4c71-6906e31f8a49896c	15.8171214766213	45.4625848134996	-73.553861533894	16.995	0	45.462494544176	-73.55394603097	1987	1010	1002.554378721645	[array]	47728	148.769614991101
4	point	45.4595012	-73.5502012	45.4595012	-73.5502012	Legement	1000	mid_34585c71-0775-4817-845f-b61c6799f38	19.0848117399819	45.4596828812314	-73.5499832094752	24.456	0.6126181187307	45.4593141758812	-73.5504173819008	2005	1010		[array]		
5	point	45.4653246	-73.5492957	45.4653246	-73.5492957	Legement	1000	mid_0868c9c0-06a649b873-aad39120849f	15.9960196755753	45.4653282826205	-73.5491369036472	15.999	0	45.4652665469313	-73.5493846517724	1999	1010		[array]		
6	point	45.4653881	-73.5480397	45.4653881	-73.5480397	Legement	1000	mid_c57006b2-78e741a98a2c11714958c5c	16.2030579217982	45.4654341314984	-73.547947957892	18.324	0	45.4653429772285	-73.548314752136	1999	1010		[array]		
7	point	45.463028	-73.5541059	45.463028	-73.5541059	Legement	1000	mid_daa3774a-19f48c2866-026a42291e5	15.7132418222245	45.4630938025466	-73.554007668611	17.011	0	45.4629589865873	-73.5541510955985	1987	1010	978.867394020127	[array]	44461	141.940141030929
8	point	45.4659358	-73.5481177	45.4659358	-73.5481177	Legement	1000	mid_123611f-694c44679c26-928b789e6c6	16.237481545851	45.4659967476613	-73.5480776441863	20.643	0	45.4658749916976	-73.5482764124668	1998	1010		[array]		
9	point	45.4721821	-73.5402189	45.4721821	-73.5402189	Homeless shelters	6000	mid_0407780c-276c414679c26-4848a4551a13	16.4183263933645	45.4721540964844	-73.540088786894	16.418	0	45.4720102077478	-73.540389627533	2009	2020	3010.4069010416665	[array]	19182	164.0775814027757
10	point	45.4647104	-73.5522685	45.4647104	-73.5522685	Legement	1000	mid_a101c0e-1502-448f-8a6c-38c25209c3	13.1821890554988	45.4647647103803	-73.5528233119314	13.295	0	45.4646561121159	-73.5529137136	1985	1010	848.620050964861	[array]	47649	175.4640093208027

(cont...)

Figure 24 – Resulting feature collection augmented with energy attributes calculated on-demand from SimStadt API

The GNOSIS Map Server's MOAWAdapter adapter process acted as a *Processes – Part 1* client to the SimStadt API in a workflow that also included an *AddAttributes* process conflating the calculated attributes with the original collection, by joining them based on UUID key attributes. The overall processing workflow is illustrated in the flow diagram shown below.

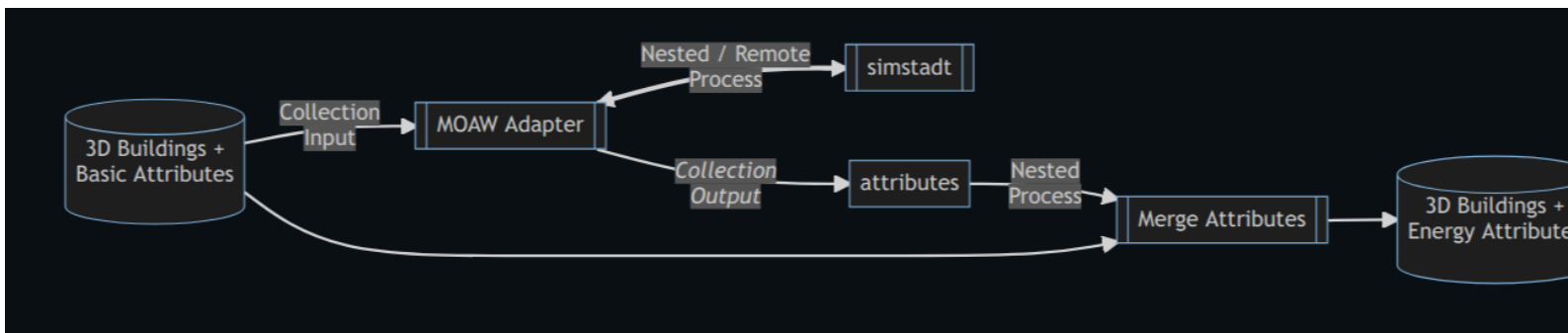


Figure 25 – Flow diagram illustrating use of Proceses - Part 3 to combine multiple nested processes in workflow to create feature collection augmented with energy attributes

The workflow execution request used for this experiment is shown below.

Example – Execution request to augment Nuns' Island dataset with building energy calculation results from SimStadt API

```

{
  "process" : "https://maps.gnosis.earth/ogcapi/processes/AddAttributes",
  "inputs" : {

```

```

    "features" : { "collection" : "https://maps.gnosis.earth/ogcapi/
collections/NunsIsland" },
    "featuresKey" : "gmlID",
    "attributesKey" : "UUID",
    "attributes" :
    {
      "process" : "https://maps.gnosis.earth/ogcapi/processes/MOAWAdapter",
      "inputs" :
      {
        "data" :
        {
          "process" : "https://steinbeis-3dps.eu/ogc-api-processes/
processes/simstadt_api",
          "inputs" : {
            "bbox" : {
              "bbox" : [ -73.5611240061519, 45.4467047317973, -
73.5332045116411, 45.4745479850084 ]
            },
            "interestedAttributes" :
            [
              "heatedVolume",
              "yearOfConstruction",
              "buildingFunction",
              "monthlyHeating",
              "specificSpaceHeatDemand",
              "spaceHeating"
            ]
          }
        }
      }
    }
  }
}

```

This workflow was automatically generated using *GNOSIS Cartographer's* workflow editor which parses process descriptions. In this case, some hard-coding was necessary due to some elements missing from the SimStadt API process description.

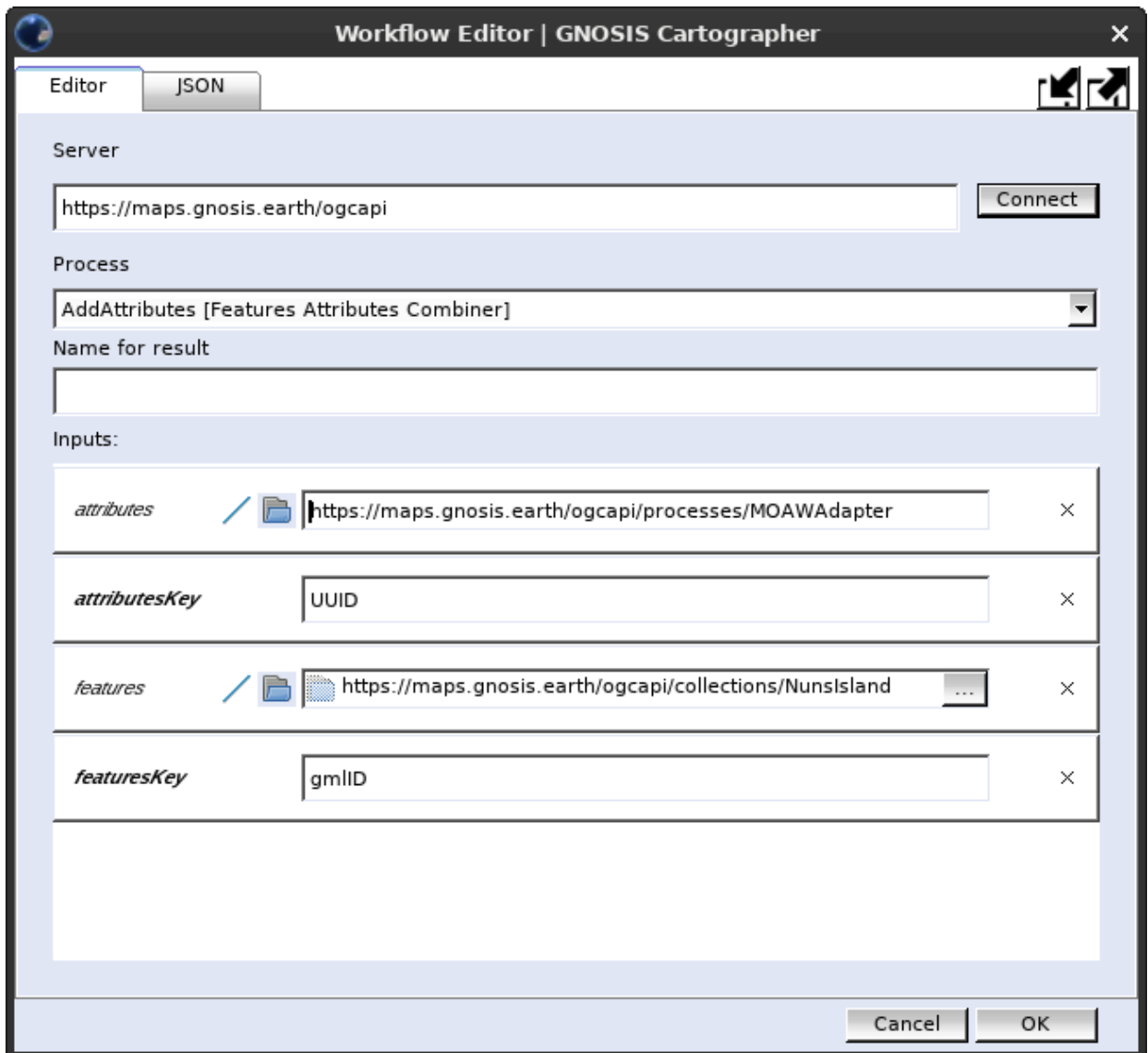


Figure 26 – GNOSIS Cartographer’s Workflow Editor configuring *AddAttributes* process to take as input output from SimStadt API (through MOAW Adapter helper process), joining attributes based on UUID

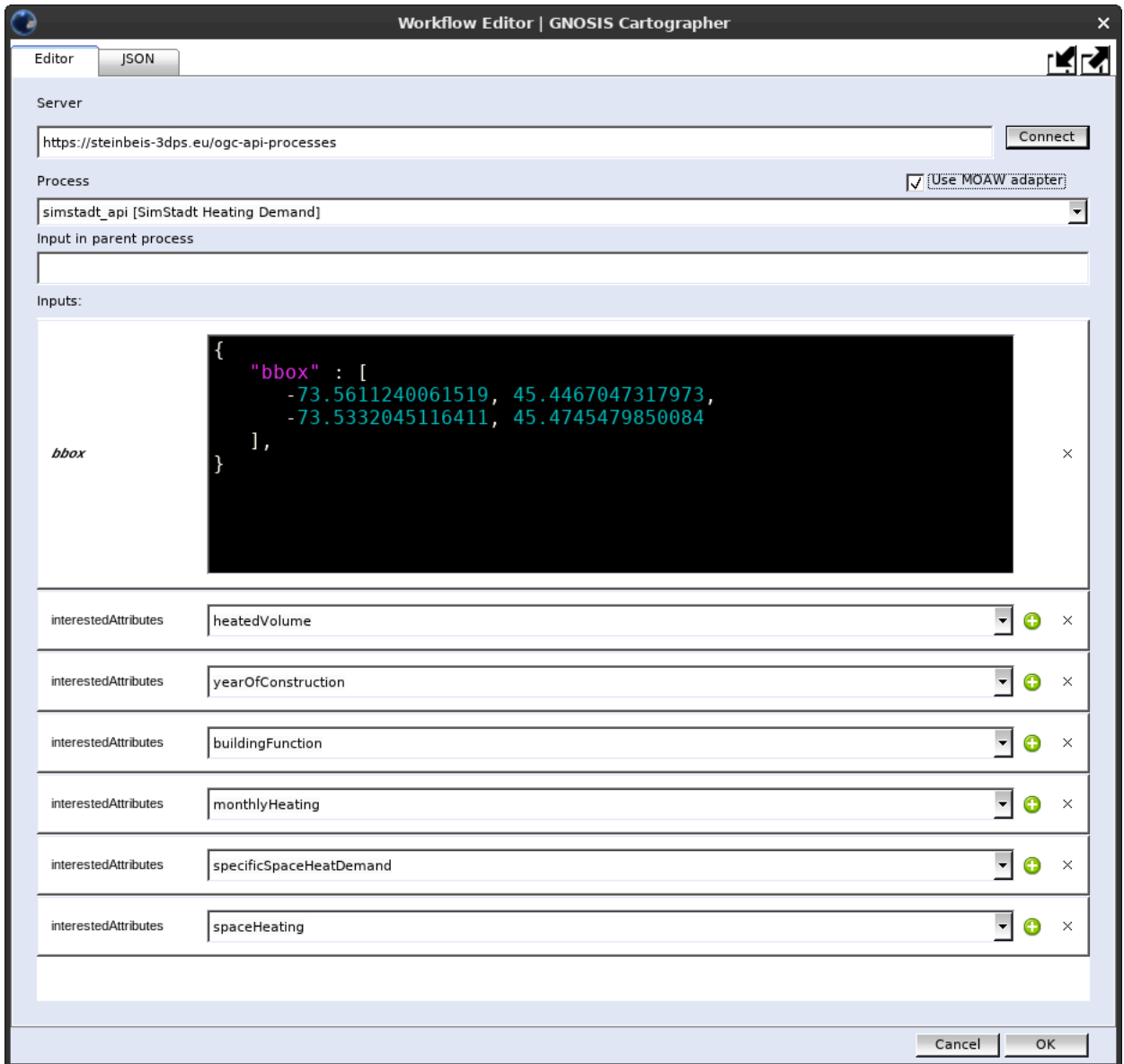


Figure 27 – GNOSIS Cartographer’s Workflow Editor configuring *SimStadt API* process, selecting overall region of interest and energy calculation attributes of interest

This experiment identified some issues to be addressed in terms of conformance to the OGC API – Processes – Part 1: Core Standard that will eventually be resolved.

Another important issue pertained to the amount of time the processing server would take to respond, sometimes even for a small extent. While waiting for this issue to be resolved, pre-computed energy calculation results for *Nuns’ Island* were used in the interim.

Integrating the attributes also required implementing support for the custom JSON energy attributes schema used for the SimStadt API results. Ecere suggested that a GeoJSON feature collection with properties (possibly with null geometry) could potentially be used to facilitate ingesting the output of the SimStadt API into clients already able to consume GeoJSON data.

6.7.4. Remaining Challenges

One of the challenges that Ecere was not able to address in Testbed-18 was integrating the feature attributes, including those computed by the energy calculation workflow using the SimStadt API, directly within the glTF models. The difficulty was that this would likely have required changes to the [Open Asset Import library](#) used to export them. However, some work already done in that direction (see [this issue](#) and [pull request](#)) indicates that this may be feasible using the *aiMetadata* structure.

Another challenge that would have been interesting to take on given more time was to integrate and visualize a more accurate terrain elevation model derived from the high resolution LiDAR point cloud data provided by the City of Montreal together with the 3D buildings data. Some experiments were done with the quantized mesh provided by HfT Stuttgart and the *Nuns' Island* subset of the Montreal 3D buildings data. However, for the version of that dataset having identifiers compatible with the SimStadt API, the elevation values were configured with zero values altitudes. With an accurate elevation model and proper buildings elevation values, the visualization would not have suffered from buildings appearing under or floating above the ground.

The buildings sourced from the original City of Montreal dataset included textures. However, the final prepared dataset including corrections and additional buildings did not preserve the textures, since the project focused more on presenting buildings colored by attributes values rather than realistic rendering. Developing support for importing textures from a CityGML dataset into a *GNOSIS Data Store* is functionality that remains to be implemented, which could then allow streaming these textured datasets as glTF or 3D Tiles.



7

TESTBED EXPERIMENTS – EXTENDED PROCESSING TRACK

TESTBED EXPERIMENTS – EXTENDED PROCESSING TRACK

7.1. Description

The extended processing track focused on a more general processing workflow than the initial processing track, using data from Montreal, Canada. While the initial processing track had a goal of providing a service using only available building data, the extended processing track sought to generalize that workflow for the entire City of Montreal and the parcels and buildings contained therein. There was enough data on buildings and their characteristics provided within the Testbed from a variety of sources (Montreal property assessment data, CityGML, EnerGuide evaluations in HOT2000 file format, and HTAP Archetypes in HOT2000 file format) that each building's characteristics could potentially be estimated and attributed to the street or civic addresses via its latitude and longitude or point. From there, the energy consumption of each building or parcel containing one or more buildings could be simulated by the Steinbeis Consortium's (or other participants') Building Energy Processing Service and subsequently attributed to civic address points.

7.2. Test Locations

This track focused on the City of Montreal in its entirety rather than only on Nuns' Island.

7.3. Test Data

The following datasets were used by the participants in the extended processing track.

- City of Montreal 3D buildings 2016 (detailed textured building geometries in CityGML LoD2 for the Côte-des-Neiges–Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun neighborhoods).
- Montreal agglomeration property roll assessment data and codification of use (attributes on building function, year of construction, building category, neighborhood name).
- Automatically Extracted Buildings from Natural Resources Canada
- City of Montreal digital terrain model (DTM) (building elevation data with a planimetry of ± 20 cm and altimetry of ± 20 cm)

- Geoindex shared platform (Province of Quebec)
- Montreal’s City of Montreal Building, alteration and demolition permits
- NRCan CanmetENERGY HTAP Archetype Database
- NRCan CanmetENERGY Reduced HOT2000 File Representation

7.4. Extended processing track summary

The following sections detail the participants’ roles and interactions within this work track and are divided by the type of web service being provided by the participant to the track task. Each web service section describes the activities that occurred during the testbed to complete the deployment of the service as well as any relevant comments, conclusions, and results. A description of the actions performed by the participants within this track follows in the table below.

Table 12 – Extended processing track summary

PARTICIPANT ROLE	PARTICIPANTS	DESCRIPTION
Building Energy Processing Service (D126)	Steinbeis Consortium	The Steinbeis Consortium contributed to the Extended Processing Track by preparing and distributing the 3D building stock dataset covering the entire City of Montreal (34 neighborhoods) in CityGML data format for building energy simulation. The activity built upon the work on Nuns’ Island during the Initial Processing Track. Inputs from the City of Montreal CityGML LoD 2 (6 boroughs), Montreal agglomeration property roll assessment, Automatically Extracted Buildings from NRCan, and LiDAR data (Montreal DTM) data were conflated to synthesize a hybrid CityGML LoD 1 & 2 dataset for the entirety of Montreal.
Building Energy Data Service (D122 and D123)	interactive instruments	The activity provided OGC compliant Web API to access the City of Montreal hybrid CityGML LoD 1 & 2 dataset. An API implementation was provided for accessing City of Montreal building data as 3D Tiles, Vector Tiles, and Features with multiple data encodings. An API was also provided to access NRCan housing archetypes from the HTAP Archetype Database for the Province of Quebec in Reduced HOT2000 File Representation data format.

7.5. Building Energy Processing Service (Steinbeis Consortium)

Concordia University, part of the Steinbeis Consortium, prepared the data for the extended process processing track from the following data sources.

- City of Montreal 3D buildings 2016 (detailed textured building geometries in CityGML LoD2 for the Côte-des-Neiges–Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun neighborhoods)
- Montreal agglomeration property roll assessment data and codification of use (attributes on building function, year of construction, building category, neighborhood name)
- Automatically Extracted Buildings from Natural Resources Canada
- City of Montreal digital terrain model (DTM) (building elevation data with a planimetry of ± 20 cm and altimetry of ± 20 cm)

7.5.1. Montreal data acquisition, cleaning, and conflation

The City of Montreal 3D building dataset for Nuns' Island served as the primary data source for different geospatial and building energy web services implemented within the Initial Processing Track. These services and related API endpoints were further enriched in the Extended Processing Track to cover the entire City of Montreal. A product of the Extended Processing Track was a 3D building dataset of Montreal in CityGML containing a combination of LoD 1 and 2 formats. A hybrid CityGML LoD 1 and 2 dataset was generated due to the unavailability of a 3D building stock dataset within the Testbed that covered the entire City of Montreal (34 neighborhoods, also known as the urban agglomeration of Montreal). A high-level flowchart representing the data conflation process used to produce the City of Montreal CityGML LoD 1 and 2 dataset is shown in the figure below.

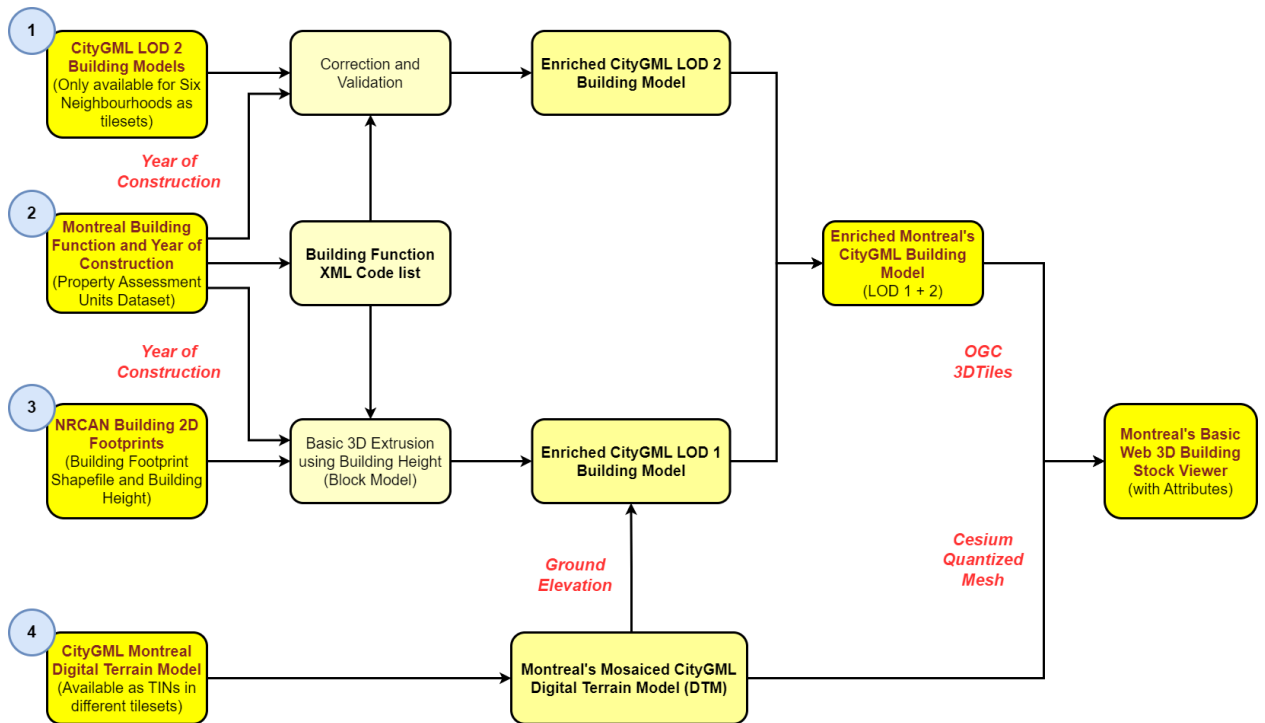


Figure 28 – Data conflation process behind production of Montreal wide CityGML building stock dataset.

The City of Montreal 3D city model dataset is openly available and can be obtained as textured CityGML LoD 2 files from the open data portal of the City of Montreal. The CityGML dataset was only available for the Côte-des-Neiges-Notre-Dame-de-Grâce, Outremont, Plateau-Mont-Royal, The Southwest, Ville Marie, and Verdun neighborhoods within the City of Montreal. During the conflation, the textures from the original CityGML data were removed to reduce the file size. The gml IDs of certain buildings and surface polygons within the original CityGML tilesets into the merged CityGML dataset, violating the constraints of XML/GML encoding that require an identifier to always start with a letter. Subsequently, using [SAFE Software's Feature Manipulation Engine \(FME\)](#) new building GML IDs were produced for the entire CityGML datasets. A copy of the original building GML IDs were kept for referring back to the original dataset whenever required. Additionally, CRS metadata was found to be missing in the original dataset. Thus, the processed CityGML dataset was correctly referenced to its native vertical datum NAD83 CRS (CSRS)/MTM zone 8 + CGVD28 height (a compound CRS of EPSG:2950 and EPSG:5713). As a prerequisite for usage in building energy simulation APIs, the merged CityGML dataset needed to also contain CityGML attributes for each building's function and year of construction. Therefore, the processed CityGML dataset for the six neighborhoods of Montreal was then further enriched with the attributes of building year of construction and building function, the CODE_UTIL and ANNEE_CONSTRUCTION attribute fields from Montreal's open property assessment unit shapefile dataset, respectively. As part of the enrichment process, a building function code list was also developed based on a publicly available Montreal agglomeration property roll assessment data and codification of use (building function) and integrated with the CityGML model of Montreal.

A snippet view of Montreal's building function XML codelist is shown below. A complete list of Montreal's building function XML codelist is published at https://transfer.hft-stuttgart.de/pages/montreal-citygml/codelist/Building/_AbstractBuilding_function.xml.

Example

```
<gml:Dictionary xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/gml http://schemas.opengis.net/gml/3.1.1/profiles/SimpleDictionary/1.0.0/gmlSimpleDictionaryProfile.xsd" gml:id="_AbstractBuilding_function">
  <gml:name>_AbstractBuilding_function</gml:name>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="id1">
      <gml:description>Logement</gml:description>
      <gml:name>1000</gml:name>
    </gml:Definition>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry>
    <gml:Definition gml:id="id2">
      <gml:description>Chalet ou maison de villégiature</gml:description>
      <gml:name>1100</gml:name>
    </gml:Definition>
  </gml:dictionaryEntry>
</gml:Dictionary>
<gml:dictionaryEntry>
  <gml:Definition gml:id="id3">
    <gml:description>Maison de chambres et pension</gml:description>
    <gml:name>1510</gml:name>
  </gml:Definition>
</gml:dictionaryEntry>
```

To generate a CityGML LoD 1 model for the remaining neighborhoods in the City of Montreal, openly available building footprints in shapefile format from NRCan were used. The model was generated for the remaining neighborhoods in the City of Montreal by extruding building footprints with building height values. NRCan published the building footprints in a geographic CRS of NAD83 (EPSG: 4617). Therefore, to correctly align the building footprints with the merged City of Montreal CityGML dataset (six neighborhoods) CRS, the footprints were reprojected to NAD83(CSRs)/MTM zone 8 (EPSG: 2950).

The building height values for each building footprint were derived by from the heightmax attribute provided with the building footprint shapefile. To obtain the correct ground elevation for each building, the extruded building footprints were overlaid on the mosaiced Montreal Digital Terrain Model and adjusted based on the calculated mean terrain elevation for each building footprint. The terrain adjusted 3D buildings were given a vertical CRS (CGVD28, EPSG: 5713) and conflated with the building function and year of construction attributes from the Montreal's property assessment unit dataset.

The building footprint IDs (feature_id) of many building footprints were found to start with numbers which violated the XML/GML encoding constraints requiring an identifier to always start with a letter. Subsequently, new GML IDs were produced, and the extruded building footprints were converted to a CityGML LOD 1 building model. A copy of the original building GML IDs were kept in the produced CityGML dataset for referring back to the original building footprint shapefile dataset as required.

Ultimately, both the CityGML datasets (LOD 1 and LOD 2) were merged to generate a single hybrid CityGML LoD 1 and 2 building stock model covering the entire City of Montreal.

Additional attributes such as building parcel IDs, building categories, and neighborhood name were also conflated from the Montreal's property assessment dataset to further enrich the City of Montreal CityGML model.

A [CesiumJS based web visualization](#) conforming to the OGC 3D Tiles standard, was developed to showcase the CityGML dataset for the entire City of Montreal and is shown in the figure below.

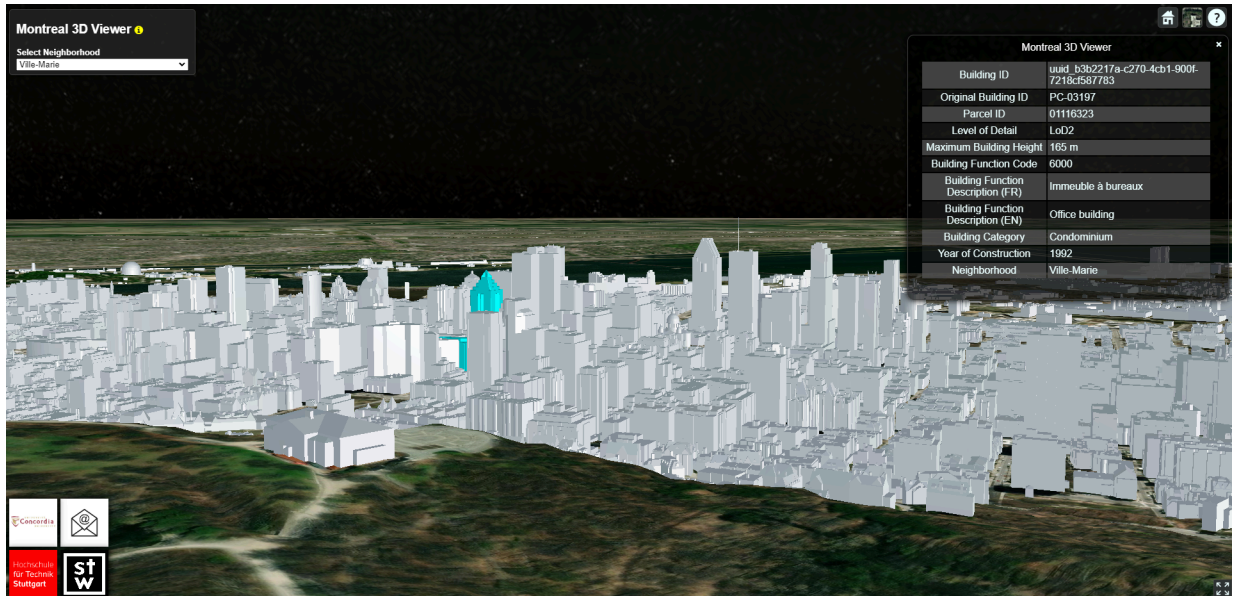


Figure 29 – Montreal web 3D viewer.

7.5.2. Challenges Faced

Within the Extended Processing Track, the Steinbeis Consortium faced a number of challenges. To start, the primary reason for producing hybrid CityGML LoD 1 and 2 dataset within this track was the unavailability of a City of Montreal 3D building dataset. Four different open datasets were processed and conflated to generate a City of Montreal CityGML building dataset appropriate for building energy simulation.

Previous challenges from the Initial Processing Track, such as missing CRS, unavailability of building function and year of construction in the original CityGML LoD 2 model, invalid gml IDs, and geometrical issues persisted. In addition, the building footprints shapefile that was used to fill in the CityGML 3D model for remaining neighborhoods in the City of Montreal also contained building footprint IDs invalid for use in CityGML, in addition to geometrical issues, different CRS, and missing attributes on building function and year of construction.

The majority of these issues were identified and solved during the process of conflating the City of Montreal 3D building dataset in CityGML. However, similar to the Nuns' Island dataset, issues with building geometries and the fact that the building functions and year of construction were conflated from Montreal's property assessment unit, which only provides general information on the use of property, year of construction, category, approximate dimensions, and registration number on a **land parcel level** (not building level), still exists.

7.6. Building Energy Data Service (interactive instruments)

interactive instruments provided a Building Energy Data Service instance. For the Extended processing track, the services were accessed using the following OGC compliant Web APIs.

- An API for accessing building data for the City of Montreal.
- An API for accessing NRCan Housing Archetypes for the Province of Quebec.

The Building Energy Data Service APIs were deployed using [Idproxy](#). Idproxy allows a developer to quickly set up Web APIs that make geospatial data available via HTTP.

The APIs created by interactive instruments created for the Testbed-18 will be available until at least June 30th, 2023. The URLs documented within this section should resolve at least until that date.

Consistent with the recommendations for OGC Web APIs, the API implementations also supported web browsers as API clients. That is, the resources provided by each API must include:

- the resources can be reached through links from the landing page of the API; and
- the resources are available in HTML that can be presented by the web browser to the user.

7.6.1. API implementation for accessing City of Montreal building data

The API implementation described within this section provided access to City of Montreal building data.

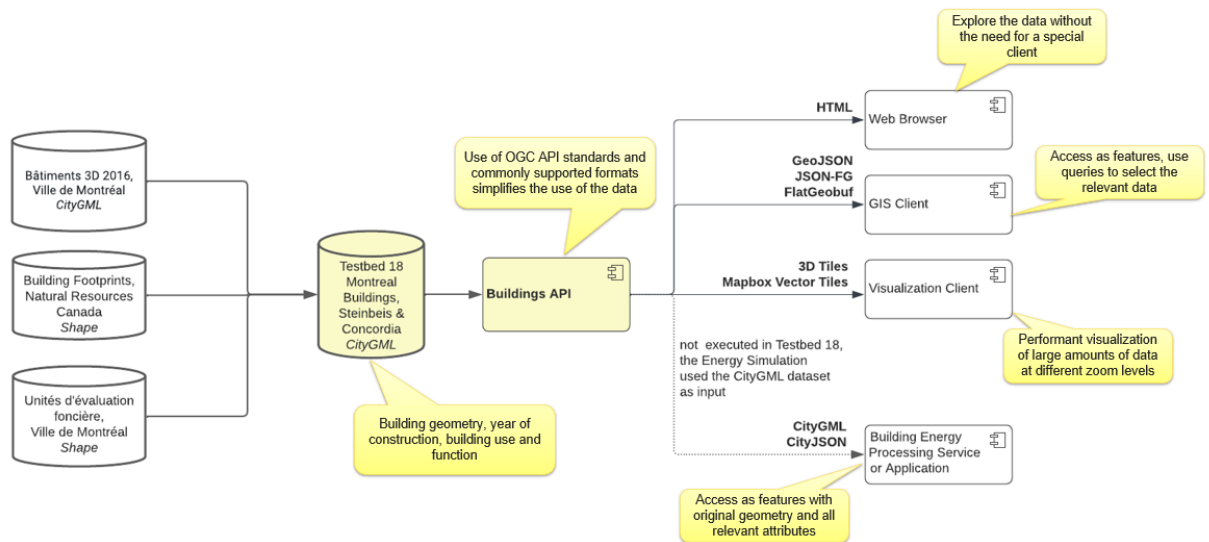


Figure 30 – The extended processing track workflow for accessing the City of Montreal data.

The remainder of the section describes the resources provided by the API with their representations.

7.6.1.1. Standard OGC Web API resources

As an OGC Web API, the API provided the following resources:

- a [Landing Page](#) for the API that provides basic information about the API and links to the main sub-resources in the API;
- a [Conformance Declaration](#) that identifies the OGC Conformance Classes implemented by the API, identified through their URI; and
- a [API description](#) based on the OpenAPI 3.0 standard, the HTML representation uses the Swagger UI library.

7.6.1.2. 3D Tiles

The constructed API for this track is based on existing and draft OGC API specs/standards whose conformance classes are used as building blocks to implement the API instance used in this Testbed. The relevant API building blocks implemented in the API for accessing the buildings as 3D Tiles are specified in the draft [OGC API – 3D GeoVolumes, Editor’s Draft](#).

The OGC Community Standard [3D Tiles 1.1](#) with [gITF 2.0](#) content was implemented for Testbed-18 as a data encoding with the following capabilities and extensions.

- Implicit [quadtree tiling](#) was used for the tileset. The first level (level 0) consists of one tile covering the data extent. For each subsequent level there are four tiles for each parent tile.

That is, on level 1 there are four tiles, on level 2 there are 16 tiles, etc. 3D Tiles supports various options for how to organize the data. The Montreal data tileset was configured as follows.

- Based on the extent of the dataset (City of Montreal), seven levels were configured (levels 0 to 7).
- The first level with content was level 3 (at levels 0 to 2 the scale was too small for buildings).
- There were 3 levels in each subtree (a subtree is a resource to indicate to clients which tiles have content).
- The additive (ADD) refinement strategy was used (3D Tiles supports REPLACE where the content of a tile replaces all content included in the parent tile and ADD where the content of a tile adds to the content of the ancestor tiles). The level on which a building was placed depended on the diagonal of the 3D bounding box of the building.
- For tiles with content, the content was provided as a glTF 2.0 Binary file.
- The glTF extension EXT_mesh_features was used to associate each vertex with a feature. Originally the implementation supported two different feature identifiers. One for each vertex (because different vertices may have been associated with different surface types) and one for each building (because no other attributes varied by vertex). However, this approach was changed to a single feature identifier by vertex, because Cesium JS at the time of Testbed-18 could only access attributes of a single set of feature identifiers and clients needed access to the surface type as well as the other building attributes.
- The glTF extension EXT_structural_metadata was used to associate attributes with each feature. All attributes in the source data were included in the tile for each building.
- The glTF extension KHR_mesh_quantization was used to reduce the data volume by using integer values for vertex coordinates (data type: short) and normals (data type: byte).

The 3D Tiles Community Standard used for the Testbed uncovered several problems that were added as issues in the 3D Tiles GitHub repository:

- Collection: “contentExtent” property
- Collection: “children” and “content” properties
- Media types

Sample requests:

Example – The root tile of the 3D Tiles tileset (“tileset.json”)

```
{
  "asset": {
    "version": "1.1",
    "generator": "ldproxy",
```

```

    "copyright": "HFT Stuttgart, Natural Resources Canada, Ville de
Montreal - Service des Infrastructures du réseau routier - Division de la
géomatique, Ville de Montreal - Service de l'évaluation foncière"
  },
  "geometricError": 10000,
  "root": {
    "boundingVolume": {
      "region": [
        -1.2911393430159392,    0.7924168563830882,    -
1.2824542632150329,
        0.7976526494340113, -32.11120000000051978,
231.9489499999999964
      ]
    },
    "geometricError": 2048,
    "refine": "ADD",
    "content": {
      "uri": "https://d123.ldproxy.net/montreal/collections/buildings/
3dtiles/content_%7Blevel%7D_%7Bx%7D_%7By%7D"
    },
    "implicitTiling": {
      "subdivisionScheme": "QUADTREE",
      "availableLevels": 8,
      "subtreeLevels": 3,
      "subtrees": {
        "uri": "https://d123.ldproxy.net/montreal/collections/
buildings/3dtiles/subtree_%7Blevel%7D_%7Bx%7D_%7By%7D"
      }
    }
  }
}
}
}

```



Figure 31 – The tileset visualized in [Cesium JS](#)

In the rendering, surfaces that were tagged as roofs in the data were shown in red to differentiate them from the other surfaces, shown in gray.

7.6.1.3. Vector Tiles

The [OGC API – Tiles – Part 1: Core Standard](#) specified the conformance classes required to implement the API for accessing the buildings as Mapbox Vector Tiles.

[Mapbox Vector Tiles 2.0](#) was supported as a data encoding.

The link template for the tiles was <https://d123.ldproxy.net/montreal/tiles/WebMercatorQuad/{z}/{y}/{x}?f=mvmt>.

Tiles were provided for the City of Montreal in the WebMercatorQuat tiling scheme for levels 11 to 18. The building footprint was used for the building geometry. All attributes in the source data were included for each building.

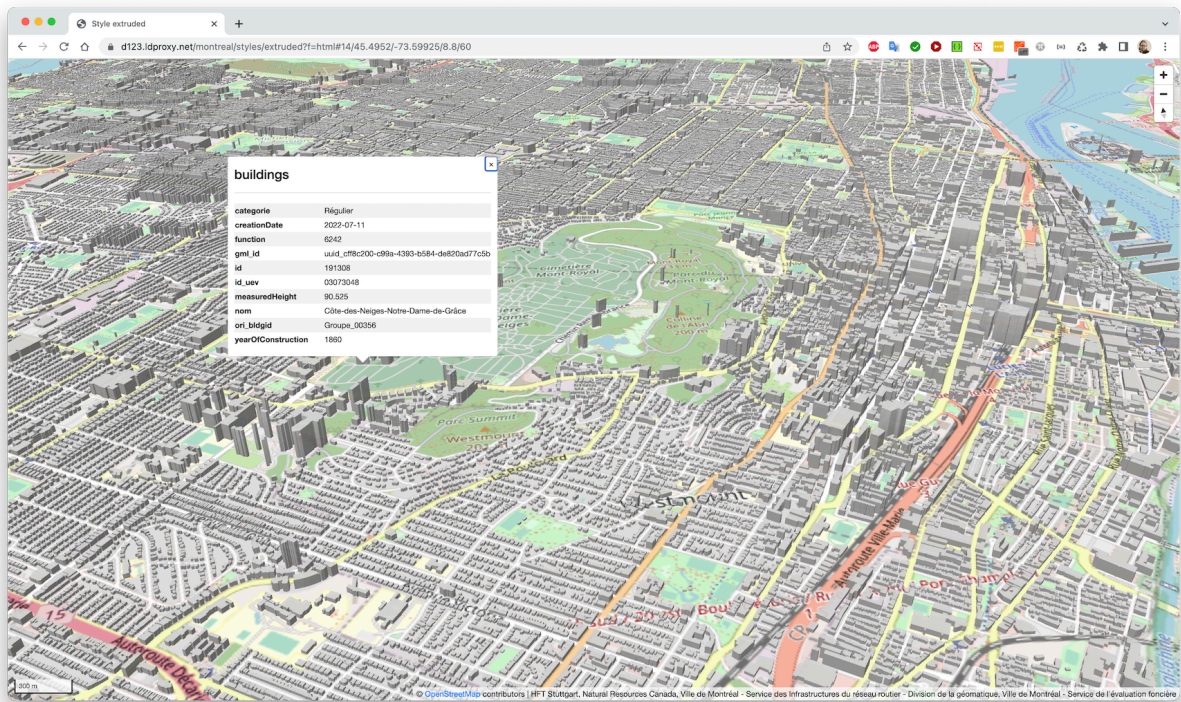


Figure 32 – Sample map with the Mapbox Vector Tiles in [MapLibre JS](#)

7.6.1.4. Features

The relevant OGC Standards (API building blocks) for accessing the buildings as features were:

- [OGC API – Features – Part 1: Core 1.0.1](#)
- [OGC API – Features – Part 2: Coordinate Reference Systems by Reference 1.0.1](#)
- [OGC API – Features – Part 3: Filtering, Editor's Draft](#)

- [Common Query Language \(CQL2\), Editor's Draft](#)

In addition, several experimental API extensions were supported.

Multiple data encodings were supported.

- [CityJSON 1.1](#), both as a single JSON document or as CityJSON text sequences
- [CityGML 2.0](#)
- [glTF 2.0 Binary](#)
- [GeoJSON](#)
- [JSON-FG 0.1](#)
- [FlatGeobuf](#)
- CSV

The CityJSON and CityGML encodings provided all the building information in the source data. The glTF representation had the same characteristics as the glTF models encoded in the 3D Tiles tileset. The GeoJSON and FlatGeobuf representations provided the building footprint as the primary geometry, the JSON-FG representation provided the LoD 1 or LoD 2 solid as the primary geometry; no semantics of the boundary surfaces was provided in these encodings. Finally, a CSV representation was provided that did not include any building geometry.

The following table shows the file size of the uncompressed response for each format for two neighborhoods in the City of Montreal. The buildings in the Westmount district are provided at LoD 1. The buildings in the Outremont district are provided at LoD 2.

Table 13 – Comparison of data volumes in MB by format

DISTRICT	CITYJSON	CITYJSON TEXT SEQUENCES	CITYGML	GLTF	JSON-FG	FLATGEOBUF
Westmount (LoD 1)	18.3	13.4	132	25.2	96	2.6
Outremont (LoD 2)	3	2.6	13.9	5.6	7.7	1.1

Sample requests:

Buildings

Filter nom=Ville-Marie

« < 1 > »

function	Immeuble commercial
yearOfConstruction	1861
measuredHeight	50.79700885009766
gml_id	uuid_e24945e8-b55f-4c08-ba83-1f0a7c37d104
nom	Ville-Marie
categorie	Régulier
id_uev	01036846
ori_bldgid	1340631
surfaceType	wall

« < 1 > »

Logement

Internal ID	247696
UUID	uuid_cc2af44e-c594-45b4-8123-a57f83c56784
Creation Date	2022-07-12
Unit Category	Régulier
UEF Property ID	01040366
Arrondissement	Ville-Marie
Original ID	2767637
Function	Logement
Year Of Construction	1865
Measured Height [m]	14.106999999999993

NOTE: The screenshots included a mix of English and French. The French text was from the source datasets, the English text was from the API. The browser requested that the language of the response be provided in English, but the building attributes were only available in French. If attribute content in the source data was multilingual (e.g., all linguistic text or controlled vocabularies would be available in both English and French), the responses could have all been in French or English.

Figure 33 – Buildings in Ville-Marie build before 1870 ([HTML](#), [gITF-Binary](#), [CityJSON](#))

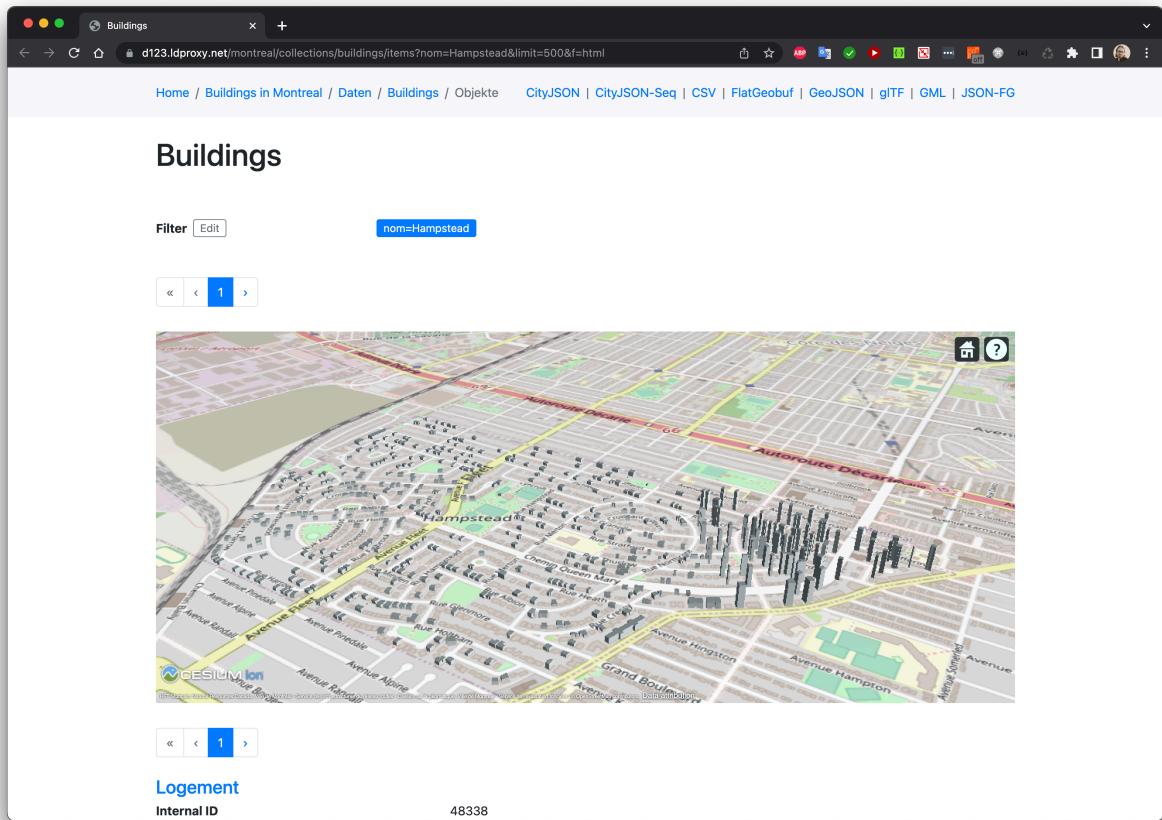


Figure 34 – 500 buildings in Hampstead (HTML, gITF-Binary, CityJSON)

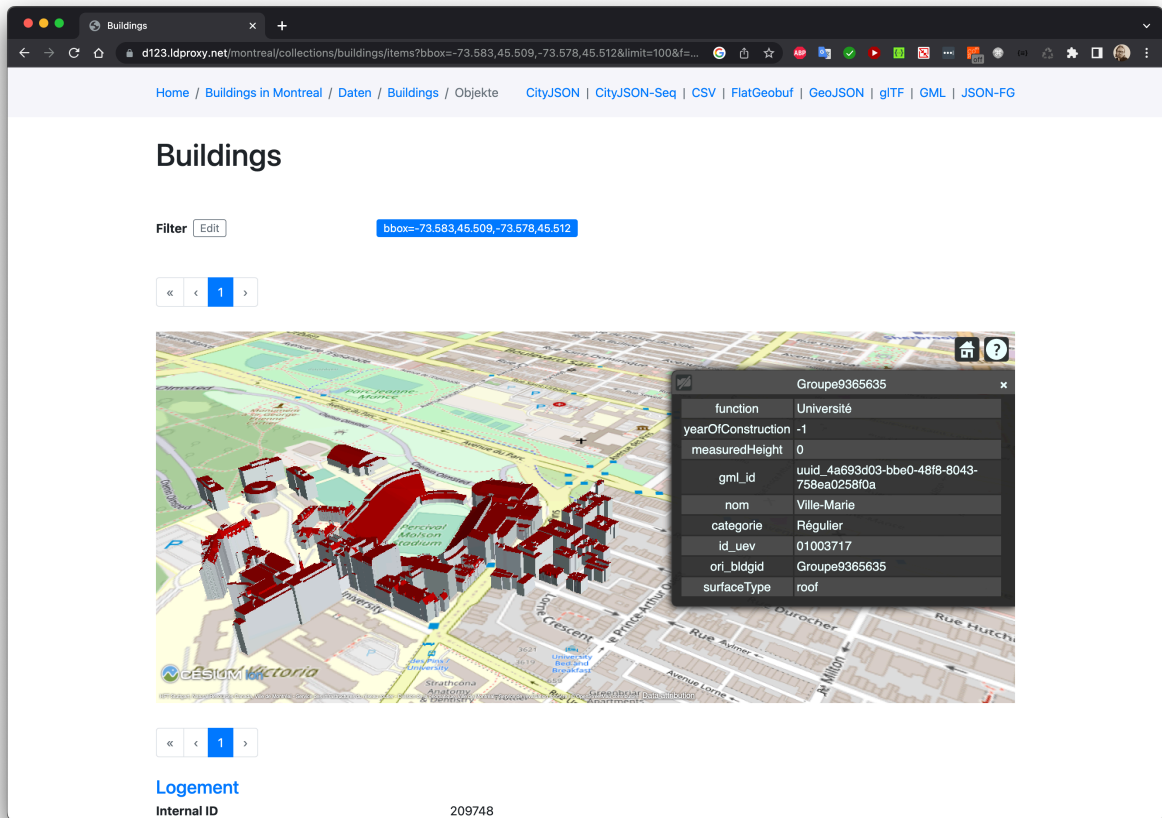


Figure 35 – Buildings in a small bounding box around Percival Molson Stadium (HTML, gITF-Binary, CityJSON)

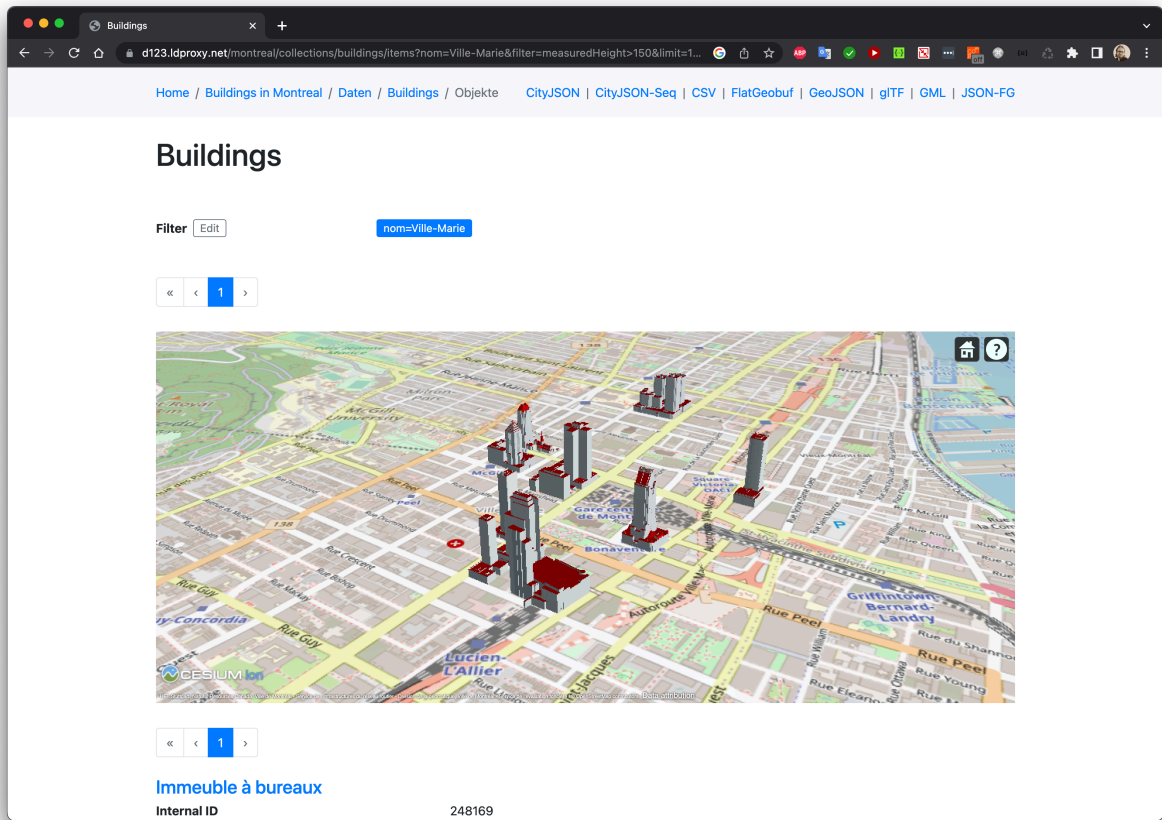


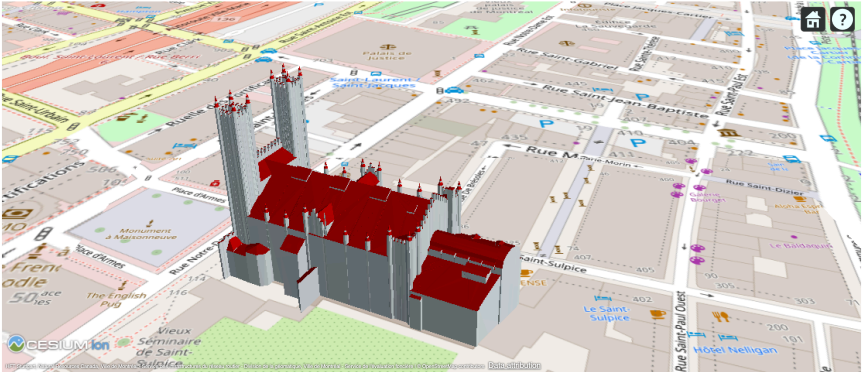
Figure 36 – Buildings taller than 150 meters in Ville-Marie ([HTML](#), [gITF-Binary](#), [CityJSON](#))

Église, synagogue, mosquée x +

d123.ldproxy.net/montreal/collections/buildings/items/248460?l=html

Home / Buildings in Montreal / Datas / Buildings / Objekte / Église, synagogue, mosquée et temple

CityJSON | CityJSON-Seq | CSV | FlatGeobuf | GeoJSON | glTF | GML | JSON-FG



Église, synagogue, mosquée et temple

Internal ID	248460
UUID	uuid_9ea4602f-46d1-4dae-a786-4f5f43a2b5f9
Creation Date	2022-07-12
Unit Category	Régulier
UEF Property ID	01091097
Arrondissement	Ville-Marie
Original ID	2104973
Function	Église, synagogue, mosquée et temple
Year Of Construction	1829
Measured Height [m]	73.19599999999998

Figure 37 – Notre-Dame (HTML, glTF-Binary, CityJSON)

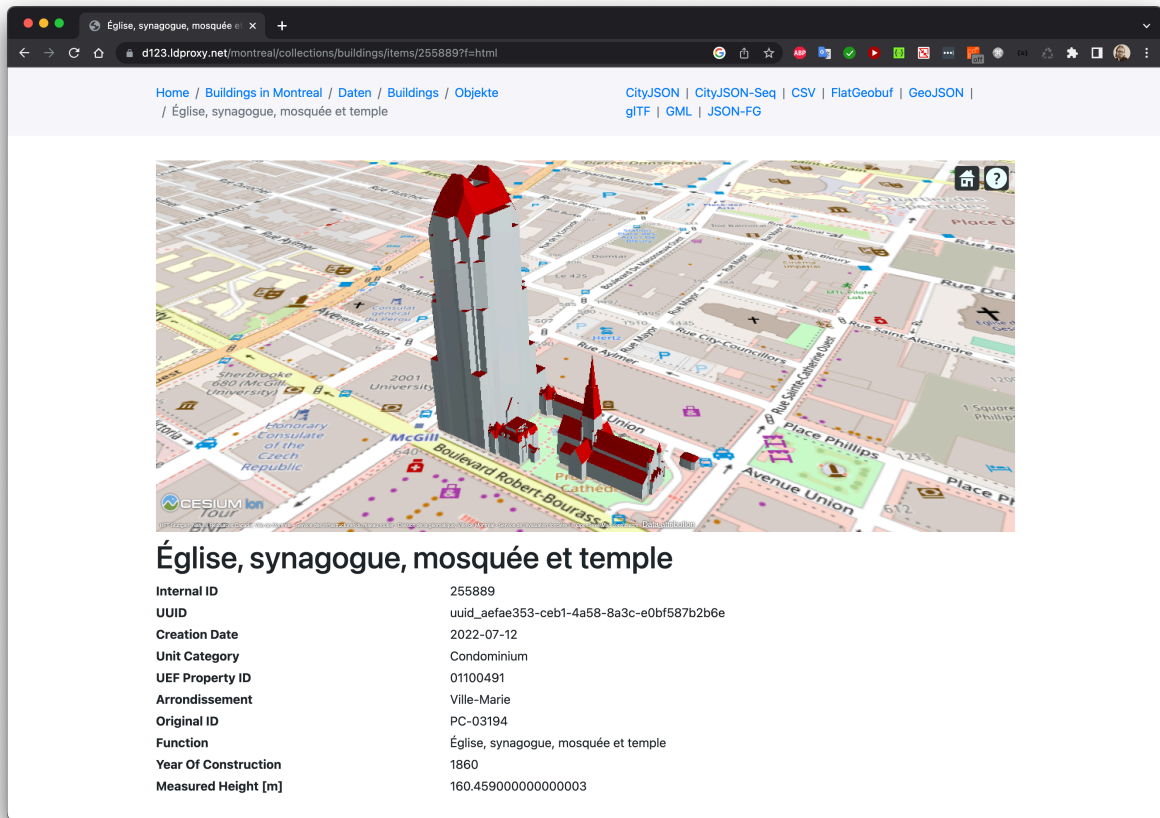


Figure 38 — A single building feature in the dataset, that is actually a collection of buildings including a church and an office building ([HTML](#), [gITF-Binary](#), [CityJSON](#))

Example 1 — Fetch buildings in a bounding box using the 'bbox' parameter (west,south,east,north) ([GeoJSON](#), [CSV](#))

```
id,gml_id,creationDate,nom,categorie,id_uev,ori_bldgid,function,
yearOfConstruction,measuredHeight
239773,uuid_e3047148-0081-4b71-a89f-cb665136f04e,2022-07-14,Verdun,
Condominium,01123405,V03_00154,1000,1993,14.924
244652,uuid_cce7bc5f-fd27-4962-97a5-e0c92562bc70,2022-07-14,Verdun,
Condominium,01124643,V03_00153,1000,1993,14.202
247556,uuid_36d6e06d-2518-487a-a66f-87cc3bb3821e,2022-07-14,Verdun,
Condominium,01123405,V03_00155,1000,1993,14.655
```

Example 2 — Fetch the building at a location (if any) using the non-standardized 'intersects' parameter with a Well-known Text point geometry ([GeoJSON](#), [CSV](#))

```
id,gml_id,creationDate,nom,categorie,id_uev,ori_bldgid,function,
yearOfConstruction,measuredHeight
9,uuid_df188a90-83f1-4c91-b9db-7b74326b1698,,Ahuntsic-Cartierville,
Régulier,03059627,df188a90-83f1-4c91-b9db-7b74326b1698,1000,1960,8.49
```

Example 3 — Same as before, but restricting the response to the various building identifiers using the 'properties' parameter ([GeoJSON](#), [CSV](#))

```
id,gml_id,id_uev,ori_bldgid
```

9,uuid_df188a90-83f1-4c91-b9db-7b74326b1698,03059627,df188a90-83f1-4c91-b9db-7b74326b1698

Example 4 – Same as before, but suppress the geometry in the GeoJSON representation (GeoJSON)

```
{
  "type": "FeatureCollection",
  "numberReturned": 1,
  "timeStamp": "2022-11-02T13:32:55Z",
  "features": [
    {
      "type": "Feature",
      "id": 9,
      "geometry": null,
      "properties": {
        "gml_id": "uuid_df188a90-83f1-4c91-b9db-7b74326b1698",
        "id_uev": "03059627",
        "ori_bldgid": "df188a90-83f1-4c91-b9db-7b74326b1698"
      }
    }
  ],
  "links": [ ... ]
}
```

7.6.1.5. Known issues

There were a number of issues that led to problems in using the City of Montreal data concerning the adherence to the rules of the gml:LinearRing. The Linear Ring is the key element used to represent the boundaries of the polygons that describe 3D geometry in CityGML. A linear ring is a sequence of points defining the boundary of a polygon where (i) the first and last points represent the same point and (ii) all points of the sequence besides the start and end point are different. The City of Montreal data had the following inconsistencies with the definition of a linear ring.

- Polygons in the source data were often not planar, which violated the geometry requirements.
- Sometimes the ring geometries had consecutive identical points or polygons where all points were collinear. These were ignored when importing the data.

These additional issues were observed.

- Often multiple buildings were represented as a single building in the source dataset. The [Notre-Dame-Des-Neiges cemetery](#) was one example where many buildings are bundled as a single building feature.

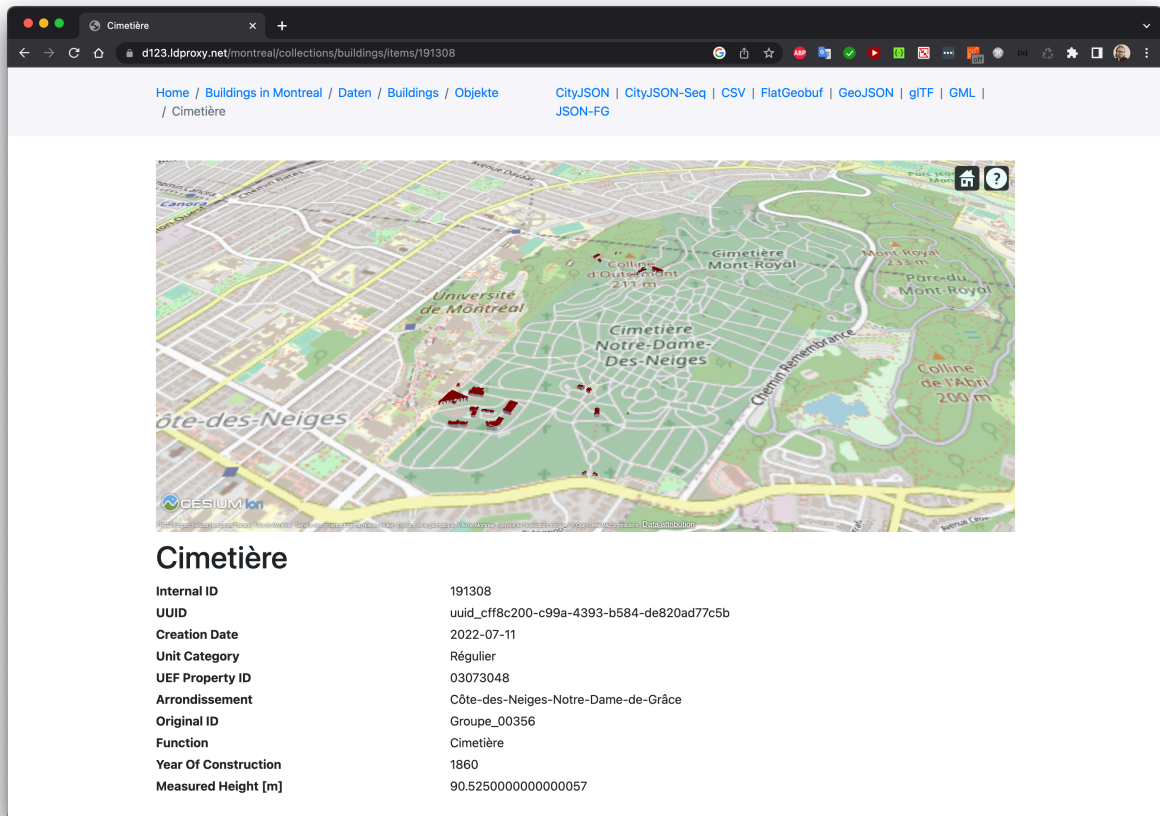


Figure 39 – Example where many buildings are bundled as a single building

- The LoD 2 building data violated constraints of the CityGML model. The LoD 2 solid geometries claimed to be solids, but in general were not solids. The solid geometry just referenced a set of polygons, but often these were not the patches of a single closed composite surface. That is, the `gml:Solid` elements did not conform to the requirements of GML (or ISO 19107). To fix this issue, the building solids would need to be analyzed and if the bounding surfaces form multiple shells/solids, then that building would need to be split into multiple buildings.
- There were a few hundred buildings in the dataset that were missing bounding surfaces where the solid geometry referenced bounding surfaces in other buildings. This resulted in having the same solid geometry for multiple buildings. This caused issues such as mismatched attributes and some bounding volume / visibility issues as well. Essentially, wherever there were references from the solids to bounding surfaces from other buildings, those bounding surfaces should be moved to the bounding surfaces of the building from where they were being referenced in the solid and removed from the building where they were not referenced by the solid.
- The CityJSON data in the example requests above was in the compound coordinate reference system NAD83(CSRS)/UTM zone 18N + CGVD2013 height (EPSG:6661), not the native CRS of the dataset NAD83(CSRS)/MTM zone 8 + CGVD28 height (a compound

CRS of EPSG:2950 and EPSG:5713), because CityJSON and OGC API Features both required a pre-defined compound EPSG or OGC CRS.

7.6.2. Quebec Housing Archetypes API

The interactive instruments for consumption of Quebec Housing Archetypes API provided clients access to 1644 housing archetypes that statistically represent dwelling characteristics and energy performance of low-rise Canadian residential buildings within the Province of Quebec. The Quebec archetypes were from the Canadian HTAP archetypes, filtered to select only those within the province of Quebec. The Quebec Housing Archetypes API serves as an example of how NRCan could make the HTAP archetypes more FAIR (findable, accessible, interoperable, reusable).

The individual records were anonymized and detailed location information was removed. The archetypes are intended to support energy analysis but may prove useful for other applications such as deriving additional building characteristics to supplement property assessment and 3D data in support of energy analysis for individual buildings.

NRCan developed the archetype housing models by consolidating data from the following two sources.

- Audits of Canadian homes completed through the EnerGuide rating program ([more information](#)).
- The Survey of Household Energy Use (SHEU), administered by Statistics Canada ([more information](#)).

The archetype data used within the Testbed differed slightly from the data in the NRCan HTAP archetype database of HOT2000 files. The data served by the Quebec House Archetype API was provided in the Reduced HOT2000 File Representation to reduce the number of attributes. A diagram of the workflow follows.

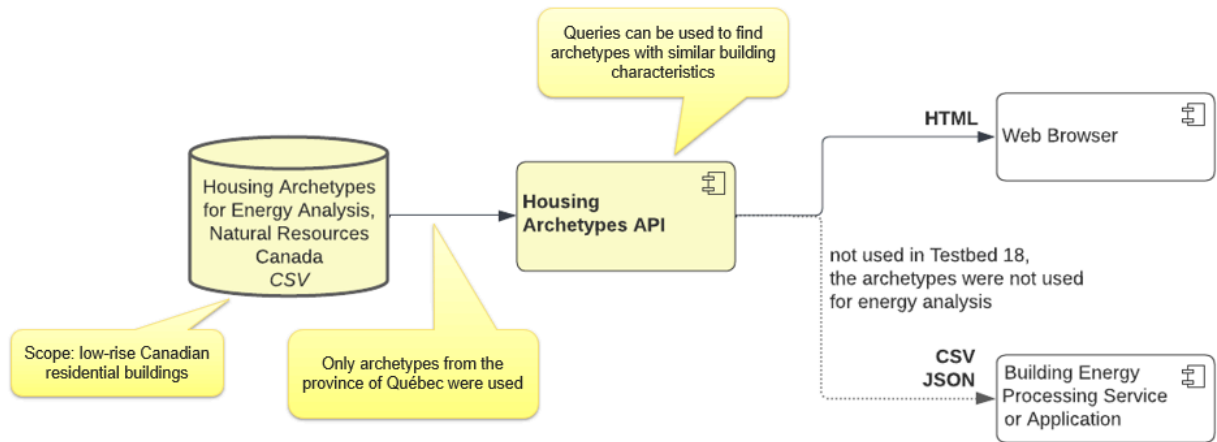


Figure 40 – The extended processing track workflow for accessing the Quebec Housing Archetypes.

The remainder of the section describes the resources and their representations accessible by the Quebec Housing Archetypes API.

7.6.2.1. Standard OGC Web API resources

The Quebec Housing Archetypes API provided the following resources:

- a [Landing Page](#) for the API that provides basic information about the API and links to the main sub-resources in the API;
- a [Conformance Declaration](#) that identifies the OGC Conformance Classes implemented by the API, identified through their URI; and
- an [API description](#) based on the OpenAPI 3.0 standard. The HTML representation uses the Swagger UI library.

7.6.2.2. Features

The relevant OGC API Standards and draft specifications for accessing the buildings as features were as follows.

- [OGC API – Features – Part 1: Core 1.0.1](#)
- [OGC API – Features – Part 3: Filtering, Editor's Draft](#)
- [Common Query Language \(CQL2\), Editor's Draft](#)

[GeoJSON](#) and CSV were supported as encodings for archetypes.

The Quebec Housing Archetypes API supported filtering archetypes using attribute names as query parameters (all values were combined with an implicit AND) or using filter expressions using [Common Query Language \(CQL2\)](#).

Sample requests:

- Archetypes for dwellings built in the 1990s ([HTML](#), [GeoJSON](#), [CSV](#))
- Archetypes for L-shaped dwellings with two storeys built in the 1990s ([HTML](#), [GeoJSON](#), [CSV](#))

7.6.2.3. Known issues

There were some challenges in authoring and using the API:

- The .h2k data file format and the HOT2000 data in general were not well documented at the time of Testbed-18. Having human readable titles and a definition for each attribute as well as information about the possible values (for attributes with enumerated values) or the value ranges and units (for numeric attributes) would be useful. This information was either not available (definitions; value ranges; enumerated values) or included in an Excel spreadsheet that was not consistently formatted (title, units). As a result, configuring the API so that clients could generate meaningful queries or that users could browse the data in a web browser was unnecessarily complex and time consuming. Value ranges and enumerated values were determined from the values in the existing archetypes.
- The sheer number of attributes was overwhelming and it was difficult for someone who was not an expert to decide which were the most important attributes and, in particular, which attributes were most valuable for filtering.
- If documentation were to be made available about which queries were most valuable for a user, these queries could be pre-configured in the API so that they would be easy to execute by users.



8

TESTBED EXPERIMENTS – DATA VISUALIZATION TRACK

TESTBED EXPERIMENTS – DATA VISUALIZATION TRACK

8.1. Description

The data visualization track documented the workflows consuming the outputs of the initial processing track and the extended processing track and displayed them through novel and innovative visualization tools.

8.2. Relevance to test bed goals

Client prototypes have applications in energy labeling of buildings on an individual basis and related metrics such as GHG emissions. The same simulation of energy related metrics for either a number of buildings in a geographic area or a number of buildings of the same type across a city or region could be accomplished with batch simulation processes. This capacity represents a linkage to program planning as in theory when optimization scenarios are run, backed with the right data including operating energy costs and capital costs of retrofits, the results could provide program managers with information about the most cost-effective retrofits for individual, archetypical, or geospatial groups of buildings. They could also produce summary statistics about building performance and labeling at any scale of geography useful for efficiency program planning by governments and utilities. This application was a specific aim for Testbed-18.

8.3. Data Visualization Track Summary

The following sections detail the participants' roles and interactions within this work track. Each section describes the activities that occurred during the testbed to complete the deployment of the client as well as any relevant comments, conclusions, and results. A description of the actions performed by the participants within this track follows in the table below.

Table 14

PARTICIPANTS (PARTICIPANT ROLE)	DESCRIPTION
---------------------------------------	-------------

The buildings in the Nuns' Island dataset were joined with the results of the energy simulation provided by the Building Energy Processing Service (yearly and monthly heating demand as well as the specific space heat demand). The information was visually presented as styled buildings rendered in Cesium JS and MapLibre GL JS. Cesium JS was used to render the feature data provided as 3D Tiles 1.1 and glTF 2.0. MapLibre GL JS was used to render feature data provided as Mapbox Vector Tiles or GeoJSON using the MapLibre Style Specification.

interactive instruments (Building Energy Client D124 and D125)

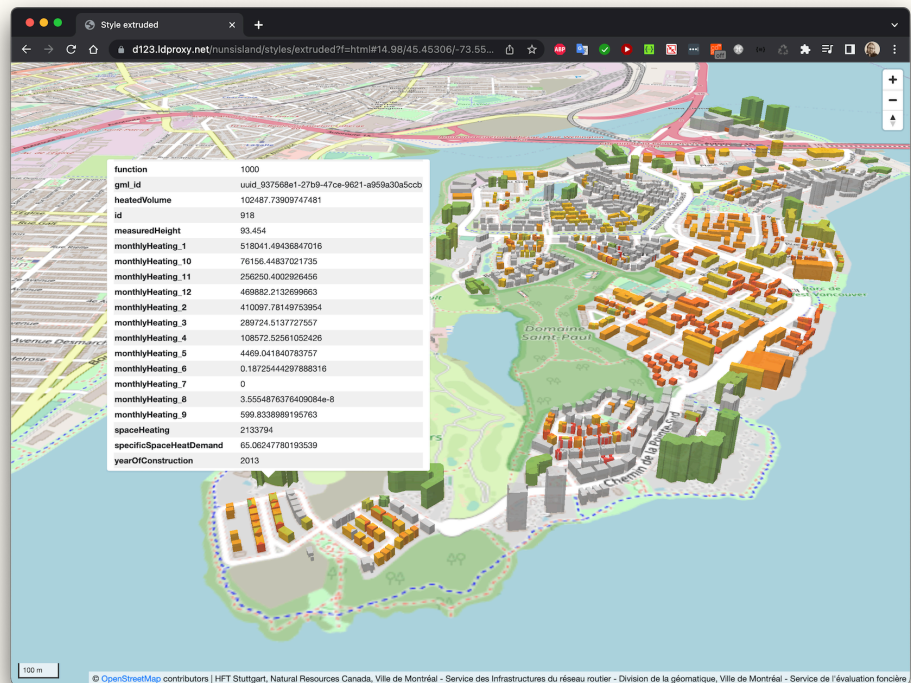


Figure 41 – Map with the buildings color-coded based on the specific space heating demand

Steinbeis Consortium (Building Energy Client D124 and D125)

The developed Building Energy Client (D125) web application enabled exploration of existing geospatial and building datasets available through the Building Energy Data Services for Nuns' Island. The web application provided a geospatial visualization of 3D GeoVolumes in 3D Tiles format on a 3D globe based on Cesium JS. Furthermore, the client web application supported starting a building energy simulation using the Building Energy Processing Service (D126). Building energy attributes selected by the user within the application interface were simulated on-the-fly by the processing service and the result was returned to the client. The client conflated the results with the selected geospatial dataset containing the building geometry and styling was then applied to the map visualization. Within the client application, clicking on an individual building displayed existing and simulated attributes for that building.

**PARTICIPANTS
(PARTICIPANT
ROLE)**

DESCRIPTION



Figure 42 – Result visualization in Building Energy Client D125 Steinbeis Consortium

Climative (Building Energy Client D124 and D125)

The goal for Testbed-18 client application development was to evaluate the capabilities of the Building Energy Data Service (D122) and External Geospatial Data Service (D127 and D128) to provide standardized access to both EnerGuide Rating system data and individual building characteristics. These results were not achieved within Testbed-18.

Properate (Building Energy Client D124 and D125)

The Client (D124) used the Building Energy Processing Service (D126) to acquire data. The input to the Client was a user defined geographic bounding box with four coordinates, entered through the client application interface. The client application then constructed a query for the Processing Service to perform an energy simulation for all buildings within the bounding box. The API response contained a list of buildings, each with a nested attribute of spaceHeating. The client application displayed the results to the user within the interface as an average of the spaceHeating attributes for all buildings contained within the selected bounding box.

GeoSolutions (Building Energy Client D124 and D125)

The client application developed by GeoSolutions is a frontend-only single page application based on the MapStore framework with custom components that focus on following aspects:

- exploration and visualization of 3D data listed inside an OGC feature API collection; and
- a style classification of features available in a collection mixing different content such as items, 3d tiles, and query-able data.

The client supported the inclusion of 3D Tiles data with a configured Catalog plugin provided out-of-the-box by MapStore. The Catalog plugin User Interface supported adding new services including the tile set JSON href that is the entry point of a 3D Tiles layer.

**PARTICIPANTS
(PARTICIPANT
ROLE)**

DESCRIPTION



Figure 43 – Style classification based on monthly energy demand property applied to the interactive instruments's Nun's Island collection visualized in 3D inside a MapStore custom application

Ethar Inc. (Building Energy Client D124 and D125)

The objective of the Ethar augmented reality (AR) client application was to color-code 3D building polygons based on energy-use data associated with individual buildings, ideally offering several time-domain datasets, filterable by building features (e.g., building height, year of construction, etc.). In keeping with OGC's mission, a key requirement for the project was to utilize or create open-source tools and code wherever possible. The following visualization modes were produced as part of the Testbed-18 work.

- A WebGL client implementation was produced to illustrate the Nuns' Island structures and allow manipulation of various energy parameters and render styling.
- Variables available through the drop-down in the upper right corner are listed below. As each variable option is selected, the 3D buildings on the map are rendered in the appropriate color:
 - Heated Volume
 - Monthly Heating
 - Space Heating
 - Specific Space Heat Demand
 - Year of Construction

PARTICIPANTS
(PARTICIPANT
ROLE)

DESCRIPTION

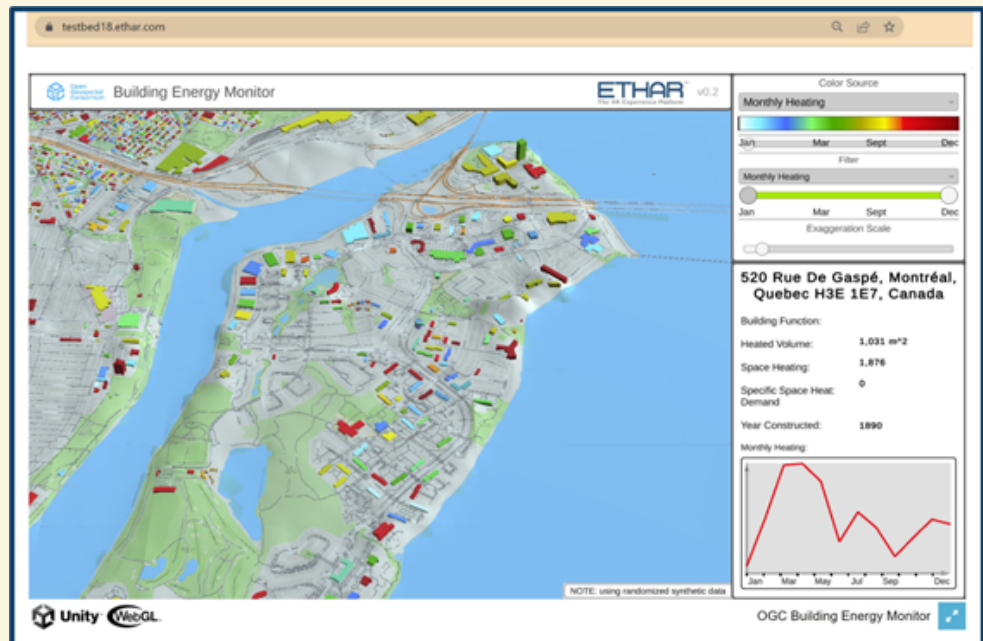


Figure 44 – Ethar WebGL Client Desktop Visualization

- The WebGL functionality from the desktop client was migrated to an AR device client application. The rendering was designed to spawn a 3D colorized map of terrain, natural and created features, basemap roads and waterways, and color-coded building models based on selected building energy variables. A 2D printed icon served as a “trackable” anchor for the rendering.
- The 3D AR data to be visualized was designed to be accessed using a handheld Android smartphone or tablet and a prototype apk was produced.

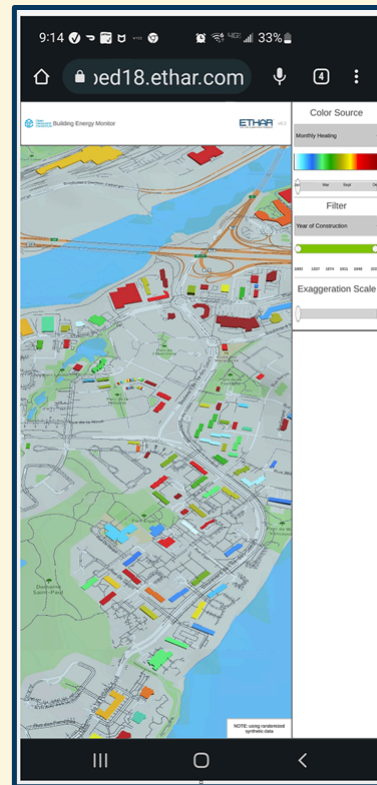


Figure 45 – Ethar WebGL Client Mobile Android Visualization

Because building energy data was not available via servers for the majority of the OGC T18 period of performance, the visualizations were mockups using simulated building data created by Ethar. While the applications were not mature enough to work with live data, the mockups demonstrated the value of using GeoAR visualization for building energy analysis.

8.4. Visualization in the Browser as part of OGC Web APIs (interactive instruments)

OGC API Standards typically recommend providing resources in an HTML representation so that a web browser can be used to explore, understand, and use the API resources. This eliminates the need to have a special client to explore the API and its resources.

For APIs that provide the data in a representation that is designed to support visualizing feature data, JavaScript libraries can be used to visualize that data in a web browser. If energy-related attributes are included in building feature data, the JavaScript libraries can also be used to style the data based on these attributes.

The Montreal dataset only has information about building function, building category, and the year of construction that can be used to group the buildings, for example by type and vintage.

NOTE: The building function and category attributes are from the Montreal Property Unit Assessment Dataset (UEF). The domain of values of the function attributes are available as a spreadsheet, the attributes in the UEF dataset is "CODE_UTILISATION". The category values are from the attribute "CATEGORIE_UEF" with the values "Condominium" and "Régulier".

The buildings in the Nuns' Island dataset have been joined with the results of the energy simulation provided by D125 (yearly and monthly heating demand as well as the specific space heat demand), so more information is available to be visually presented by Clients.

Idproxy, the software used to provide the APIs in the Building Energy Data Service D123, uses:

- Cesium JS to render feature data provided as 3D Tiles 1.1 and glTF 2.0; and
- MapLibre GL JS to render feature data provided as Mapbox Vector Tiles or GeoJSON using the MapLibre Style Specification.

The section in Track 2 on deliverable D123 includes several screenshots with buildings rendered in Cesium JS and MapLibre GL JS.

The following image shows a map using Mapbox Vector Tiles to style the buildings according to the specific space heating demand of the building that was generated by the building energy simulation. The coloring steps are based on the classification and colors in some German building energy certificates.

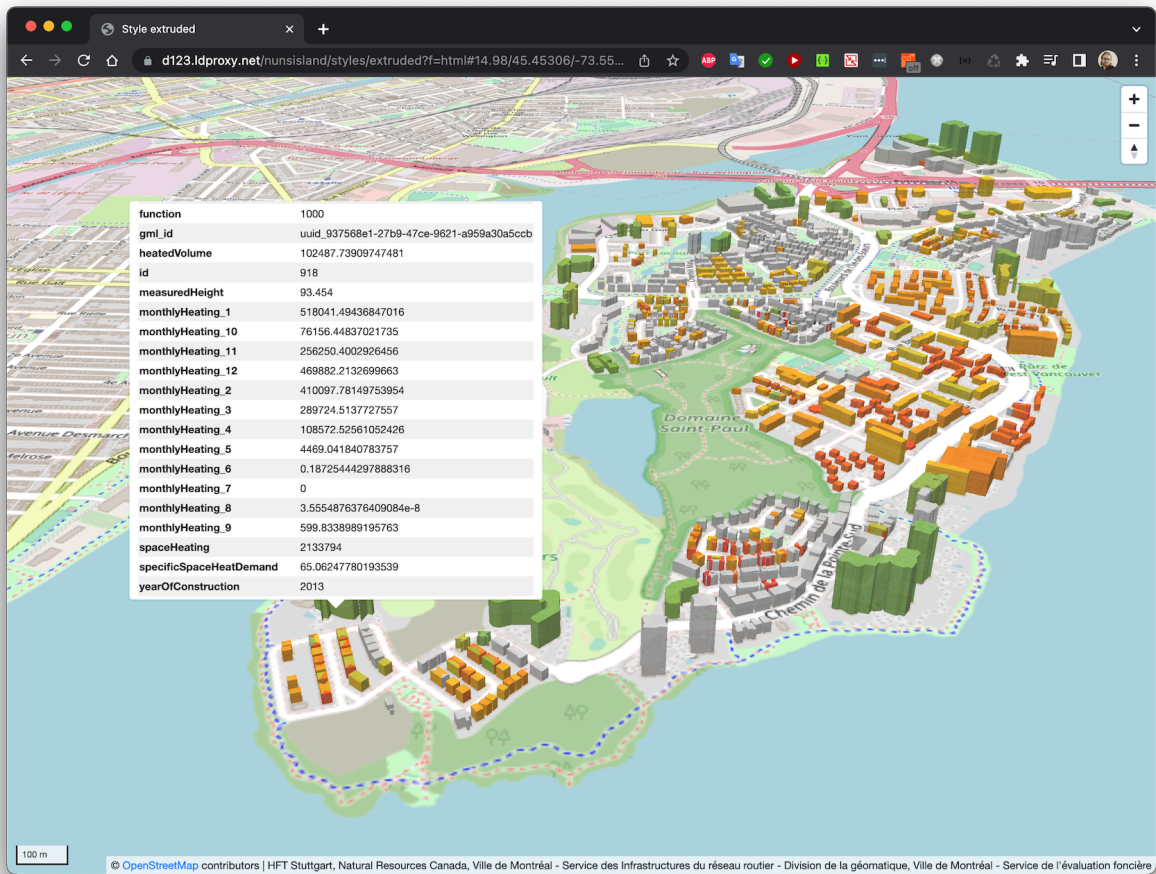


Figure 46 – Map with the buildings color-coded based on space heating demand

The style in the MapLibre style is shown below:

Example – The style

```
{
  "version" : 8,
  "name" : "Buildings",
  "center" : [ -73.55144, 45.45903 ],
  "zoom" : 14.0,
  "bearing" : 0.0,
  "pitch" : 60.0,
  "sources" : {
    "buildings" : {
      "type" : "vector",
      "tiles" : [ "https://d123.ldproxy.net/nunsisland/tiles/WebMercatorQuad/{z}/{y}/{x}?f=mvt" ],
      "scheme" : "xyz",
      "minzoom" : 11,
      "maxzoom" : 16,
      "attribution" : "HFT Stuttgart, Natural Resources Canada, Ville de Montreal - Service des Infrastructures du réseau routier - Division de la géomatique, Ville de Montreal - Service de l'évaluation foncière"
    },
    "basemap" : {
      "type" : "raster",
```

```

        "tiles" : [ "https://a.tile.openstreetmap.org/{z}/{x}/{y}.png", "https://
b.tile.openstreetmap.org/{z}/{x}/{y}.png", "https://c.tile.openstreetmap.org/
{z}/{x}/{y}.png" ],
        "scheme" : "xyz",
        "attribution" : "&#xa9; <a href=\"https://osm.org/copyright\">
OpenStreetMap</a> contributors"
    }
},
"glyphs" : "https://sg.geodatenzentrum.de/gdz_basemapde_vektor/fonts/
{fontstack}/{range}.pbf",
"layers" : [ {
    "id" : "Basemap",
    "type" : "raster",
    "source" : "basemap"
}, {
    "id" : "Buildings 3D",
    "type" : "fill-extrusion",
    "source" : "buildings",
    "source-layer" : "buildings",
    "layout" : {
        "visibility" : "visible"
    },
    "paint" : {
        "fill-extrusion-color" : [
            "interpolate",
            ["linear"],
            [
                "case",
                ["has", "specificSpaceHeatDemand"],
                ["get", "specificSpaceHeatDemand"],
                -1
            ],
            -1,
            "#ccc",
            1,
            "#71ad47",
            30,
            "#70ad47",
            50,
            "#92b613",
            75,
            "#71ad47",
            100,
            "#f5c001",
            130,
            "#f58c03",
            160,
            "#f25e04",
            200,
            "#fc5a08",
            250,
            "#f3241a"
        ],
        "fill-extrusion-height" : [ "get", "measuredHeight" ],
        "fill-extrusion-opacity" : 0.8
    }
} ]
}

```

8.5. Building Energy Client as 3D Web Application (Steinbeis Consortium)

To demonstrate the use of the draft OGC API – 3D GeoVolumes and OGC API – Processes in web environments, a 3D web application was developed to connect to Building Energy Data Services (D122 and D123) and Building Energy Processing Service (D126).

The developed Building Energy Client (D125) enables exploration of existing datasets available through the Building Energy Data Services. Next to the 3D GeoVolumes-Explorer (see GeoVolumes slide out on the left side of the figure below), the web application provides a geospatial visualization of 3D GeoVolumes in 3D Tiles format on a 3D globe based on Cesium JS (see the 3D map representation on the right side of the figure below).

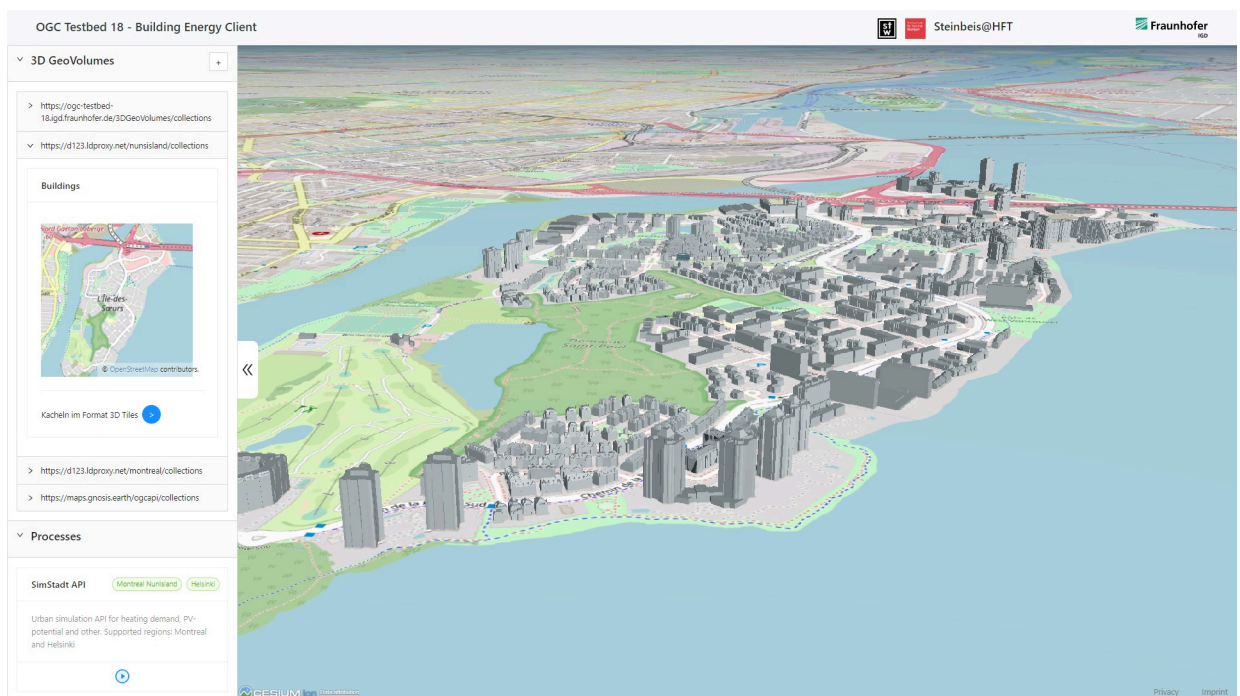


Figure 47 – Overview of Building Energy Client D125 Steinbeis Consortium

The 3DGeoVolumes-Explorer also supports adding new OGC API collections endpoints, which are then filtered by content type 'application/json+3dtiles.' Please note that the media type used for GeoVolumes using the 3D Tiles format is still under discussion. See the [GitHub repository](#) of the OGC API – 3D GeoVolumes candidate standard. Furthermore, the client supports starting a building energy simulation which is provided by the Building Energy Processing Service D126 and is described in the Initial Processing Track.

After selecting the building dataset to be used for a simulation, a simulation workflow can be started as follows.

- The simulation process configuration is done by selecting the area of interest and the building attributes which should be simulated and returned.

- The styling of buildings by energy-related attributes can be chosen by the user (figure below).

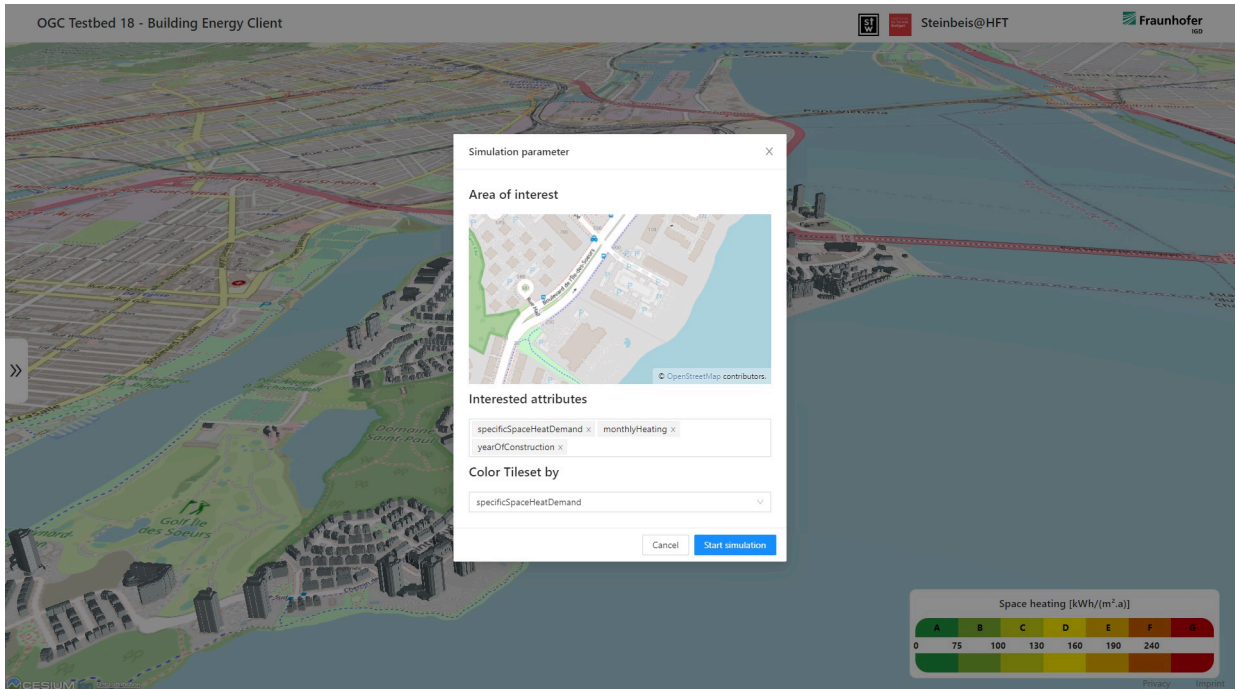


Figure 48 – Simulation configuration in Building Energy Client D125 Steinbeis Consortium.

- After submitting the simulation configuration, the corresponding HTTP post request is generated based on the selected user configuration (figure below).

Example

```
POST /ogc-api-processes/processes/simstadt_api/execution
HTTP/1.1
Host: steinbeis-3dps.eu
Accept: application/json, text/plain, */*
Content-Type: application/json
Content-Length: 270
```

```
{
  "inputs": {
    "bbox": {
      "bbox": [-73.5488, 45.4536, -73.5416, 45.4564]
    },
    "interestedAttributes": [
      "specificSpaceHeatDemand",
      "monthlyHeating",
      "yearOfConstruction"
    ]
  }
}
```

- Then the request is sent to the Building Energy Processing Service D126 and executed.

The previously selected building energy attributes (e.g., monthly heating demand and specific space heat demand) are then simulated by the processing service on-the-fly and the result is returned to the client. The client conflates the result with the selected geospatial dataset containing the building geometry and the styling is applied to the map visualization (figure below).



Figure 49 – Result visualization in Building Energy Client D125 Steinbeis Consortium

By clicking on an individual building, existing and simulated attributes for the building are shown. The Steinbeis building energy client is available at <https://ogc-testbed-18.igd.fraunhofer.de/>

In addition, to demonstrate the data interoperability of the Steinbeis Building Energy Client, connections were made to the Building Energy Data Services from D122, D123 and D127 (figure below).

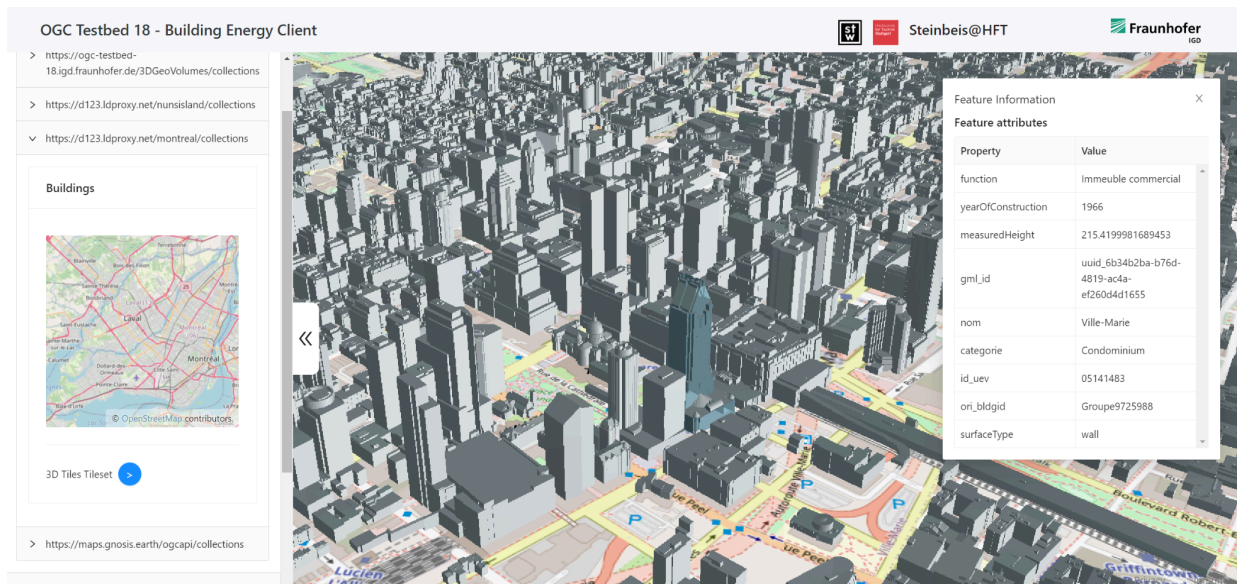


Figure 50 – Visualizing building energy data service D123 on the Steinbeis Consortium client.

The features presented by this client prototype have applications in energy labeling (and related metrics such as GHG emissions) of buildings on an individual basis. The same simulation of energy related metrics for either a number of buildings in a geographic area or a number of buildings of the same type across a city or region could be accomplished with batch simulation processes. This capacity represents a linkage to program planning, as in theory when optimization scenarios are run, backed with the right data including operating energy costs and capital costs of retrofits, the results could provide program managers with information about the most cost-effective retrofits for dwellings on an individual or archetypical basis, as well as groups of dwellings in selected areas of interest such as neighborhoods or whole cities. This application was a specific aim of Testbed-18.

8.6. Building energy client (Climative)

8.6.1. Primary Use Cases:

- access and evaluation of EnerGuide Ratings for a given region (Montreal Region) for machine learning model training and building energy simulations; and
- access and evaluation of individual building characteristics to provide inputs to machine learning models.

For the purpose of this project, the goal was to evaluate the Building Energy Data Service (D122) and External Geospatial Data Service (D127 and D128) capability to provide

standardized access to both EnerGuide Rating system data and individual building characteristics.

8.6.2. Data Models

The following data models were used.

- EnerGuide Rating System Database (Energide rating system (Inputs and Analysis))
- Building Attributes (Property Assessment Roll)

For Part 9 buildings (Single or semi-detached residential buildings three storeys or less) the minimum set of attributes for dwellings in the building stock, as relevant to Climative's work, are the following.

- Year of construction
- Building Type (Single-detached, Semi-detached, Row-House, Mobile Home, etc.)
- Building Location
- Building Footprint (GEOJSON) or (Ground Level Floor Area)
- Building Height or (Floors)

Tested Project Data Sources

- Dwellings (Clause 7.6.1.4)
- H2K Data ([data_htap])
- Housing Archetypes (Clause 7.6.2)

8.6.3. Approach:

The Testbed-18 activities revolved around the use of data standards and accessibility through a set of standardized APIs:

- evaluated API based data retrieval (from above referenced APIs); and
- evaluated HOT2000 TSV Data for Montreal Region (Not implemented via API).

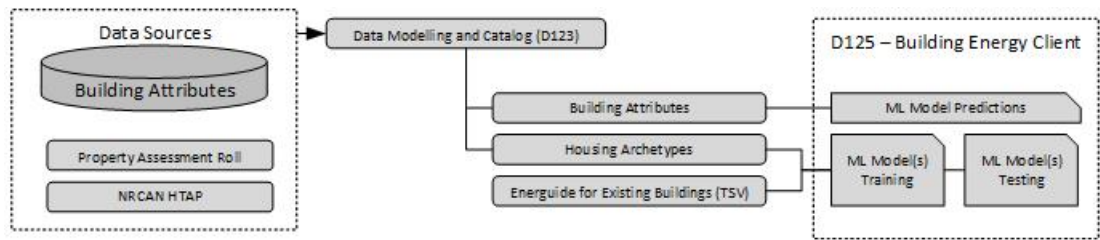


Figure 51 – Climative data processing workflow

Although Testbed-18 involved a number of data modeling approaches and the data was generally available from the regions under study, using the proposed data model endpoints (provided by other participants in Testbed-18) for Machine Learning training was difficult without reverting to the original data sets due to the way Climative ingests training data.

8.6.4. Model Training

The ERS database TSV data is available (for authorized users) directly from NRCAN in limited fashion (limited queries, etc.). Climative proposes that data access to this repository be improved through new initiatives with NRCAN that make access to data adhere to FAIR principles.

8.6.5. Building Attributes

The large-scale assessment and modeling of buildings requires a standardized set of data for buildings in any given region. Testbed-18 explored open data in the Montreal area. Many of the data processing services focused on Commercial Buildings (Part 3) whereas the Climative contribution to the project was primarily focused on Residential buildings (Part 9). Minimum data required for ML Based assessments of Part 9 building data is generally available via property tax assessment or open government data. For the proposed project this property tax assessment data was available as open source: (376k records).

8.7. Building energy client (Properate)

8.7.1. Primary Use-case

Get the average “Space Heating Demand” of buildings in a bounding box provided by the Client’s user. Industry and government use for the result may be in building benchmarking and regional comparison.

8.7.2. Approach

The Client (D124) uses the Building Energy Processing Service (D126) to acquire the necessary data. The input to the Client is a bounding box with four coordinates, which follows the [bbox.yaml](#) schema as specified in the [OGC API – Processes Standard](#). The client then constructs a query for the Processing Service and displays the results to the user.

8.7.3. Process

Once the Client receives the bounding box coordinate inputs from the user, it sends a POST request to https://steinbeis-3dps.eu/ogc-api-processes/processes/simstadt_api/execution with a JSON-formatted body containing a bounding box attribute (the user coordinates) and “interestedAttributes”: [“spaceHeating”].

The Client asynchronously waits for the server’s response. The response contains a list of buildings, each with a nested property of spaceHeating. To get the average across all buildings, a simple operation is necessary.

Where:

- answer is the body of the response from the server, parsed from JSON
- average is a function that takes in an array of numbers to return their average.

Example

```
average(
  answer.buildings
    .map( building => /* iterate through the buildings */
      building.parts.reduce( (sum, part) =>
        sum + part.attributes.spaceHeating, 0 /* sum the Heat Demand
of all parts */
      )
    )
  .filter( spaceHeating => spaceHeating) /* to filter out NaN and 0 */
)
```

Example code to determine average spaceHeating demand

The Client then displays the average spaceHeating demand to the user.

8.7.4. Challenges

There were a number of challenges for smooth operation of this Client.

- By design, the client receives all the building data from the Building Energy Processing Service and then proceeds to generate the average. Doing the data processing on the

client side requires high rates of data transfer which will be hard to scale due to service and infrastructure costs.

- The Building Energy Processing Service may take a long time to respond to the Client's query. In case of any connection disruption during the "simstadt_api" process execution, the results may be hard to recover. While the Building Energy Processing Service provides a list of all jobs, programmatically finding the relevant job appears to be challenging.
- As it can be seen in the code example, a filter function was necessary because some buildings had a missing spaceHeating value.

8.8. Building energy client as 3D web app map viewer D128 (GeoSolutions)

The client application developed by GeoSolutions is a frontend-only single page application based on the MapStore framework with custom components that fill focus on following aspects:

- exploration and visualization of 3D data listed inside an OGC feature API collection; and
- style classification of features available in a collection mixing different content such as items, 3D Tiles and queryables data.

MapStore is a highly modular Open Source WebGIS framework based on Reactjs and JavaScript technologies and Cesium, OpenLayers, and Leaflet as mapping libraries.

Source code and demo links for the application:

- [Source code of MapStore building energy client app](#)
- [Demo of Nuns' Island collection using interactive instruments endpoint](#)
- [Demo of City of Montreal collection using Ecere endpoint](#)

====Exploration and visualization of 3D data

The client supports including 3D Tiles data with a configured Catalog plugin provided out-of-the-box by MapStore. The Catalog plugin User Interface supports adding new services including the tile set JSON href that is the entry point of a 3D Tiles layer. By analyzing the content of each OGC feature API collection provided by other participants it is possible to get a direct link to a tile set entry point. After some tests the client was able to render the content of all 3D Tiles with some limitations. Below are considerations and feedback about this experiment.

- The tested endpoints have different ways of defining the real property in case of 3D tiles (eg. <http://www.opengis.net/def/rel/ogc/0.0/tileset-3dtiles> or <http://www.opengis.net/def/rel/ogc/1.0/bvh>). The client needs a consistent way to detect these kinds of links.

- The optional field properties that describe the attributes available in a feature are missing in some of the 3D tile tile set json entry points tested. Even if optional, this information could be useful to create style or filter content of a tile set without waiting for the download of an actual feature. The query-able link could be used as fallback in an OGC API – Features collection context but the information provided in a query-able response could be different from the actual content of a 3D Tile feature (e.g., less properties available).
- Some large datasets such as Montreal have performance issues while loading and navigating in a web client. The reason seems related to the tiling structure and the LoD of the features. In the case of energy data visualization it is likely possible to reduce the detail of the building in particular when the camera is far away from the features.
- Some tested 3D Tiles included textures that in some cases slowed down the web application. It may be possible to exclude the textures from a feature if the aim is energy data visualization where the style color represents energy demand information.

Tested endpoints:

- <https://d123.ldproxy.net/montreal/collections/buildings/3dtiles>
- <https://d123.ldproxy.net/nunsisland/collections/buildings/3dtiles>
- <https://maps.gnosis.earth/ogcapi/collections/SanDiegoCDB:CoronadoBridge/3DTiles/tileset.json>
- <https://maps.gnosis.earth/ogcapi/collections/SanDiegoCDB:Buildings/3DTiles/tileset.json>
- <https://maps.gnosis.earth/ogcapi/collections/SanDiegoCDB:Trees/3DTiles/tileset.json>
- <https://maps.gnosis.earth/ogcapi/collections/montreal/3DTiles/tileset.json>
- <https://ogc-testbed-18.igd.fraunhofer.de/3DGeoVolumes/collections/Montreal/3dtiles/tileset.json>
- <https://ogc-testbed-18.igd.fraunhofer.de/3DGeoVolumes/collections/Helsinki/3dtiles/tileset.json>

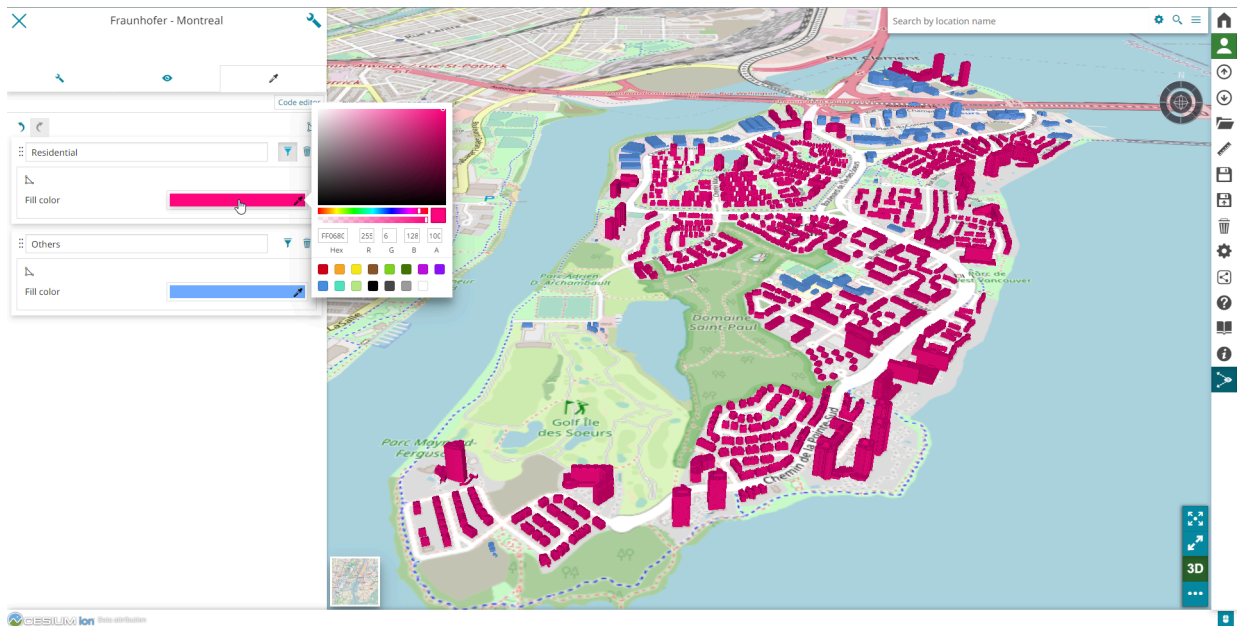


Figure 52 – Montreal 3D Tiles layer retrieved from an OGC API features endpoint rendered inside a MapStore WebGIS client

8.8.1. Style classification of features

This second experiment demonstrated the use of OGC API – Features to create a classified style within a client by accessing the collection entry point. The workflow steps are as follows.

- Read the collection query parameter from the application URL that contain the href for an OGC API Features collection.
- Send a request to the collection href and retrieve all available links.
- Extract the GeoJSON items, 3D Tiles and queryables links to use them as follows.
 - GeoJSON items: to compute the classification within the client using Jenks optimization method, also called the Jenks natural breaks classification method. This is a data clustering method designed to determine the best arrangement of values into different classes for visualization purposes.
 - 3D Tiles: to visualize the volume of the features.
 - Queryables: to get all available properties inside the collection.
- Create two layers in the map based on the response of the collection: one layer for the 3D Tiles and another for the GeoJSON features.

Once all the information and layers were ready, the client rendered a style classification panel that enabled the user to select one of the numerical properties and a color ramp to generate a classified style. The endpoint provided by an interactive instrument's API related to Nuns Island

contained an array property with nested information about energy demands. As such, the client was improved to support this type of property. Each feature in the GeoJSON items collection has the energyDemands property with this structure:

```
"energyDemands": [
  {
    "endUse": "spaceHeating",
    "energyAmount": {
      "variableProperties": {
        "thematicDescription": "Space Heat Demand",
        "acquisitionMethod": "Simulation",
        "interpolationType": "SucceedingTotal",
        "source": "SimStadt"
      },
      "temporalExtent": {
        "begin": "2022-01-01",
        "end": "2022-12-31"
      },
      "timeInterval": {
        "value": 1,
        "unit": "year"
      },
      "values": [
        ...
      ],
      "unit": "kWh"
    }
  }
]
```

Figure 53 – The property structure of energyDemands

If the energyDemands array property is selected, the client displays additional fields to compute the classification based on the energy demand option selected.

The feedback and considerations based on this experiment are given below.

- energyDemands nested properties in a GeoJSON feature collection could be difficult to manage within a client for styling or filtering. In this particular case, the energyDemands property object structure is repeated for each feature in the collection. This energyDemands property structure could be maintained outside of the GeoJSON collection and kept only inside the GeoJSON structure if there was a reference to the energy demand property inside the feature. An approach like this could reduce the size of the GeoJSON items and simplify access to their properties.
- The implementation developed in this experiment creates the style class breakpoints within the client. To work effectively, all collection items need to be ready and downloaded into the client. This approach works well only with a small set of features as shown for the Nuns' island scenario. It would be beneficial if an OGC API features implementation exposed statistical information about the queryable properties (e.g., minimum and maximum values) that could help to set UI components within the client. Another beneficial addition for large datasets could be a process endpoint that would generate classification breakpoints given intervals and methods to move the computation to a server.

Tested endpoints:

- <https://d123.lproxy.net/nunsisland/collections/buildings?f=json>
- <https://maps.gnosis.earth/ogcapi/collections/montreal?f=json>



Figure 54 – Style classification based on monthly energy demand property applied to the interactive instruments's Nun's Island collection visualized in 3D inside a MapStore custom application

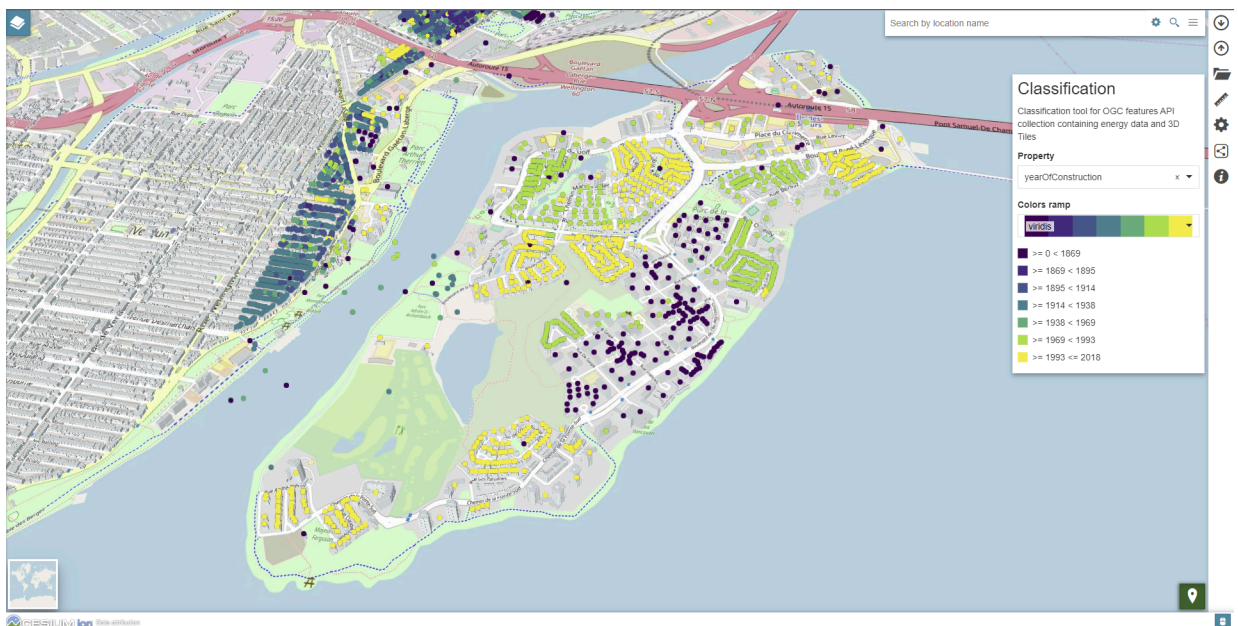


Figure 55 – Style classification based on monthly energy demand property applied to the interactive instruments' Nuns' Island collection visualized in 3D inside a MapStore custom application

8.9. Unity-Based Augmented Reality Visualization (Ethar Inc.)

8.9.1. Contribution Summary

Under the auspices of the Open Geospatial Consortium Testbed 18 project, a team led by Ethar, Inc. (Huntsville, AL) conducted analysis and implementation of three-dimensional (3D) and Augmented Reality (AR) visualizations for geospatial data, in particular, building energy data. The team included Ethar Inc., the Open AR Cloud association, Tech Maven Geospatial, and Quiet Pixel. Findings include the following.

- The use of 3D/AR for viewing building energy data is a promising method for analysis of the data supporting recommendations and actions for building energy management.
- ETHAR studied several 3D/AR approaches to creating and viewing building data and implemented a prototype 3D desktop client application and a tabletop AR client viewer. Mock-ups of simulated building energy data are provided for each viewing mode.
- A standards-based architecture for AR from the European Telecommunications Standards Institute (ETSI) is extended for geospatial AR, indicating where OGC standards are applicable.
- An evaluation of OGC Standards relative to geospatial AR indicates needed changes to the specifications, in particular to accommodate 3D Tile and 3D Tile Next datasets.
- An evaluation of existing AR software for geospatial AR indicates that the AR software ecosystem is still in development and the adoption of geospatial standards by game platform developers could increase the compatibility of existing code with geospatial data sets.
- The implementation of geospatial data in 3D/AR for Testbed 18 was limited for two reasons: AR software is not currently compatible with geospatial data formats; and building energy data was not available through Testbed 18 APIs early enough in the project to be fully utilized.
- Demonstrations of a 3D desktop app, and of a “tabletop” AR app for Android mobile devices, were created to visualize simulated building energy data, and to assess the compelling value of these unique viewing modes.

8.9.2. Ethar T18 Objectives – The value of GeoAR for building energy analysis

Many professionals in utilities, resource management, urban planning, construction, and other energy-related sectors are becoming familiar with the benefits of visualizing geospatially-located data via three-dimensional (3D) and Augmented Reality (AR) technologies. AR tools

are a powerful new user interface into Digital Twins developed to mirror real-world assets, infrastructure, and scenes. AR visualization of digital elevation and other geo-located data by field workers can inform mission-critical decisions and reduce risks. Geospatial stakeholders using AR-enabled applications can engage with rich graphical information both remotely using VR and 'tabletop' AR, and onsite in the real world through AR. AR applications are valuable in urban, suburban, and rural settings, and can be applied across the entire product and service lifecycle, from design and engineering, through daily operations and maintenance.

For Testbed 18 (T18), Ethar, Inc. assembled a cadre of in-house and contractor specialists in geospatial software systems, Unity™ game engine applications, and AR development. They teamed with representatives from the Open AR Cloud Association (OARC) [together, the Ethar Team]. The partnership was charged with creating one of several Client Application deliverables (D124, D125) that accessed data provided by other T18 team Energy and Geospatial Data Services (D122, D123, D127, D128) and the Building Energy Processing Services (D126). Two approaches were to be explored.

- The Processing Service Provider integrates building energy data, and 3D terrain, map, and building model data, from various sources prior to delivery to the Clients through a new OGC Building Energy Web API. Clients then enable visualization and interaction for the end-user.
- Data is integrated on the Client side after retrieval from the Processing Service via the Web API, and/or from Building Energy and external Geospatial Data Services. Clients then enable visualization and interaction for the end-user.

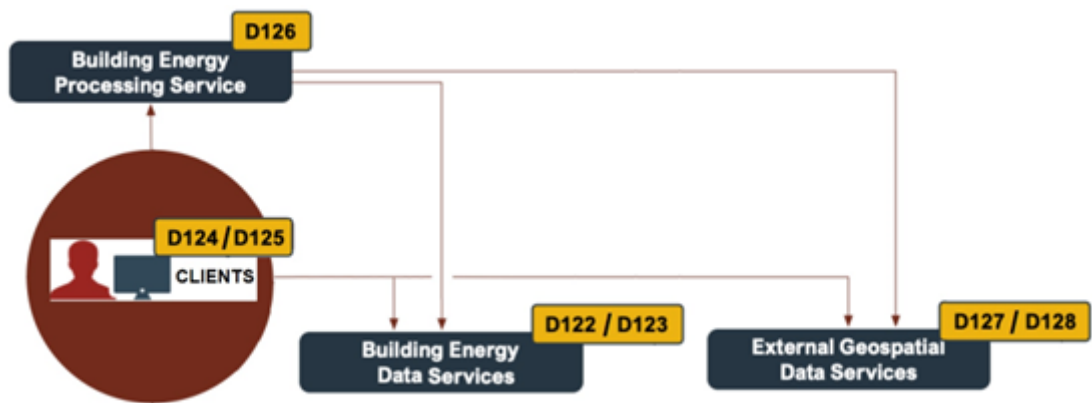


Figure 56 – Testbed 18 Building Energy Deliverable Relationships (Credit: OGC)

A total of four different organizations created Client application deliverables for T18 (D124 / D125). Three teams created conventional desktop clients. The Ethar team proposed a unique Augmented Reality and Virtual Reality (AR/VR) Client approach wherein building energy and geospatial data would be viewable in three modes:

1. remotely via 3D desktop and VR visualization;
2. remotely via AR “tabletop” visualization; and
3. onsite (“in-situ”) life-size visualization via AR at full-scale in the real-world.

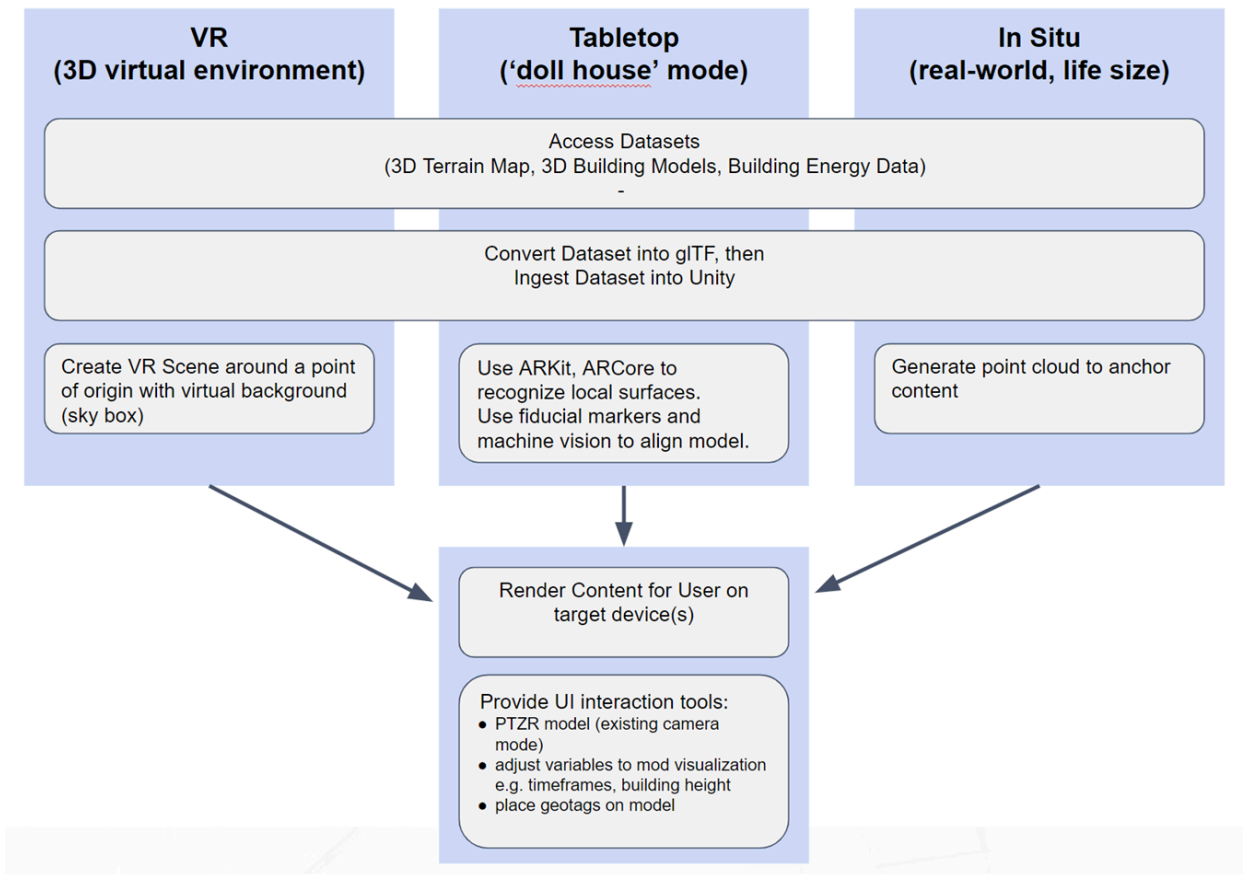


Figure 57 – AR Viewing Mode Process Options (Credit: Ethar)

Where practical, the Ethar Client was also intended to incorporate use of the OGC Geopose standard. These AR visualization elements would then share a common, real-time, graphical representation of data with the other T18 participants for use by field personnel onsite and remotely. The primary deliverable was to be a prototype demonstration including implementation guidelines and sample code, and contributions to the Testbed findings, best practices, and recommendations to inform potential new OGC specifications and standards development.

8.9.3. Ethar section overview

This section describes analysis and implementation of three-dimensional (3D) and augmented reality (AR) to geospatial data, in particular Building Energy Data. The use of 3D/AR visualization for Building Energy Data is a promising method for analysis of the data leading to recommendations and actions for building energy management.

This section begins with a comparison of an architecture standard for AR with OGC geospatial standards. Opportunities to implement OGC standards in the ETSI architecture are indicated later in the section, as well as findings that some features are lacking in OGC specs for use with AR. The section describes the concept and potential benefits of applying AR to geospatial data of building energy.

The section also provides a survey of various AR implementations attempted by the Ethar team, and the needed development to make them compatible with OGC Standards. The 3D/AR client application approaches attempted met with limited success for several reasons, among them: the lack of reasonably complete 3DTiles/Unity plugins from which to begin Ethar’s development; the lack of available test datasets and defined parameters; delayed definition of API endpoints; unavailable or unreliable data access; gaps in datasets; and unique peculiarities in and limitations of using the Testbed data services.

Finally, recommendations and next steps are offered for consideration at the end of the section.

8.9.4. General AR Solution Architecture

The Ethar team pursued a contagion of various architectures and tools to develop the AR/VR Client application. Dead-ends and impracticalities were found to be abundant. To begin the project, the baseline requirements were established and initial research and assessments were conducted.

The architecture of a general AR/VR solution is unique relative to conventional desktop GIS architectures due to the peculiarities of the AR/VR hardware, software, and user interaction elements.

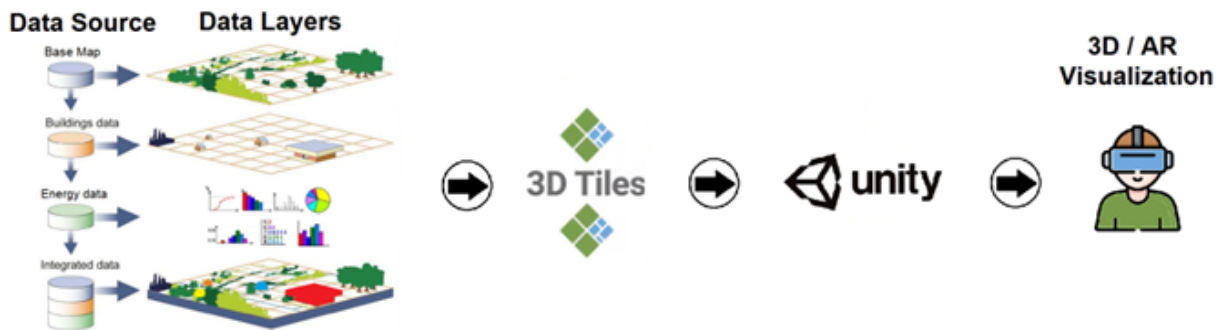


Figure 58 – GeoAR Conceptual Rendering Process (Credit: Ethar)

The figure below shows the application of OGC Standards to areas of applicability with respect to the general AR solution architecture defined by the ETSI AR Framework standard.

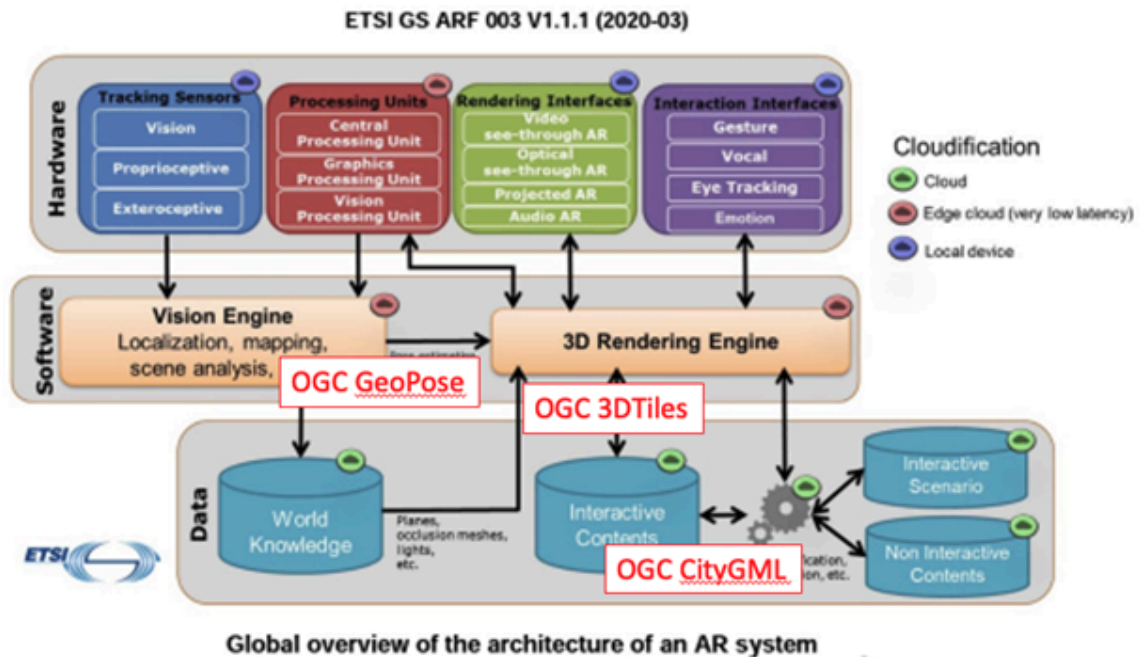


Figure 59 – ETSI AR Framework incorporating OGC Standards (Credit: ETSI)

8.9.5. OGC standards and AR

Multiple challenges were revealed when trying to integrate OGC standards into the AR client architecture.

8.9.5.1. Findings of OGC standards readiness for AR for buildings

OGC standards were developed for use with geospatial data. AR data has certain localized geospatial characteristics, but the more popular development environments come from the gaming industry (e.g., Unity and Unreal). The task of integrating large-scale geospatial base-maps and 3D building and natural features into game engines for dynamic generation of AR renderings is not trivial and it is challenging to link OGC-compatible datasets and tools into the AR experience creation process.

8.9.5.2. Working with Tiles, generally

Both the gaming and geospatial visualization approaches make use of tiles. But there are incompatibilities of gaming tiles extended to geo and geo tiles extended to AR. A key difference is Coordinate Reference systems. Among other differences, game platform geo elements are primarily web Mercator projections, which do not work well with 3DTiles.

The team needed to find a way to make requests for tiles. Could a Scene help determine a BBOX/Bounds and potentially zoom level? Could the system query what the unity-xyz is

rendering from Zoom_Level, Tile_Column, Tile_Row to then request that from Mapsui? The solution also required an in-app API. How could data be configured to make it flexible enough to support a configuration file (YML, JSON,XML, etc.)? Other questions considered included how to execute feature info type requests for pop ups and how to query attributes.

URL's established for testing/integration (MapTileServer.xyz is Tech Maven Geospatial's Tile Server):

- <https://s3.amazonaws.com/elevation-tiles-prod/terrarium/{z}/{x}/{y}.png>
- <https://s3.amazonaws.com/elevation-tiles-prod/normal/{z}/{x}/{y}.png>
- http://maptileserver.xyz:8000/services/osm_vectortiles
- http://maptileserver.xyz:8000/services/osm_vectortiles/tiles/{z}/{x}/{y}.pbf
- https://techmavengeo.cloud/vectortile_stylesheets/openmaptiles/klokantech-3D/style.json
- https://techmavengeo.cloud/vectortile_stylesheets/openmaptiles/klokantech-3D/3d_style_with_HERE_satellite.json
- <http://maptileserver.xyz:8000/services/satellite/tiles/{z}/{x}/{y}.jpg>
- http://server.arcgisonline.com/ArcGIS/rest/services/ESRI_StreetMap_World_2D/MapServer/tile/{z}/{y}/{x}.jpg
- https://techmavengeo.cloud/vectortile_stylesheets/reference_layers/geonames_vectortiles.txt this text file lists 9 point layers
- http://maptileserver.xyz:8000/services/reference_layers/worldwide_gadm/tiles/{z}/{x}/{y}.pbf
- http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/tile/{z}/{y}/{x}.png
- <http://tile.stamen.com/terrain/{z}/{x}/{y}.jpg>
- https://techmavengeo.cloud/demo_data_tileserver/montreal/tileserver.php?/index.json?/montreal_buildings_from_citygml/{z}/{x}/{y}.pbf
- https://techmavengeo.cloud/demo_data_tileserver/montreal/tileserver.php?/montreal_buildings_from_citygml.json

Download source files included:

- https://techmavengeo.cloud/test/ethar/montreal_buildings/montreal_buildings_from_citygml.mbtils
- https://techmavengeo.cloud/test/ethar/montreal_buildings/montreal_buildings_from_citygml.gpkg
- https://techmavengeo.cloud/test/ethar/montreal_buildings/montreal_osm_bldgs.gpkg

Upon encountering challenges with some of the above datasets and approaches it was evaluated whether it made sense to pivot and use either Mapbox Unity or ESRI Unity. This will be discussed later in this document.

- <https://docs.mapbox.com/unity/maps/guides/styling/>
- <https://blog.mapbox.com/styling-3d-buildings-in-unity-cecbe156f898>
- <https://www.nuget.org/packages/mapbox-vector-tile>
- <https://www.nuget.org/packages/Mapbox.VectorTile/>

Or if we chose not to read vector tiles:

- <https://github.com/reymtm/tilequery>
- <https://github.com/stevage/QueryRemoteTiles>

We also investigated other libraries to understand how they might be utilized with Unity, and information related to Tiles.

- <https://github.com/timokorkalainen/Unity-GeoJSONObject>
- <https://api.mtr.pub/ElmarJ/GeoJsonCityBuilder>
- <https://github.com/Geodan/UnityViewer>
- <https://github.com/aiunderstand/B3dmViewerUnity>
- <https://github.com/bertt/b3dm.tooling>
- <https://github.com/bbbbx/3d-tile-content-inspector>
- <https://github.com/openspacing/tilefile-info-viewer>
- <https://docs.mapbox.com/mapbox-unity-sdk/api/unity/Mapbox.Unity.Telemetry.html>

OGC API features – use GET REQUEST

- <https://app.swaggerhub.com/apis/OGC/ogcapi-features-1-example-1/1.0.1>

8.9.5.3. OGC 3D Tiles for AR

3D Tiles is an open standard built on glTF and various 3D data types. It is used for working with massive, heterogeneous 3D geospatial datasets such as point clouds, buildings, photogrammetry, and vector data. Whereas standards like glTF compress and optimize 3D assets for runtime efficiency and sharing, 3D Tiles takes that to the global scale by creating a spatial index of 3D content which can quickly deliver large metropolitan datasets, such as for Montreal, Québec. The specification enables renderings with prioritized levels of detail: streets and nearby buildings are presented in high resolution, and buildings and landscapes in the distance are delivered with

lower resolution. 3D Tiles Next is the latest iteration of the standard, providing improved 3D analytics, efficient queries of 3D data and improved support for contextual data.

The team studied the details and implications of using 3D Tiles. This included becoming familiar with the tile format header structure and the format of the actual payload data and looking at the structure of batched #D models and spending time understanding the declarative styling possibilities. It was a challenge to coordinate the vertical elevation of disparate datasets. 3D Tiles are in EPSG:4979 (which is WGS84 datum) and includes height above the ellipsoid, whereas 4326 does not. https://epsg.org/crs_4979/WGS-84.html <https://epsg.io/4979>.

Finally, the team discussed how 3D Tiles defines the z-axis as up for local Cartesian coordinate systems. In contrast, glTF – the target model format for manipulation in Unity – considers the y-axis as up. Transformations had to be considered for transforming the glTF asset at runtime so the z-axis points upwards. Additionally, glTF assets have a unique node hierarchy with transforms. <https://github.com/CesiumGS/3d-tiles/tree/main/specification#transforms>.

While working with 3D Tiles, the team found inconsistencies in documented vs. practical functionality and additional complications due to different possible implementations of the standard.

Some Tools for 3DTiles C# .NET

- <https://github.com/bertt/b3dm.tooling>
- <https://github.com/bertt/b3dm-tile-cs> <https://github.com/bertt/3dtiles>
- Downloading https://github.com/bertt/3dtiles_downloader

View Quantized Mesh Terrain Tiles in Unity

- <https://github.com/jhkluiver/UnityQuantizedMeshViewer>
- <https://github.com/mvangog/QuantizedMeshTiles>

8.9.5.4. OGC CityGML for building energy

Geodata challenges faced the team from several perspectives. The enormous number of geospatial data and modeling formats, versions, and extensions meant that it is impossible in the near term to support all variations. The challenge due to non-existent or incomplete data standards is obvious to OGC members. Tools are not readily available to easily integrate data. Field names, formats, and usage in building energy datasets are not aligned. Equally challenging is that within given building energy datasets, entries are not complete in most cases. For example, energy usage data may only be available for one out of every ten buildings in a given area of interest.

Another example of dataset incompatibilities is the lack of a standardized format for building addresses. Inconsistent notation of street names, building and unit numbers, city, region, country, mailing codes, and more. In some instances, building locations are only provided with latitude and longitude approximations. There is no agreed-upon origin point for buildings, so reference may be to centroids or prominent cardinal direction corners. In many cases, the tools

developed for working with these disparate datasets seem to be kludged together and are not at all easy to use or combine for a given intended result.

For this OGC project, the guidance was to focus on using the 3D Tiles Community Standard for working with 3D terrain and building model data. Testbed service providers created datasets that were disparate and had nuanced differences that created challenges. For example, some sets did not use glTF for 3D assets at the base tile level. Some datasets utilized a unique extension that caused compatibility issues. 3D terrain data was not included in the 3D Tiles datasets the team reviewed. This varied internal structure of the data meant that each version the Ethar team attempted to consume had to be unique.

8.9.5.5. Geospatial Metadata Standards

Currently, the lack of geospatial metadata standards was found to be a barrier to easy adoption. Different data tiles formats, different bounding boxes referencing, limited dataset (geographically), building attribute datasets lacking metadata with definitions (min/max, range, etc.). The unit of measure must also be provided for all variables (e.g., the time domains for provided parameters were not clear and units could have been provided as metadata).

8.9.5.6. Property Identifier Standards

The 3D Tile datasets reviewed by interactive instruments (D123) were JSON files with text descriptions with no 3D building geometry, no 3D terrain elevation data, no road maps, water, or other geologic features. It is high-level metadata that indicates boundaries where the system should then look for that content data. At the bottom of the text file, instructions are provided to load a separate 3D building file from a second dataset associated with a given building ID. The text field can also include data files that describe the name or address of the buildings. There is only a lat/long box describing the location of any given asset (a building ID, 2D property outline, 3D building, a one-dimensional point of interest, etc.).

The 3D Tiles JSON file will indicate where to find and download a given 3DBM binary file when the team tries to begin loading geo assets. The 3DBM has four possible sections in it, and in each, data can be stored in the following different ways.

- The data can contain another section of JSON text providing more information about what is being loaded (building ID, civic address, etc.).
- The data can contain a glTF reference and/or asset definition attached at the end of the binary file. The actual 3D asset model definition may be in the 3DBM file or may be in an attached binary file.
- The glTF file also allows the use of extensions. These are csv text strings of the formats used when the asset was created (e.g., Cesium with an .rtc extension). This aspect of the Cesium standard provides instructions for how a file should be treated with multiple options.

This is where another challenge lies. New AR client applications must support all the flexibility that went into the top-level text file. The app must also support the five or six different ways

the 3DBM file can be structured and the different ways the glTF data can be structured. These must all be managed correctly to get the right data showing up in the right way at the right geolocation and with the proper orientation. This degree of complexity eventually breaks all of the existing plug-ins and open source software options the team explored because none of them were designed to work with so many possible edge designs and data formats.

Even after dealing with all of the above, the results provided the ability to put a square box on a map that does not include energy, 3D terrain, or basemap data, which must come from other sources. For AR, the energy data, 3D terrain, and basemap must be aligned perfectly from a lat/long perspective to achieve a viable rendering and place the visualization in the real-world setting of the viewer.

8.9.6. Applying Geo-AR to building energy analysis

Methods for creating dynamic augmented reality views of geospatial and building energy data were explored.

8.9.6.1. AR/VR visualization of building energy data

Recently, game engines are becoming increasingly integrated into GIS, digital twins, and industrial metaverse tools. As the transformation to spatial computing continues it will be important to explore ways to improve how 3D Tiles can bring massive scale to USD models, improve support for more game engines, and unlock new 3D AI capabilities.

The Ethar objective was to color-code 3D building polygons based on energy-use data associated with individual addresses, ideally offering several time-domain datasets filterable by building features (e.g., building height, year of construction, etc.). In keeping with OGC's mission, a key requirement for the project was to utilize or create open-source tools and code wherever possible.

To provide practical boundaries to the application scope, several additional guidelines were established as follows.

- Utilize the OGC 3D Tiles community standard for streaming data, if possible.
- Focus on only one suburban area of one metropolitan region initially. Specifically, the dataset selected was for Nuns' Island, an island located in the Saint Lawrence River within the city of Montreal, Quebec.
- Initially, visualize the 3D dataset using only one AR visualization mode. The team selected a tabletop Augmented Reality approach, triggered via image-tracking of a selected fiducial marker (an object or shape placed as a reference point for tracking devices, such as a smartphone camera). The marker could be printed out and placed on any table or horizontal surface as an origin point for placement of the digital 3D model.
- Initially, create only the fundamental pan / tilt / zoom / rotate (PTZR) interactive functionality for the user.

- Create only one styling methodology for the building energy data, to be applied to the extruded 2D building profiles, color coding of each structure based on its tier within a small subset of the available energy parameters (monthly heating [energy usage], heated volume, year of construction, etc.).
- Deploy the application for only one software / hardware platform: Android Phone .apk
- Cache the dataset locally at first, then consider a REST architecture only after the OGC API teams deliver fully-defined endpoints.

Several extended scope factors were considered “nice to have”:

- modify Dataset visualization on the fly (e.g., filter by this feature of the dataset);
- visualize additional datasets layered onto the scene;
- deploy for additional hardware (Android Tablet, iOS devices); and
- visualize from additional perspectives (VR bird’s eye isometric, on-site real-world scale).

3D City Models of Montreal: <https://donnees.montreal.ca/ville-de-montreal/batiment-3d-2016-maquette-citygml-lod2-avec-textures2>

Additional datasets related to buildings/lots (e.g., building usage, total floors, year of construction, etc.): <https://donnees.montreal.ca/ville-de-montreal/unites-evaluation-fonciere>

Several visualization approaches were considered:

- Scene Server
- Feature Server
- Unity Direct
- 3D Tiles in Unity

8.9.6.1.1. The Value of AR/VR Geodata Visualization

Augmented Reality (AR) provides new and exciting ways to visualize information and interact with the real world, including in the fields of geography, architecture, and urban design. AR makes navigating with maps and interpreting mapped data more intuitive. Digital elevations can be observed and dynamically exaggerated to distinguish hills, mountains, elevations, and slopes. AR is a graphical user interface for spatial data, which enhances and improves what users can see and perceive versus using conventional web or desktop applications. Users can rapidly recognize the content and implications of mapped information projected in the real world in a natural fashion at first glance. No translation of abstract symbols or 2D mapping data need take place as it does in conventional mapping and map applications.

Additionally, AR allows the projection of symbols and data out into the real-world spaces where it can be associated with human-created and natural environmental features. This engagement

of sight, sound, and even haptic feedback vibrations can provide a deeper situational awareness and keen perception of spaces and objects within them. Adding the capability to color code or distinguish building attributes through height adjustments or other stylistic methods can allow more rapid and accurate understanding and analysis of metropolitan facility characteristics. This supports better and faster decision-making.

Two-dimensional geospatial applications that are overlaid with 3D features will improve spatial orientation, memory performance, and distance estimation of objects in space. AR can uniquely open the possibility of relating complex information in a simple, direct fashion to assets in the specific moment. This can aid in search, navigation, and planning tasks, and in correlating mapped features and object representations with their real-world counterparts.

If geospatial data is already being fed into mobile devices for common mapping applications, then leveraging the 6DOF knowledge of the location and orientation of the device as determined by the camera and sensors makes that data accessible in new and valuable ways. AR can funnel timely geospatial data to end users to help them make decisions both remotely and on-site. It provides a two-way interface with which users in the field and locally can access and update information in back-end data sources in real-time. If AR visualization of a building energy dataset shows a facility is reportedly heated by one type of system but a field worker recognizes it's actually a different system, that attribute can be corrected on the spot. This tightly integrates experiences in the field with the rest of the enterprise systems.

Ultimately, Augmented Reality has the potential to improve the fundamental value proposition of mapping in the first place, which is to convey spatial information as efficiently as possible. AR is the next evolutionary step in the human relationship with geodata and deserves increased attention and development to maximize value for the geospatial community.

8.9.6.2. Three POVs Viewing

The Ethar team originally proposed a unique Augmented Reality and Virtual Reality (AR/VR) Client approach wherein building energy and geospatial data would be viewable in three modes:

1. remotely via VR visualization;
2. remotely via AR “tabletop” visualization (stretch goal: shared user perspective);
and
3. onsite (“in-situ”) life-size visualization via AR at full-scale in the real-world.

During the project execution, the challenges faced by the larger Testbed 18 team in deriving and gaining consensus regarding clear sets of geodata and building energy data formats, sources, and APIs meant the scope for this D125 Task had to change. Eventually, the following visualization modes were targeted for development:

1. remotely via 3D visualization on a desktop client; and
2. remotely via AR “tabletop” visualization.

Because building energy data was not available via servers for the majority of the OGC T18 period of performance, the visualizations herein are mockups using simulated building data created by Ethar. While the applications were not mature enough to work with live data, the mockups do demonstrate the value of using GeoAR visualization for building energy analysis.

8.9.6.3. 3D WebGL desktop visualization of building energy data

The team first focused on approaches for developing a 3D desktop visualization client that pulled in building energy data from a cloud storage service. This mode utilized WebGL. A WebGL client implementation was produced to illustrate the Nuns' Island structures and allow manipulation of various energy parameters and render styling. The link to this visualization is here: <https://testbed18.ethar.com/>

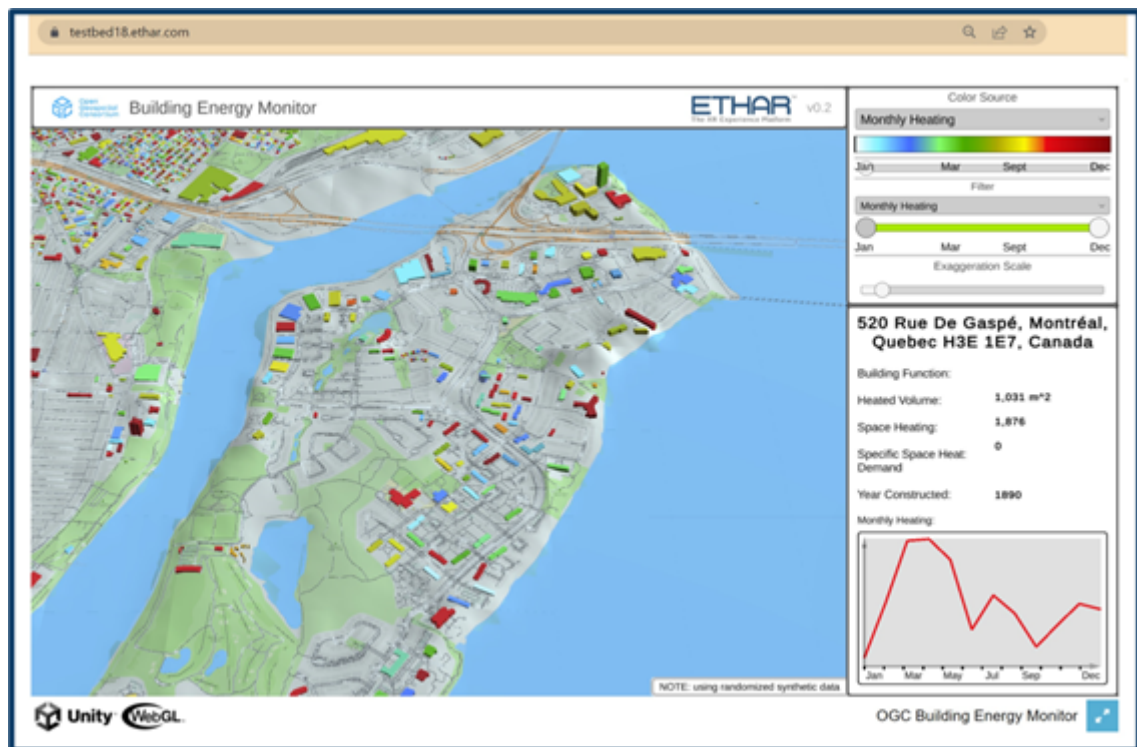


Figure 60 – Ethar WebGL Client Desktop Visualization (Credit: Ethar)



Figure 61 – Ethar WebGL Client Mobile Android Visualization (Credit: Ethar)

Variables available through the drop-down in the upper right corner include:

- Heated Volume
- Monthly Heating
- Space Heating
- Specific Space Heat Demand
- Year of Construction

A rainbow-colored indicator bar appears under the drop-down with the scale from minimum to maximum values of the selected parameter (e.g., slider selection of Monthly Heating readings over a desired 12-month period displays a graph of readings over time). A second drop-down allows filtering of data using a secondary parameter selection. Additionally, slider bars allow further refinement and narrowing of the target data by relevant metrics. Underneath those user interface tools, a slider is provided to increase or decrease the level of Z-axis exaggeration for the 3D buildings.

As each variable option is selected, the 3D buildings on the map are rendered in the appropriate color. Clicking the Maximize Icon in the lower right corner zooms the display to full screen. Clicking in the header materializes a circled “X” icon which, when clicked, exits full-screen mode.

The Pan / Tilt / Zoom / Rotate (PTZR) functionality which was previously working and stable somehow became unstable in the final build. There was not time to sort out the issue. Isolation

attempts indicated Mapbox was not deleting buildings when a tile moved offscreen. Instead, it reused them and moved them to new locations on the new tiles. The result was that buildings appeared in the wrong place once the user moved their viewpoint a short distance from the starting location, ruining the render. Zooming caused the same problem. Mapbox provided some demos which did not exhibit the same behavior. The anomaly is likely in some parameter of Ethar's implementation. Given time constraints, the decision was made to lock the camera down in one place.

8.9.6.4. AR tabletop visualization of building energy data

Next, the team focused on migrating the WebGL functionality into an AR client application. The rendering was designed to spawn a 3D colorized map of terrain, natural and created features, basemap roads and waterways, and color-coded building models based on selected building energy variables. A 2d printed icon served as a “trackable” anchor for the rendering.

The 3D AR data to be visualized was designed to be accessed using a handheld Android smartphone or tablet. The data may also be viewed with selected Android-based smartglass systems (e.g., Realwear HMT-1). For this project, an iOS version was not initially created because the process of distributing an app on an Apple device involves validation through the App Store organization, or via the Testflight over-the-air installation and testing process, both of which take extra time. To move in an agile fashion, the focus was on the Android OS options. Even then, the Ethar team only provided a limited distribution .apk file for install on the test team's devices. The app has not yet been hardened for public availability.

1. The APK file is downloaded from the Ethar server to the client device. The device must be in developer mode.
2. A link is provided to a fiducial marker, which can be printed out and placed on a table.



Figure 62 – Ethar AR Fiducial Marker (Credit: Ethar)

3. The User starts the application on their phone or tablet.
4. The User points the camera until the marker falls inside the camera's field of view.
5. The application will detect the marker and the User Interface will appear on the screen. The 3D map of the dataset will appear on the table over the marker. The user may walk around the table or view the map from any angle as long as the marker remains in view of the camera. A dropdown at top of the screen allows the user to select which of the five building attributes they would like to see. The selection determines the building color based on where the attributes fall between the min and max on the scale of selected building energy variable.

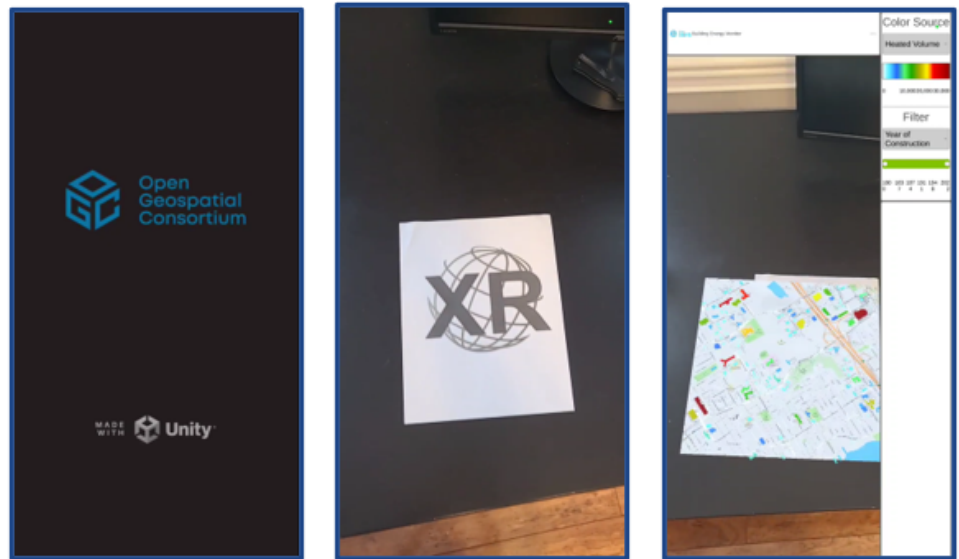


Figure 63 – Ethar Mobile Application Screenshots (Credit: Ethar)

6. The User may then tap on the screen over any rendered building with their finger to display attribute data for that POI. The UI will show the address and building energy attribute data, if available. It will also graph Monthly Heating.

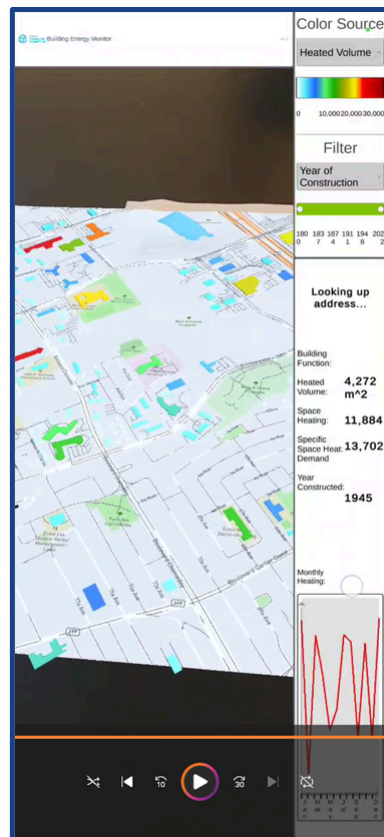


Figure 64 – Ethar Mobile Application Mapped Data (Credit: Ethar)

7. A second dropdown, below the first, allows the user to filter based on any of the remaining four variables and a slider allows windowing of a range between the

min & max values for that variable to determine which buildings are visible. For example, show Monthly Heating Source to display colors, then use the filter to only display buildings constructed from 1960-2000 and then note the difference between that and a view of buildings constructed from 2000-2020.

A short video of the application can be seen here: <https://youtube.com/shorts/qLalwM-ywly?feature=share>

8.9.6.5. AR life-size visualization of building energy data

While the Ethar team was unable to complete development of a prototype AR implementation in a 1:1 life-size “in-situ” viewing mode, the concept remains intriguing. Two mocked-up illustrations are below.

The first view is a “birds eye view” with an AR label of the building data geo-registered to the building. The viewing location in this picture is from a tall building adjacent to the viewing site. Similar perspectives could be gained from a hillside or from an airborne vehicle.

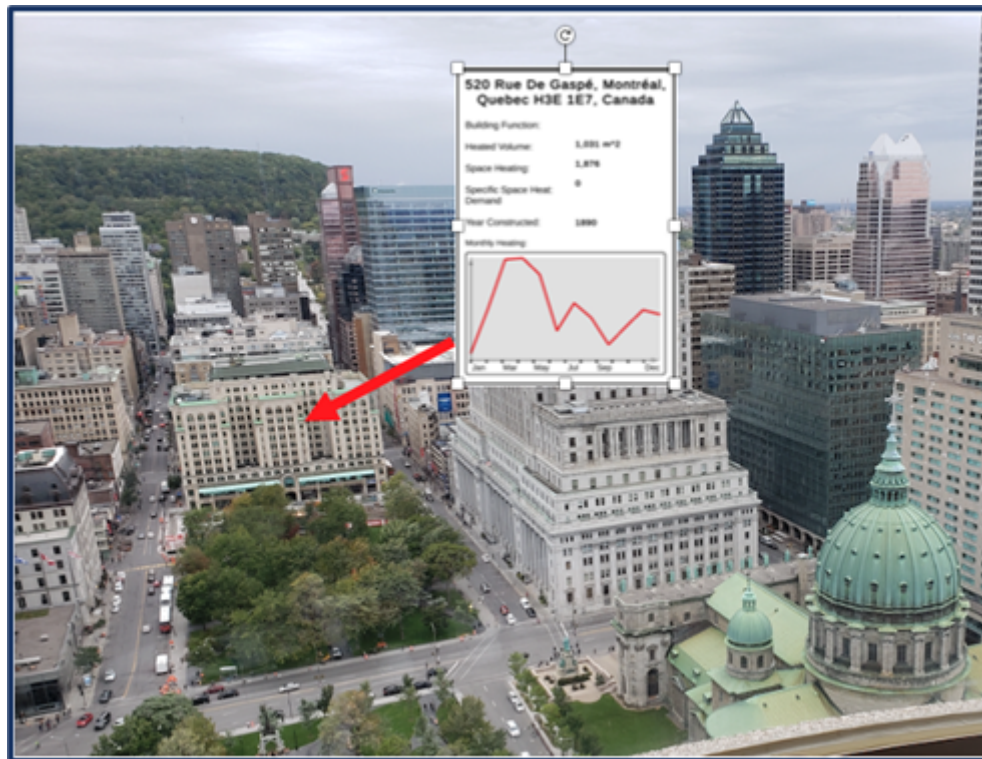


Figure 65 – Bird’s Eye rendering of AR over a real-world scene (Credit: Ethar)

The second view is a “ground level view” with an AR label of the building data geo-registered to the building. The viewing location in this picture is from street level.



Figure 66 – Bird's Eye rendering of AR over a real-world scene (Credit: Ethar)

What information can be gained from an AR life-size visualization of building energy data?

Imagine being a real estate or utility worker on a city street looking for addresses and ownership data for each lot in eyeshot. In-situ AR would allow real-world buildings to be canvassed with various color styles. Text data about the property could be called out into 2D bubbles floating in 3D space associated with each building.

The mockups here focus on individual buildings which may be a focus of a particular analysis. The individual building could have been selected based upon a query, e.g., “show me the building with the least efficient energy usage.” Alternatively a view could be provided showing a synoptic view of all buildings in the scene. Then the most and least efficient could be viewed in perspective against the other buildings in the view.

8.9.7. AR implementation

For a variety of reasons described below, and most notably for the challenge and opportunity it represented, the Ethar team elected to focus on development of the AR/VR client deliverable using the Unity® game engine. Unity provides game developers with 2D and 3D platforms to create video games. Only in the last handful of years has Unity been seriously considered for use in development of AR and VR applications. This requires a complicated subset of conventional Unity functions which must also be extended through a variety of plug-ins to achieve satisfactory results. If successful, the Ethar team’s incorporation of OGC standards aligned with Unity would make access to geo-functionality much more democratic on a global basis and aid in widespread deployment and interoperability of geo applications, especially among smaller businesses.

8.9.7.1. Why UNITY?

The use of Unity for AR is still a developing field, however, with a user-base exceeding two-billion developers, it is a hotly-pursued endeavor allowing the ingestion of geospatial data into Unity. The Ethar team found through hundreds of hours of R&D that there does not seem to be a good off-the-shelf solution for practical integration and manipulation of geodata into the Unity game engine for use in AR and VR applications. Thus, the team explored myriad approaches to pull all the functionality together. Many of those approaches resulted in dead-ends, as described below.

8.9.7.1.1. Choosing the right XR Platform

Choosing the right development platform is crucial for any XR development project, maximizing the openness of the development eco system and ensuring enough flexibility for consumer adoption, taking into account the many technical factors required to successfully operate the platform.

Ethar has chosen Unity as their primary development and deployment platform due to its extensive community developer base and ease of adoption, as well as Unity's full support for all XR devices within the target brackets of consumer consumption.

Unity provides many advantages over competitive platforms such as Web, Unreal, and others with a platforms built for XR and handling the many variations of existing and emerging hardware platforms available for the following reasons.

- Focus on XR Development and Delivery – Unity has a dedicated focus on XR development with built-in support for all the major vendors available today, as well as adopting open standards such as OpenXR to widen its reach. It offers unrivaled performance and delivery / support capabilities compared to its competition.
- Accepted by all major store vendors (Apple, Google, Microsoft, etc.) – App Stores are digital storefronts which enable users to search and review software applications and media via eCommerce sites. Users can browse app categories and read reviews and ratings, then download the applications as a one-time purchase or via subscription models.
- Larger developer community and support – Unity is a very mature platform with constant expansion for developers and consumers alike and is also one of the only tools with dedicated support to both AR (Augmented Reality) and VR (Virtual Reality) on the widest set of devices. The platform is fully supported by the Vendor (Unity) with a progressive road map of features tuned primarily at the mobile market and with sufficient expansion to additionally target more intensive workloads for dedicated devices.

Unity's adoption and support of the .NET platform enables a wide community of developers to build components and solutions for Unity projects as well as enabling the adoption of the many tools and backend services provided by service partners. Unity's support of the .NET platform also enables rapid adoption and shorter turnarounds to connect different solutions together,

such as remote rendering (cloud based rendering from nVidia / Microsoft) and networking solutions that can prove troublesome to integrate for web based solutions.

Ethar is an open development environment, as there are no “one size fits all” solutions in the market. The team uses whichever technologies fit the requirements for solutions, provided there is the technological support to make it sustainable. As the market currently stands, XR standards are still evolving and vendors are slow to adopt the new specifications. Along with many companies currently focusing on internal XR app development, using technologies such as Unity is a very robust approach. Paired with its extensive developer network and reliable deployments, Unity is the safest approach to XR development in 2023 and the foreseeable future.

8.9.7.1.2. The WebXR approach

The web is growing and changing, however, the technology required for XR development is still in its infancy and not all vendors / browsers support the emerging WebXR specifications from the World Wide Web Consortium (W3C). Critical capabilities are still missing or still in draft such as Image Tracking and Spatial Understanding, making it difficult to build a coherent solution that can be used on the many devices out in the consumer market. This makes the pitch for true Web based XR solutions more technically challenging and harder to adopt. What works on one device may not work on many others. As it stands, WebXR’s primary use case is for VR only environments. XR capabilities (when supported) are primarily vision based with no world understanding, limiting the standards usefulness for true AR solutions. For these reasons, WebXR was discounted for this Testbed 18 project.

8.9.7.1.3. Other vendors

There are a handful of other vendors in the market adopting VR, but fewer adopting AR due to the complex needs from both software and hardware, as well as the performance demands on devices to run them.

8.9.7.1.4. Unity-Compatible Solution Architecture Considerations

The UNITY AR/VR Mapping/Geospatial Data Visualization Solution was initially envisioned to be powered by these two open source packages:

- <https://github.com/nicoptere/unity-xyz> Raster tiles, vector tiles, elevation raster tiles
- <https://github.com/NASA-AMMOS/Unity3DTiles> OGC 3DTILES B3DM 3D Buildings
- <https://github.com/Unity-Technologies/com.unity.gis.streaming-framework>

One question was whether the map canvas can be controlled and how to control the Z display order. Various styling considerations were evaluated.

- Support Tile Layers only (3857 web mercator)

- 3DTILES b3dm Batched (3D Buildings)
- XYZ Raster Tiles (PNG, JPG, WebP)
- XYZ Vector Tiles (PBF) (including support for 2.5D Fill Extrusion Effect 3D Buildings)
- XYZ PNG Encoded RGB Elevation Tiles (Mapzen Spec Terrarium or Normal) <https://github.com/tilezen/joerd/blob/master/docs/use-service.md> <https://github.com/tilezen/joerd/blob/master/docs/formats.md>

The above packages have deficiencies in that they do not support the following.

Dynamic and cached mapping services

- ESRI MapServer and OGC WMS – raster
- ESRI FeatureServer, OGC WFS, and OGC API Features – vector
- ESRI ImageServer, OGC WCS – raster
- OGC WMTS -raster tiles

Static GIS Files Based content/files

- GeoJSON
- Shapefiles
- GPKG vector features SQLite
- Other (WKT, CSV, KML, GPX, etc.)

The team evaluated several methods to handle the external data (Server-Side Process or in-App process).

Server-Side Process

- Solution 1 – Build Vector Tiles from services that can request GeoJSON OGR2OGR or Tippecanoe can build vector tiles
- Solution 2 – Build Raster Tiles from services that can request BBOX or TILES MapProxy Python can be used to work with MapServer and WMS and Deliver TMS/XYZ Raster Tiles

In-App as a background process (i.e., don't show the map).

- Solution 3 – Leverage another .NET C# library like MapSui <http://mapsui.com/api/index.html> MapSui can build cache and output raster tiles (as a background rendering engine) (it also supports offline MBTILES SQLite map tiles (vector tiles, raster tiles, elevation/terrain tiles). It wasn't clear how to build this to work in a Unity project.

Use ESRI Shapefile Vector GIS Data support

- <https://github.com/Mapsui/Mapsui/tree/master/Mapsui.Nts/Providers/Shapefile>
- WFS <https://github.com/Mapsui/Mapsui/tree/master/Mapsui.Nts/Providers/Wfs>
- WMS <https://github.com/Mapsui/Mapsui/tree/master/Mapsui/Providers/Wms> <http://mapsui.com/api/Mapsui.Providers.Wms.html>

ESRI Mapping Services (MapServer, FeatureServer, ImageServer)

- <https://github.com/Mapsui/Mapsui/tree/master/Mapsui.ArcGIS> This approach can do map rendering output Raster Tiles that can be served up to unity-xyz to be show in the VR Mode, Table Top Mode, On-site Mode/in Situ (lifesize)
- <https://github.com/Mapsui/Mapsui/tree/master/Mapsui.Rendering.Skia>
- <http://mapsui.com/api/Mapsui.Rendering.Skia.html>
- <https://github.com/Mapsui/Mapsui/blob/master/Mapsui.Tiling/Provider/RasterizingTileProvider.cs>
- <https://github.com/Mapsui/Mapsui/blob/master/Mapsui/Layers/RasterizingLayer.cs>

There are functions for Clipping

- <https://github.com/Mapsui/Mapsui/blob/master/docs/api/Mapsui.Rendering.Skia.Functions.ClippingFunctions.html>
- <http://mapsui.com/api/Mapsui.Rendering.Skia.Functions.ClippingFunctions.Intersection.html>

Memory Provider

- <http://mapsui.com/api/Mapsui.Providers.MemoryProvider.html>
- <https://github.com/Mapsui/Mapsui/blob/master/Mapsui/Providers/MemoryProvider.cs>

Labels

- <http://mapsui.com/api/Mapsui.Providers.StackedLabelProvider.html>
- <https://github.com/Mapsui/Mapsui/blob/master/Mapsui/Providers/StackedLabelProvider.cs>

Async fetching

- <http://mapsui.com/documentation/async-fetching.html>

Possible means of supporting GeoJSON in MapSui

- <https://github.com/Mapsui/Mapsui/issues/1638>

Possible way to show MapSui Canvas if that was requirement in Unity

- <https://github.com/erkyrath/UnitySkiaDemo>

TMG has OpenLayers Background Process for raster tiles

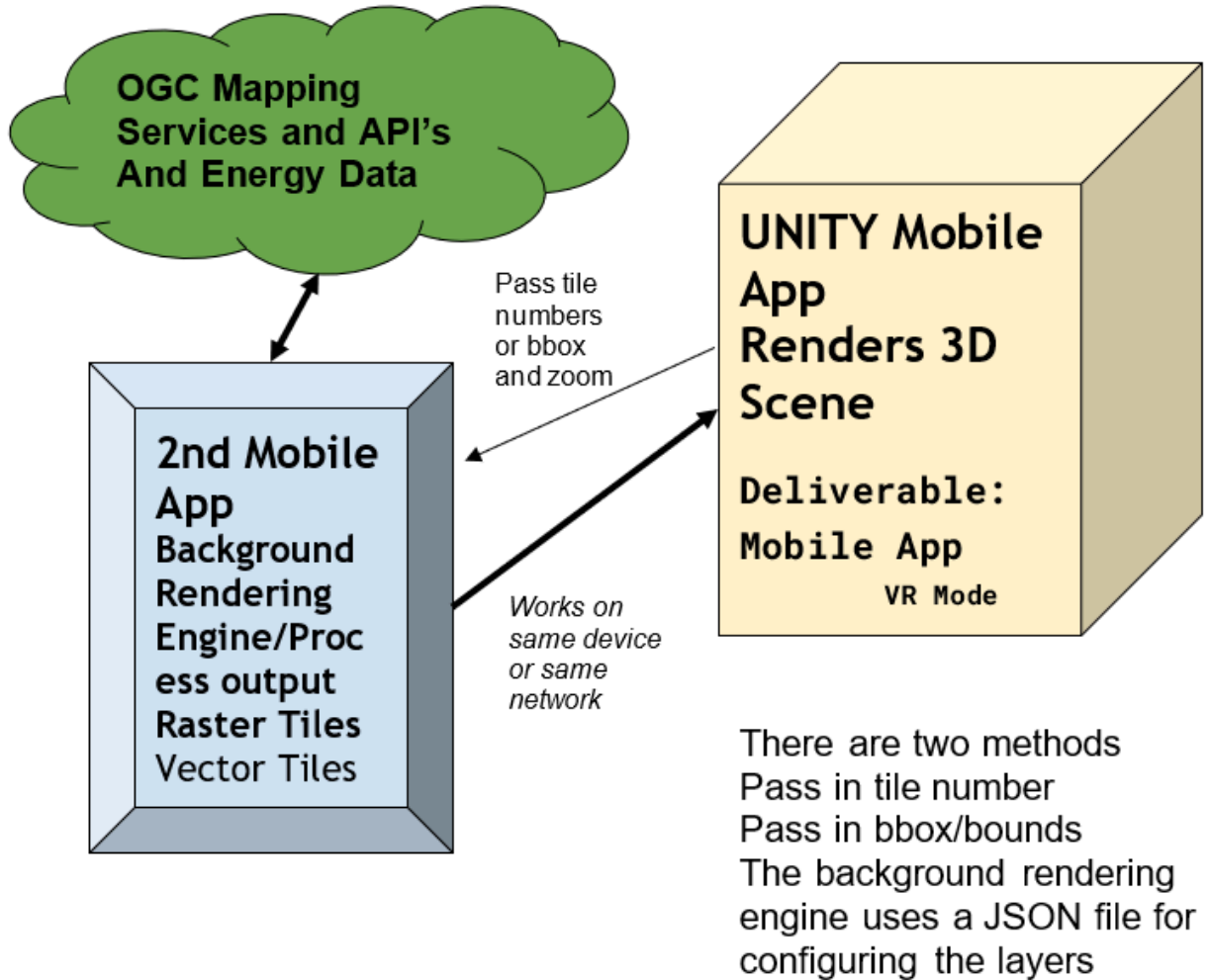


Figure 67 – Ethar AR Application Architecture Fundamentals (Credit: Ethar)

8.9.7.2. Geospatial Data Integration into Unity

There is no good off-the-shelf solution for pulling OGC-compliant geodata into a Unity application (correlated 3D terrain, maps, and 3D buildings). Further, the various formats and versions of tools and data make life complicated. Specifically, there did not appear to be a viable Unity plugin that works with the 3D Tiles, the format of choice of this OGC Testbed. The Ethar team looked at ways to elegantly integrate 3D Tile tools with Unity. Alternatively, the possibility

was explored that 3D Tile datasets provided by the Testbed data processing services would need to be converted into another format prior to integration into Unity (e.g., i3s).

- i3S (Indexed 3D Scene Layers) is an open format and OGC Community Standard originally developed by ESRI in 2015 to manage streaming of large, heterogeneously distributed, 3D geographic datasets and content.
- 3DTiles is an OGC Community Standard originally developed by the Cesium team in 2015 for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds. This specification has become widely used and there are many tools available to create & work with data using this standard.
- 3DTiles 1.1 (aka 3DTiles Next) candidate community standard is the 2021 update to the original 3D Tiles Community Standard. It expands the core 3D data manipulation functionality with new features to stream semantic metadata efficiently, run massive simulations and analytics via spatial indexes, and integrate with the glTF software and extension ecosystem. Because the specification has only been widely popularized in the past few months by the OGC and has not yet been certified as an approved Community Standard, there are fewer tools available for efficiently working with this format. Additionally, some of the features in the earlier 3D Tiles standard have been depreciated or modified, so tools that work with 3D Tiles do not, in many cases, function properly in a 3DTiles 1.1 application.

In summary, the work on this testbed highlighted the major development gap that prevents open-source, large-scale geo datasets to be imported into Unity for 3D/AR visualization and in alignment with OGC standards – in particular, 3D Tiles. Although a 3D Tiles plugin exists for the Unreal game engine, that approach is also not usable without considerable bespoke programming.

8.9.7.3. Technical Approaches Evaluated

Various solution approaches were analyzed to determine an efficient, workable application.

8.9.7.3.1. NASA AMMOS Approach

The first approach used by the Ethar team was to leverage an open source package from NASA to utilize 3D Tiles: <https://github.com/NASA-AMMOS/Unity3DTiles>.

The concept was to pair the NASA code with the Unity-XYZ Package: <https://github.com/nicoptere/unity-xyz>

to support a complete AR/VR/AR visualization of 3D Geospatial data with elevation/terrain, satellite imagery and basemaps, and reference overlays. Various example implementations were reviewed.

- The NASA AMMOS plugin was originally used to show a few small chunks of terrain from Mars. The authors invested a lot of their effort in the caching logic – buildings load

over time. 50%-60% of the code was spent on trying to identify what data needs to be displayed.

- Unfortunately, the original developers of the NASA Plugin did not spend time resolving critical issues for working with 3D Tiles elements. The code had not been significantly updated since 2016. It is possible that some of the 3D Tiles elements did not even exist at that time.
- There are parts of the JSON text file that are not supported.
- Only one of the possible four different sections of a 3DBM file is loaded, however the remaining three sections are necessary for proper object loading on the map.
- The original plugin does not accommodate the CESIUM_RTC extension to work with glTF objects. The Ethar developers did update the code to now accommodate this extension, as illustrated below.

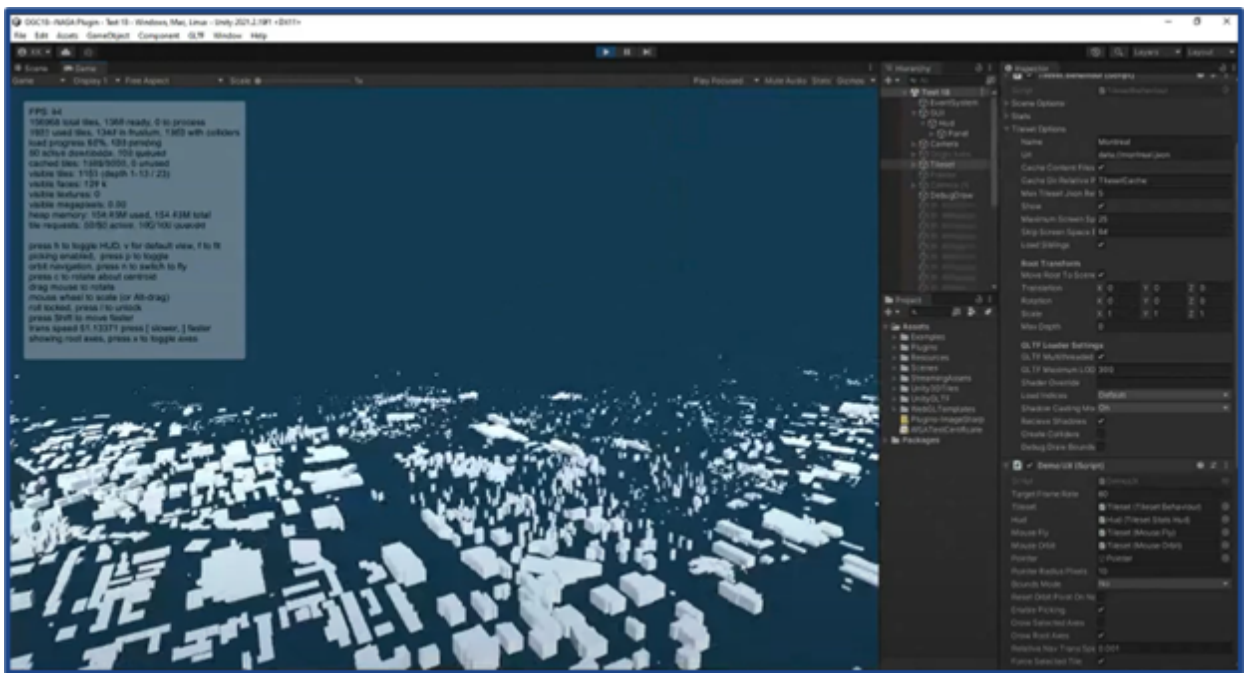


Figure 68 – NASA Unity plugin rendered through updated CESIUM_RTC code (Credit: Ethar)

- Once a basic rendering was achieved, there were still significant issues with the target building dataset inexplicably floating in space above or below ground level of the tiles. A very large amount of time was invested to try and understand and correct the cause of this effect. The team found a client viewer from an Australian developer that did correctly show the data, so the anomaly was somehow specific to the code set with which the team was working.

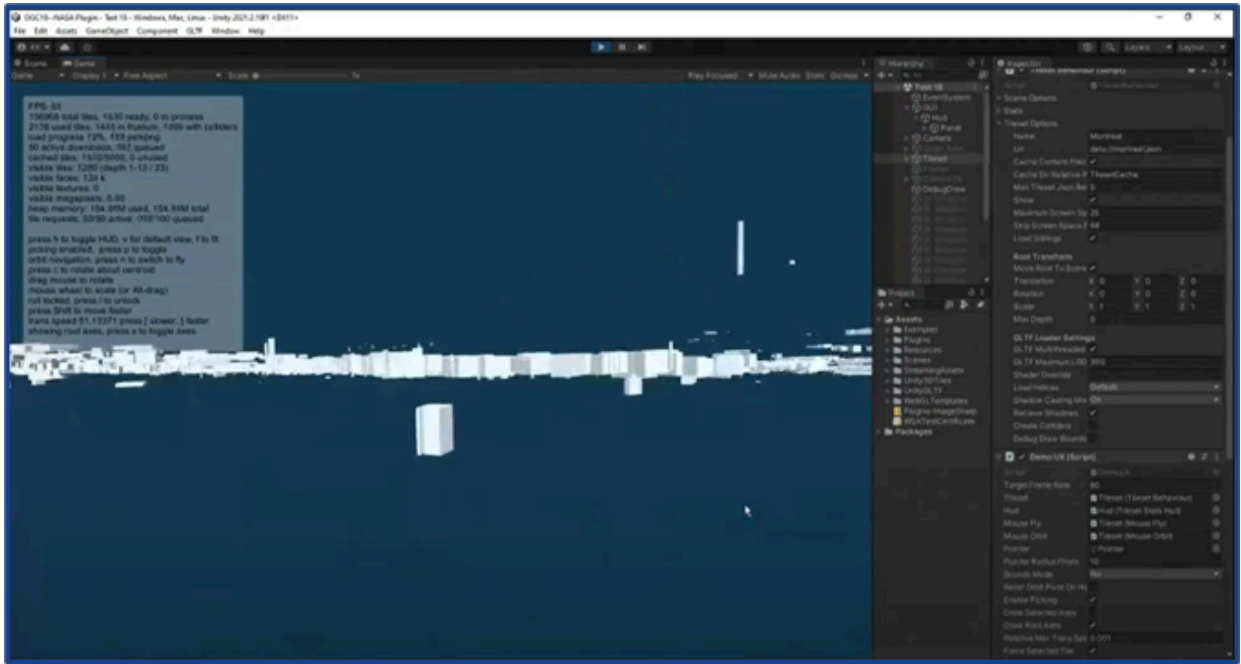


Figure 69 – NASA Unity Plugin rendering with building vertical offset anomalies (Credit: Ethar)

While the NASA plugin works with the simple datasets used by the original authors, it is not capable of dealing with complex 3D Tile datasets. Several Github branches have been created over time off the main NASA plugin code with attempts to address some of the issues, for example, some of the 3DBM items. However, none of this code was in a usable state for this OGC project. At every level, there were challenges. Ethar made significant changes to the NASA code over a two month period but was unable to overcome the nuances of all the issues. This approach became prohibitive from a time and funding perspective and was ultimately abandoned. Even if that effort had been successful, issues would still have remained to be addressed related to ingesting a basemap and terrain data.

- The original NASA AMMOS Unity was only designed to work with a small number of simple terrain datasets of Mars. With additional investment, the NASA AMMOS Unity plugin could eventually be made to work with general, complex geo datasets like those used in Testbed 18 and it would be quite valuable to the geo community to have such an open source option.
- Ethar could publicly host the code updates that were created, and this would then be the most advanced version of the plugin available, although this would still not be fully functional. While that would be a nice gesture of goodwill for the geo community, it would also require ongoing maintenance and updates to remain useful. This would then be the best Unity 3DTiles viewer – most complete but still lacking. For example, the utility is only designed to work with 3DTiles and provides no support for 3DTiles Next.
- Items still needing to be addressed included:
 - completing support for Cesium .rtc extension so buildings won't be floating up in the air;

- buildings were visible but the project still needed to provision for 3D terrain;
- still needed a basemap;
- still needed to merge in building energy data; and
- still needed to build the GUI.

8.9.7.3.2. Unity GIS Streaming Framework Approach

The Ethar team also looked into using the GIS Streaming Framework plugin for Unity, which is (was) still under development during the Testbed timeframe. This framework allows Unity to stream geospatial data into its runtime.package using the 3D Tiles Next format.

<https://github.com/Unity-Technologies/com.unity.gis.streaming-framework>.

The utility was apparently being worked on by only two developers, whose current level of support could not be determined. The framework is still quite immature, and it does not seem to provide any support for the older 3D Tiles format, although the documentation claims it does so. For that reason, the use of the Unity Streaming Framework for the OGC Testbed was deprioritized by the Ethar team.

One approach could be to leverage all the logic developed by the Ethar team up to this point, for example, the caching logic necessary to display the different datasets, and layer it on top of the 3D Tiles Next support provided through this Unity Streaming Framework. The game engine is exceptionally good at managing what is in the camera view at any given time – what needs to be displayed and how to display it. This would require quite a bit of additional work.

Several other branches off the Unity GIS Streaming Framework plugin were examined:

- <https://github.com/Geodan/UnityViewer> – Created by Geodan in the Netherlands in 2019. An incomplete application with no documentation.
- <https://github.com/aiunderstand/B3dmViewerUnity> – sounded promising, but ReadMe doesn't exist; the video link is broken; and the code hasn't been updated since 2018.
- The plugin is designed for 3DTiles Next.
- The Unity game engine is very good at figuring out which buildings in the dataset are inside of the camera view, and optimizing rendering of those buildings vs. those out-of-view.
- There does not appear to be any backwards compatibility for 3D Tiles format. Individuals at Unity may be working to provide this type of functionality, however that status is unknown.
- Because the only available datasets were in the 3D Tiles format, the Ethar team never got to the point of being able to display visual content with this plugin and this approach was abandoned.

- Invest in a project to collaborate with Unity and others to complete and refine this Unity GIS Streaming Framework plugin, including developing backwards compatibility for the 3D Tiles format.

8.9.7.3.3. Mapbox “Maps for Unity” Approach

The third approach attempted for creating a functional Testbed demo was to abandon the 3D Tiles functionality completely and utilize a Mapbox-based Unity solution: <https://www.mapbox.com/unity>.

Mapbox is a privately-held US provider of custom online maps, with a strong allegiance to creating open-source mapping libraries and applications. They offer a nice portfolio of tools for creating customized maps. With significant early contributions to OpenStreetMap, the company that had been based on providing open-source code eventually moved to a proprietary licensing model in 2020. The company still farms data from OpenStreetMap, NASA, and other open and proprietary datasets.

- The Mapbox “Maps for Unity” software development kit is a collection of tools enabling Unity developers to interact with Mapbox web services APIs to create game objects associated with real-world map data using a C#-based API and graphical user interface. The SDK includes worldwide Mapbox data:
- <https://docs.mapbox.com/mapbox-unity-sdk/> <https://docs.mapbox.com/unity/maps/examples/world-scale-manual-align-ar/>
- <https://docs.mapbox.com/unity/maps/examples/tabletop-ar/>
- <https://docs.mapbox.com/unity/maps/examples/city-simulator/>
- <https://docs.mapbox.com/unity/maps/examples/world-scale-ar/>
- <https://docs.mapbox.com/help/tutorials/create-a-map-in-unity/>

The SDK also allows for some advanced data styling: * <https://docs.mapbox.com/unity/maps/guides/styling/> * <https://blog.mapbox.com/styling-3d-buildings-in-unity-cecbe156f898>

- After extensive investigation, it was clear that Mapbox had originally made a serious investment in the Unity plugin in the 2019 time period. The code is well-structured, comprehensive, and works reliably. Unfortunately, since that original effort, the company seems to have abandoned the project. Multiple inquiries from the Github community have gone unanswered, including those from Ethar. Perhaps Mapbox determined that, at least at that time, there were not enough paying clients interested in using Unity to cost-justify the effort. According to interactions with some Mapbox employees via LinkedIn and email, the company also has no plans to maintain it in the future.
- Mapbox Unity does not support 3D Tiles, however, it does support 2.5D vector tiles for buildings. The Mapbox buildings are a substantial difference in height and shape from the OGC-provided building models. The OGC buildings are perceived to be more accurate. In a top-down view on Google Maps comparing the terrain image vs map view, it appears that

there is a better match with OGC data. Also, Mapbox buildings in downtown Montreal do not seem to be tall enough.

- Mapbox Unity users are locked into using data in the Mapbox Cloud and cannot easily reference any third party/self hosted URLs. Although it is possible to upload original and 3rd-party building data to Mapbox servers, when the Ethar team pursued this approach, it was discovered that Mapbox data limits did not allow the Testbed building datasets to effectively load. The extra effort required to work around this limitation constrained the possibilities for this OGC project. The Ethar team was therefore limited to utilizing the buildings that Mapbox provides. Unfortunately, this meant not enjoying access to the building IDs and extra building-oriented data fields provided by the Testbed Service Providers.
- Invest in a project to collaborate with Mapbox and others to complete and refine this Maps for Unity plugin. While the conventional mapping community may not have shown strong interest in utilizing game engines for rendering, the use-cases for geospatial data are increasing geometrically. GIS stakeholders who are invested in ESRI, Cesium, or proprietary platforms have had little motivation to migrate to tools designed for game developers to plan and code 2D and 3D video games. However, there are now an increasing number of Unity and Unreal Engine users who are very interested in leveraging geospatial data.
- Many business use-cases for geospatial data rely on making information easily available via web browsers. Unity and Unreal Engine are famously lacking comprehensive support for web-rendering, although that is changing. Again, traditional geospatial specialists would prefer to choose a software stack that works more easily with the web. It would be exciting and valuable to conduct a project that pushes the frontier of ingesting geospatial data into game engines for rendering via the web. This is the domain of the Metaverse in which OGC members should surely wish to master.

8.9.7.3.4. Google “Maps SDK for Unity” Approach

The team also evaluated the Unity solution from Google. Maps SDK provides development tools and pre-optimized assets for creating mobile games played in real-world environments using high quality Google Maps geo datasets. The client library identifies the user’s physical location, then retrieves related map data and geographic features (buildings, roads, waterways) at runtime rendered as Unity GameObjects.

- https://developers.google.com/maps/documentation/gaming/overview_musk

This plugin is very basic and proprietary. Because it is not focused on open geospatial data and standard formats, the platform is not easily extensible. These factors led to eliminating this approach for the complex OGC Testbed data challenge.

8.9.7.3.5. Microsoft Maps SDK for Unity Approach

Maps SDK is a Microsoft Garage project (side-projects pursued by Microsoft employees). It enables the streaming and rendering of global 3D terrain data from Bing Maps into Unity applications. The plugin has been optimized for mixed reality projects on a variety of AR/VR hardware and software platforms. Users can configure maps and attach game objects. While the Github repo includes documentation, and supporting scripts, it does not include the core source code for the SDK. The supporting scripts are Unity components and shaders that extend the SDK. Maps SDK was initiated in 2019. It has been developed almost exclusively by one Microsoft coder, and commits have dropped off significantly in the past year.

- <https://www.microsoft.com/en-us/maps/mixed-reality>
- <https://github.com/Microsoft/MapsSDK-Unity>

Like the Google plugin, Microsoft Maps SDK provides some easy-to-use but basic capabilities. Because of its proprietary nature, the Ethar team disregarded this approach for the OGC project.

8.9.7.3.6. ESRI ArcGIS Maps SDK for Unity Approach

The Ethar team explored using the ArcGIS Maps SDK for Unity from ESRI. The team would have had to convert 3D Tiles into i3S SceneServer using the loadersGL Tile Converter. ESRI does support OGC API Features, however, ESRI does not support GeoJSON, so it would be necessary to transform GeoJSON and ESRI-supported format.

- <https://developers.arcgis.com/unity/spatial-and-data-analysis/geometry/>
- <https://developers.arcgis.com/documentation/common-data-types/geometry-objects.htm>

ESRI's solution does not support vector tiles at this time. It is also very centered on using their data and hosting the data in AGOL or Portal. Because of the conversion complexities and proprietary nature of this approach, it was deprioritized by the Ethar team.

8.9.7.3.7. Cesium Unreal Plugin Approach

The other major game development software vendor for cross-platform XR development is the Unreal Engine by Epic, however this is primarily desktop focused and requires higher performant hardware. Unreal does not really have a mobile offering, especially in XR which makes it somewhat undesirable for use in this market. During the term of the Testbed 18 project, Cesium began offering a Unity plugin solution for the Unreal Engine, though not for Unity.

- <https://cesium.com/platform/cesium-for-unreal/>

The plugin is designed for commercial use and requires a Cesium ION Subscription starting at \$150/month with additional charges for increased usage. Cesium is the contributor/founder

of the 3D Tiles specification, so except for the fact that the Ethar team was focused on a Unity solution (not Unreal Engine), this might have been a workable approach. For this Testbed, the Cesium Unreal Engine plugin was thus dismissed.

8.9.7.4. AR/VR Client Development

Regardless of the method for integrating data into Unity, the Ethar team still had to develop a workable User Interface. This was one of the successful elements of the project. In particular, the capability to filter the data by a variety of building attributes turned out to work quite well. For example, the user can set a start- and end-date timeframe for when the buildings were constructed. The restricted dataset can then further filter the set based upon a set of defined building energy parameters. This should help support the objectives of the Canadian Geospatial Data Infrastructure (CGDI) program when analyzing buildings for energy efficiency and refurbishment.

Because the Ethar team could not get the Mapbox solution to work directly with the OGC Service Provider API and building energy dataset, when a given area of interest is selected in the Client, the OGC building IDs were not available. To compensate, a separate query is initiated to the Steinbeis data endpoint using the latitude and longitude associated with each building in the target set requesting any available building energy data at that location. At the time, the Steinbeis data was the most easily available from the various Building Energy team options. The program query returned the following five variables.

- Heated volume (m³)
- Monthly Heating (units unknown)
- Space Heating (units unknown)
- Specific Space Heat Demand (units unknown)
- Year of construction (year)

8.9.7.4.1. Development Environment

The development team evaluated the possibility of using FLUTTER for creation of the app to achieve some cross-platform functionality. Flutter was created by Google and is now provided as an open-source UI software development kit to build cross-platform applications for Android, iOS, Linux, macOS, Windows, and the web using a single common codebase. Adding a Unity project to an existing Flutter project or application was deprioritized in deference to C# programming because of the better established integration of C# with Unity.

- https://pub.dev/packages/flutter_unity_widget
- <https://github.com/Glartek/flutter-unity>
- <https://github.com/juicycleff/flutter-unity-view-widget>

- <https://youtu.be/4KywnaaykBU>
- <https://youtu.be/REeJmCNTfMc>
- https://youtu.be/icxfRM_qEnU

The team evaluated various features options, such as the ability to toggle the map and assets on/off, creating an opacity/transparency slider, and a means for modifying the display order on the Z-axis.

8.9.7.4.2. API and Data Service Unavailability

Both of the visualization client applications eventually developed by Ethar (Tabletop AR and WebGL) were ultimately configured to use hard-coded, random data which was synthetically-generated by the Ethar team. In this regard, there was a failure to meet one of the key objectives for the project. This shortfall was due to several understandable but limiting factors: first, for much of the Testbed project timeframe, the API was not clearly defined. To be fair, this was a complex undertaking from the beginning for all parties. Also, sufficiently detailed explanations of each endpoint were late in coming or never provided even once APIs had been published by the service providers. Additionally, there were peculiarities and limitations of using the data services which affected their reliability and were not thoroughly documented. Again, the teams were moving rapidly and did not utilize quality control measures which would normally be present in a production program. Finally, the gaps in the incomplete test datasets were never sufficiently filled. There are hundreds of fields in each set, many of which require an individual interpolation algorithm to synthesize missing data. This would be a complicated endeavor in any instance and was not forthcoming in the limited timeframe with the limited budget. The resulting data gaps did not allow generation of any meaningful visualization.

The Ethar team focused attention on working with the Steinbeis data server. Another issue arose near the end of the project time frame when trying to access the datasets in near real time. Test evidence seemed to indicate the client application was issuing GET requests through the endpoint at the rate of approximately two-hundred requests per minute. This rate is desirable for an AR application where the user is constantly shifting the location and orientation of their handheld device or smartglasses relative to the real world. The service provider database was unable to respond rapidly enough and was shutting down access to the data as though the request were a denial of service attack. For various reasons, Steinbeis did not find it practical to circumvent this server limitation with the time allotted, so suggested perhaps Ethar could limit data queries to only once or twice per minute. This would result in an unacceptable lag time and there were not enough resources working the project to negotiate and then code a meaningful modification.

For all the reasons mentioned, the Ethar team self-generated a set of random data building on the Steinbeis dataset. This was then used to color code renderings for all of the five target variables in the client. All 3D buildings in Montreal will appear white except for the Nuns' Island structures where actual building energy data was available and the buildings are color coded.

8.9.8. Recommendations and Next Steps

There were a fair number of lessons-learned throughout this project. Some key findings include the following.

8.9.8.1. Serial vs. Parallel Testbed Development Paths

As the Testbed initiative developed over time, it became apparent that test datasets, processing services, and the target API would not be available to Client developers in a usable format until very late in the Testbed term. In order for Client vendors to make progress in the mean-time, a parallel development path was established for those efforts using mocked-up datasets within the Initial Processing Track. The Ethar team made the decision to self-synthesize data and hard-code it into the application until the Testbed API and datasets became available. The team gathered requirements and conceptually designed the AR application.

The diverse Testbed 18 team on the building energy tasks were developing data sources, service processes, API formats, and the client applications all at the same time. It took the majority of the project term before the project was cleanly divided into “Tracks”, with the Initial Processing Track being to synthesize data and hard-code it into the Client applications until such time as the ‘real’ datasets and API endpoints became available. Unfortunately, given other challenges in developing the Unity AR app, and instability and random unavailability of the Processing Service datasets, the Ethar team was never able to create a fully-functioning app through the API.

In future Testbeds, a sample Processing Service, API, and synthesized dataset MUST be provided to the Client development teams within the first month or the funding provided to them will be wasted while those teams sit in on meetings discussing the challenges and opportunities of developing various Processing Service approaches. Unless real data is provided at the outset, it is imperative that a working, representative dataset be provided.

8.9.8.2. Game Engine AR Support using Web-Based, Open-Source 3D Geospatial Data

In the big picture, developers who have been focused on using geospatial data in Android and iOS applications have mostly had to rely on top-down 2D maps from Google, Mapbox, or other sources like OpenStreetMaps. There has been little need for enabling 3D terrain and building data for mobile applications, and not many example 3D map viewers in the marketplace. Interestingly, as augmented reality and virtual reality applications become more mainstream, there will be a growing demand for 3D data. This will require significant investment in tools and plugins which provide Unity, Unreal Engine, and other game engine support for working with geospatial data. In keeping with OGC’s FAIR principles, follow-on testbed projects should be explored to develop and demonstrate AR applications using game engine tools that work with open-source data and OGC APIs and frameworks.

8.9.8.3. Next Steps

- Finish supporting existing 3DTiles format via the NASA plugin.
- Prepare for the future by evaluating the Unity 3DTiles Next GIS Streaming (as of now, there is no known support for using 3DTiles with Unity).
- Expand visualization capabilities for Canada’s building energy programs — e.g., drag bounding box and highlight top ten energy using buildings within the selected area.
- Ultimately, the team must be able to get the OGC building data ID so that the full info can be accessed. The team must collaborate with whomever is building the official endpoints. The Ethar team was never able to use building data through Mapbox via a separate query to the Steinbeis data.

8.9.8.4. Postscript: Cesium plug-in for Unity

On November 22, 2022, after completion of the technical research and software development for Testbed 18, Cesium (cesium.com) announced Cesium for Unity is now available as a preview. Cesium for Unity combines the 3D geospatial streaming capability of Cesium, 3D Tiles, and the robust foundations of Cesium Native with the Unity ecosystem.

Real-World 3D Geospatial Capability for Unity:

- built on open standards and APIs, Cesium for Unity combines the 3D geospatial capability of Cesium and 3D Tiles with the Unity ecosystem;
- a full-scale high-accuracy WGS84 globe;
- visualize massive high-resolution real-world photogrammetry and 3D geospatial content at runtime using 3D Tiles ;
- free and open-source visualization plugin;
- integrated with Unity’s Game Objects, Components, Character Controllers, and more;
- optional paid subscription to Cesium Ion for one-click access to global curated 3D content including terrain, imagery, 3D cities, and photogrammetry; and
- support for multiple platforms including Windows, macOS, Android, and VR platforms such as Quest 2 and Quest Pro.

While the 3DTiles plugin for Unity from Cesium looks promising, there is still much to learn and challenges to overcome in integrating large-scale datasets into augmented reality applications.



9

TESTBED CHALLENGES, LESSONS LEARNED, AND RECOMMENDATIONS

TESTBED CHALLENGES, LESSONS LEARNED, AND RECOMMENDATIONS

Through the Testbed-18 activities that worked towards the Testbed goals, there were challenges, lessons learned, and recommendations that arrived from those challenges which are documented in this section. The challenges, lessons learned, and recommendations from the Testbed can help provide direction for future testbeds, research, or pilot initiatives as well as future actions that stakeholders can take to enable continued progress towards geospatial buildings and energy data interoperability.

The challenges, lessons learned, and recommendations presented within this section are loosely classified as either data or practice related and the contents of the following subsections are divided as such. Each of the subsections are further divided into general and specific sets of comments. For clarity, some recommendations from previous investigative track sections are also repeated or referenced here.

9.1. Data, standards, and interoperability

Data could be described as the key issue within the Testbed. Data provided for use in the Testbed experiments did not meet the requirements of making location information and services FAIR – Findable, Accessible, Interoperable, and Reusable. There were also quality and useability issues with the datasets used for the Testbed. In the following subsections general and specific comments about the data and standards used within the Testbed are made.

9.1.1. General comments

The datasets that provided 3D volumes and building characteristics like the CityGML data from the City of Montreal were provided in a usable format but generally didn't contain enough data for the type of analysis the Testbed goals aimed to address. Part of the Testbed goals was the conflation of the input datasets to create a relatively complete 3D volumes and building characteristics dataset for the City of Montreal but the availability of the input datasets for this region is an anomaly rather than the norm (in Canada). The following lessons were learned.

- Datasets that contain 3D volumes and building characteristics should be offered for all buildings in a region with notes where the datasets are incomplete.
- Datasets containing 3D volumes and building characteristics should be provided in a variety of formats that mirror the data formats of other jurisdictions.
- Dataset schema choices should be aware of the type of parameters that they will need to store. For example, the complex value storage capabilities of CityGML Energy ADE, and other data formats such as Mapbox Vector Tiles, FlatGeobuf, and glTF are designed to only support attributes with simple values. In the 3D Tiles / glTF representation of the

combined geospatial and energy datasets accessed within the Testbed, the monthly data was therefore not included in the conflated datasets in the 3D Tiles / gITF formats.

Building characteristics related datasets were not formatted to easily allow conflation with other related datasets and did not contain enough data to be useful in a broader geographic context (provincially or nationally in Canada). Inconsistent naming of attributes, schemas, and data ontologies that connect the datasets and terms within them to each other can present significant technical and administrative barriers to data conflation. Attributes and parameters within different datasets and standards are named differently, but represent similar or the same data points, attributes, or parameters. This leaves the dataset user performing extra work to ensure that dataset semantics differentiating them are considered properly during the conflation process. Sometimes the technical nuance of parameter matching can present a barrier to one who may be expert in other domains (i.e., geospatial software development) but not building science and energy performance. This was especially observed within the Testbed when geospatial software developers tried to use Canadian building characteristics and building energy data.

Examples of semantics' issues

- Levels of geography and their boundaries and naming conventions are a challenge when trying to conflate datasets containing 3D volumes, building characteristics, building energy, and other related data such as sociodemographic attributes. Within Canada those geographic boundaries can include [Statistics Canada boundaries](#) (including both Census geographies and Forward Sortation Areas from Canada Post), weather file zones (e.g., [HOT2000 Climate Map zones](#) versus [Engineering Climate Datasets zones](#)), government boundaries (including municipal boundaries and the cadaster or parcel fabric consisting of individual lots and property assessment) and individual addresses and associated points. The previous geographic boundaries can intersect with each other and different datasets can and do use different geographic boundaries which presents a strong barrier to data conflation. What is known as the “modifiable areal unit problem” is well known to geographers and geospatial analysts but is not well recognized amongst engineering and energy efficiency professionals. This is especially complicated when geographic boundaries commonly provided with building characteristics and energy data cross government boundaries such as one FSA region lying within multiple municipalities in Canada. For statistical robustness of spatially integrated datasets, it is considered best practice to attribute buildings and energy data at the parcel scale, while simultaneously assigning higher aggregations such as by geography and building type to the parcels. For example, a parcel would inherit the name of the census tract, municipality, region, and province within which it is contained. Similarly, a building on a parcel would be assigned several hierarchical building types. This enables data to be grouped and aggregated to any scale for privacy and commercial data compliance and reporting purposes and avoids statistical inaccuracies inherent in disaggregating data provided at higher geographies to lower geographies.
- Working at the individual building level, the unique identifying labels can differ between datasets as well and it can often be unclear whether the label refers to a land parcel, a group of buildings on a land parcel, or individual buildings on a land parcel. Datasets would be easier to work with if consistent semantics were used when referring to a unique building. Unique building identification schemes such as the [one proposed by Statistics](#)

Canada could be adopted for unique identifiers when publishing datasets in usable formats (FAIR) such as CityGML.

- Datasets within Canada contain a mix of English and French. French text was present in many of the datasets used in the Testbed. In some use cases during the Testbed, the internet browser requested that the language of the response be provided in English, but building attributes were only available in French. If attribute content in the source data was multilingual (e.g., all linguistic text or controlled vocabularies would be available in both English and French), the responses could have all been in French or English. Consistent application of official languages in datasets as well as schema designs that support multilingual data would be beneficial for data interoperability.
- Units used for standards and energy datasets vary between different building topologies and contributing organizations. Within this Testbed, the units of energy that were used was kWh. Yet for low rise housing in Canada, GJ is the convention for energy consumption which was identified as another general data semantics issue related to building energy.

There are many Canadian data sources at a variety of geographic scales that could be considered relevant to the goals of the Testbed, which included the following:

- building parcel fabric (which contains unique numeric identifiers for each parcel, is essential for building energy mapping for spatializing the property assessment data and managing the one-to-many relationships in both directions between parcels and buildings);
- some building permit data (electronic and paper format); and
- remotely sensed data (orthophotos, lidar, satellite).

There is a strong need for coherency in the provision of building stock data and for accelerating efforts to provide it in a standardized and portable format. Within Canada, a national push towards semantic data interoperability would remove barriers to 3D volumes and building characteristics data conflation and enable data interoperability and further NRCan's goals for the Testbed. To that end, conducting stakeholder engagement that seeks to produce guidelines or standards that define both the semantics of and schemas of building characteristics would enable data interoperability. The national building layer initiative is an excellent example of a pathway to enable improved buildings and energy data interoperability.

9.1.2. Specific challenges

9.1.2.1. Montreal 3D data (Building characteristics datasets)

There were a number of issues that led to problems in using the City of Montreal data as follows.

- Lack of available valid building geometry for all buildings in the city.

- Need to represent each building as a single building and not aggregate multiple buildings as one building. The Notre-Dame-Des-Neiges cemetery was one example where many buildings are bundled as a single building feature.
- Need for data on below-grade construction, including geometry, number of stories, basement type and presence of unheated parking lots to be able to supply the energy calculation engines with appropriate input data.
- Polygons in the source data were often not planar, which violated the geometry requirements.
- Sometimes the ring geometries had consecutive identical points or polygons where all points were collinear. These were ignored when importing the data.
- The CityGML LoD 2 building data violated constraints of the CityGML model. The LoD 2 solid geometries claimed to be solids, but in general were not solids. The solid geometry just referenced a set of polygons, but often these were not the patches of a single closed composite surface. That is, the gml:Solid elements did not conform to the requirements of GML (or ISO 19107). To fix this issue, the building solids would need to be analyzed and if the bounding surfaces form multiple shells/solids, then that building would need to be split into multiple buildings.
- There were a few hundred buildings in the dataset that were missing bounding surfaces where the solid geometry referenced bounding surfaces in other buildings. This resulted in having the same solid geometry for multiple buildings. This caused issues such as mismatched attributes and some bounding volume / visibility issues as well. Essentially, wherever there were references from the solids to bounding surfaces from other buildings, those bounding surfaces should be moved to the bounding surfaces of the building from where they were being referenced in the solid, and removed from the building where they were not referenced by the solid.
- The City of Montreal information is presented based on land parcels. This Testbed-18 task required information to be presented based on buildings in order to assign the correct attributes (such as building type) to the buildings. However, the parcel fabric is still essential for effectively managing buildings data and because it is the legal spatial scale to which municipal by-laws are applied. Parcel-wise data could still help planners identify regions to target energy programs towards. Although not considered in the Testbed, the point in polygon data conflation approach is more effective than address data linking for mapping measured utility data. The meter, expressed spatially as a point or latitude and longitude, is placed within the polygon of the parcel to which the property assessment data and building footprints are associated.
- Although there is a difference between similar building types and the activities being done in them, similar archetypes are assigned to similar buildings even if their functions are different. Building type refers to building typology (e.g., office building), but function relates to the activities done in the building (e.g., the difference between an institutional office and a commercial office).

9.1.2.2. Housing archetypes data

Generally, the Testbed participants found low-rise housing related building characteristics and energy data hard to use and conflate with the other Testbed data. These comments applied to both EnerGuide for housing and the HTAP Housing Archetypes. One of the largest issues was lack of access to documentation in the form of data dictionaries, diagrams, and technical guidance providing information on how the data could be used beyond a building energy simulation context. Non-building scientists (e.g., geospatial software developers, planners) need context on how to identify the most important parameters within each data source and instructions on how to use them. Additionally, the lack of specific addresses within the provided datasets or alternate parameters to tie a building identifier to an individual database entry prevented the simple joining of these NRCan provided building characteristics and energy datasets to the other Testbed datasets and therefore to their general use within Testbed-18.

Specific issues encountered by Testbed participants while trying to use the above datasets are as follows.

- The .h2k data file format and the HOT2000 data in general were not well documented at the time of Testbed-18. Providing potential users (e.g., geospatial software developers) access to human readable titles and a definition for each attribute as well as information about the possible values (for attributes with enumerated values) or the value ranges and units (for numeric attributes) would be useful. That lack of information combined with the large number of parameters in the HTAP files prevented their use in the Testbed.
- To some extent, the reduced H2K format mitigated the above-mentioned issue with a large number of parameters. The same issues regarding human readable titles and a definition for each attribute as well as information about the possible values (for attributes with enumerated values) or the value ranges and units (for numeric attributes) was still present, however. This information was either not available (definitions, value ranges, enumerated values) or included in an Excel spreadsheet that was not consistently formatted (title, units). As a result, configuring the Quebec Housing Archetype API so that clients could generate meaningful queries or that users could browse in a web browser was unnecessarily complex and time consuming. Value ranges and enumerated values were determined from the values in the existing archetypes.
- In all cases for HOT2000 file data, the sheer number of attributes was overwhelming and it was difficult for someone who was not an expert to decide which were the most important attributes and, in particular, which attributes were most valuable for filtering. Formats such as the reduced H2K format developed by NRCan's CanmetENERGY, which focus on representing statistically important building characteristics for the HOT2000 energy simulation engine, provide a pathway towards a more consumable and FAIR dataset. In addition, the provision of HOT2000 file data in formats that are already used within the North American and global building energy communities, such as EnergyADE, HPXML, or GBXML would facilitate simpler pathways towards interoperability of HOT2000 and EnerGuide for Housing data.
- To help with providing an API for archetypes in order to query archetypes, having information about how users want to query the data would be useful. Such queries could be pre-configured in the API so that they are easy to execute by users. Stakeholder

consultation is needed to determine the type of use cases enabled by the housing archetypes. NRCAN should consider publishing the archetypes containing only the most important attributes for building energy analysis.

- Although NRCAN has developed Canadian archetypes for residential (HTAP) and larger (BTAP) buildings, it would be useful if those archetypes were statistically relevant at a lower geographical scale than the provincial level on which they are already provided. The municipal and/or regional district levels to serve rural areas would be ideal. The archetypes were based on documents and procedures that depend on older references to [StatsCan SHEU](#) and a normalization procedure to determine the statistical distribution across provinces. The ability to access documents pertaining to the process used to arrive at the archetypes is limited and unclear. Also, enabling the archetype's use on a municipal scale would allow stakeholders to conflate municipal and 3D data to the archetypes and work towards accomplishing the Testbed goals. Ideally, all datasets would adhere to FAIR principles and would be kept up to date as the regional housing topology evolves over time.

9.1.2.3. Other specific recommendations and thoughts

- Identifying the building parameters which significantly influence building energy performance (such as lighting power density (W/m²) and equipment power density (W/m²), etc.) which are part of the energyADE standard, are not contained in the energy datasets (other than the in standardized load profiles from the EnerGuide for homes documentation). Are they important? What if more is needed by some stakeholders?
- The building characteristics and energy datasets consider different energy systems at the building level, but there is a lack of energy systems information associated with each building which can be aggregated to larger geographies as required (i.e., distribution network, neighborhood, or city-wide scales). When performing the kind of analysis included within the Testbed goals, the energy system assets and their coordination at geographic levels that encompass many buildings (e.g., community battery storage to smooth “peaky” electricity loads) becomes important and should be considering during planning activities. Stakeholder consultation is needed on this topic to ensure building and utility related planning use cases take a coordinated approach.
- The Testbed work using the provided datasets did not explore future weather or micro-climate weather files which might have a strong effect on the use cases put forward in the Testbed goals.
- The SimStadt simulation is computationally expensive and challenging due to using so many input parameters. There is potential to explore less computational intensive processes to achieve similar goals.
- Validation for the Simstadt building energy simulation engine was not performed within the Testbed due to the lack of available building energy data for individual buildings to compare against the outputs of Simstadt. More broadly, it is recommended that requiring building energy simulation engines, such as virtual energy assessment or audit tools or other new energy calculation approaches, conform to known energy simulation engine

standards such as ASHAE 140 or other accepted validation procedures that ensure their outputs can be trusted by end users.

- Integrating attributes directly within glTF (since attributes extensions likely are not supported by the [assimp library](#) used to export) was not able to be addressed. Other unaddressed challenges included vertical datum challenges (Canadian Geodetic Vertical Datum of 2013 (CGVD2013) vs. European Petroleum Survey Group (EPSG):4379 / CRS84h), importing textures from the original CityGML, and processing and integrating high resolution LiDAR point cloud data from City of Montreal to derive Digital Terrain Models.
- The CityJSON data produced and used within the Testbed was in the compound coordinate reference system NAD83(CSRs)/UTM zone 18N + CGVD2013 height (EPSG:6661), not the native CRS of the dataset NAD83(CSRs)/MTM zone 8 + CGVD28 height (a compound CRS of EPSG:2950 and EPSG:5713), because CityJSON and OGC API Features both required a pre-defined compound EPSG or OGC CRS.
- A recommendation coming out of the Testbed activities is to explore different coordinate reference systems and identify those that are optimal for buildings and energy data.
- To visualize the energy simulation dataset on a web based client a building identifier, i.e., the gml ID of an individual building, is used to combine building geometry from the Building Energy Data Services and simulation results coming from the Building Energy Processing Service. Based on the lessons learnt during the development of Steinbeis Consortium building energy web client – D125, it was found that the property name representing the gml ID may differ from dataset to dataset (e.g., gml_id or gmlID). To overcome this issue, a consistent property naming convention for building identifiers across datasets would enable better interoperability and is recommended.
- There is an opportunity for improvement of:
 - client interoperability of accessing 3D data via OGC Features API, in particular 3D Tiles;
 - versatility of OGC API features to compose visualizations based on different items sources: 3D Tiles, Vector Tiles, and GeoJSON; and
 - classification of energy data client side for styling purposes based on OGC Features API data.
- Improved documentation of workflows and quality control needed in a production environment.

9.2. Practices, collaboration, and communication

The practices, collaboration, and communication used both internally and externally at organizations which provide datasets and/or create guidelines and standards was also an

indirect challenge within the Testbed. The indirect challenge was that the often-siloed nature of these organizations' activities led to semantic barriers in using the datasets and standards together for the purpose of accomplishing the Testbed's goals. More work can be done to bridge silos in government, standards organizations, and industry to enable building related data interoperability. This can be accomplished through stakeholder consultation activities to present available data and identify missing data within relevant datasets. The following subsections contain more information, general and specific challenges, and lessons learned related to practices, collaboration, and communication among building energy and geospatial data stakeholders.

9.2.1. General challenges

9.2.1.1. Governments

Governments at all levels within Canada often both provide open data but also inadvertently act as barriers to its use. The participants in the Testbeds noted a number of challenges and lessons that can be learned to enable more interoperable datasets for building energy and geospatial data.

- More technical guidance from government entities is needed on how to use the open data they provide. A good example of this within the Testbed was for the building characteristics and energy data provided by NRCan. More documentation would have reduced barriers to implementation.
- There needs to be legal means established within and by government to share protected or private data with qualified organizations such as utility usage or home energy ratings at the address level. These tools and methods exist in some government departments and within the healthcare industry so there is precedent for their use in other areas of government. The Testbed participants were able to illustrate the problem through the Testbed activities. Privacy and accessibility of EnerGuide (and HTAP) data for specific addresses was a recurring issue in the Testbed and was a significant barrier in conducting the Testbed experiments. The participants were unable to conflate the provided building energy data for the unique building identifiers inside the chosen OGC representations because they were provided at the Forward Sortation Area (FSA) scale and, lacking a civic address, could not be tied to a specific dwelling and therefore could not serve or display that data within the Testbed.
- More connectivity between government departments working on the energy mapping techniques is needed to enable a whole of government approach to the issue of buildings and energy geospatial data interoperability. This is closely tied to the semantics issue discussed in the previous section and a top-down look at semantics would be a good fit for the higher levels of government to collaborate on with the standards setting communities – including OGC. Connectivity on semantics and the successes that it can yield was shown in the Testbed in response to the inability to use the provided building energy data. The Testbed participants used building characteristics datasets available through the City of Montreal and the agglomeration of Montreal to populate CityGML and CityJSON representations. Those data representations with building characteristics for each unique identifier were able to be processed by an OGC compliant API implementation of the

Simstadt energy simulation engine and have energy consumption estimates assigned to them. This is an example of how standards compliant datasets enabled the Testbed goals through adherence to FAIR principles and by prioritizing interoperability.

- Drawing upon the above comment, governments should use the same semantics and terminology across municipal, provincial, and federal datasets. Stakeholder engagement activities should be used to determine how to arrive at those semantics. Lower levels of government (municipal level) need federal coordination and standards on how to provide 3D and other building data including building permit and property assessment data. All levels of government should work together towards offering 3D building data in the same formats (e.g., CityJSON or CityGML rather than many different types of data). Placing all of those datasets on one portal or national building layer would facilitate their use and standards adoption.
- To start to facilitate the use of Canadian housing data for geospatial and building energy analyses, it would help stakeholders to have access to a set of building characteristics and energy data in formats that are interoperable and FAIR. That need could be resolved by creating scripts to allow the translation of large datasets of proprietary EnerGuide data or HOT2000 archetype data to formats like CityGML energyADE, CityJSON, or simply to JSON or CSV formats. Documentation should also be provided for the user to help interpret the data.
- To enable buildings data aggregation to any scale, it is recommended to attribute buildings and/or parcels with:
 - hierarchies of building types/functions;
 - year of construction & vintages; and
 - geographies (e.g., census tract, neighborhood, FSA etc.).
- To socialize the Testbed concept and findings, government staff, academics, and consultants should receive training on creating 3D volumes and building characteristics datasets conforming to priority standards from available datasets.
- Government entities in Canada should continue to share and build towards an Energy Spatial Data Infrastructure (ESDI).
- The examples of whole of government approaches that can be looked to for ideas and inspiration include [Japan's PLATEAU project](#), which showed how Japan's Ministry of Lands, Infrastructure, Transport, and Tourism collaborated directly with municipalities across the country to train them in CityGML Urban Planning ADE to create standardized data for dozens of use cases. The United Kingdom's [Ordnance Survey](#) conducted extensive stakeholder consultation and development of the Data Hub, which includes buildings and attributes including type, age, number of floors, presence of basement, and roof attributes including shape, aspect, and presence of solar panels and green roofs.

9.2.1.2. Standards development organizations

The following are some standards development related lessons observed during the Testbed:

- Standards organizations generally work in their own specific field and partnership is often limited to co-developed standards. In an environment like the Testbed where multiple technical disciplines work together (geospatial and building science), the potential for multiple standards organizations to bring their knowledge to bear on a problem is clear. One aspect where this was obvious in the Testbed was semantics: nomenclature and geographic boundaries. There is an opportunity for disparate standards organizations to come together in an environment like Testbeds to collaborate on semantics, notation, and terminology within standards to further FAIR data principles <https://www.go-fair.org/fair-principles/>.
- Standards organizations should strive to educate stakeholders outside of their direct user base to those organizations that are served by their users. Standards uptake is tied to user organizations understanding the value of standards and requiring or encouraging their uptake. In the case of Testbed-18 and the building energy and geospatial interoperability exercise, there are ample standards offered that could facilitate interoperability – uptake and adoption is key. The Testbed participants demonstrated through their activities that current OGC standards could enable the Testbed goals.

9.3. Examples of integration potential and selected data models/standards

9.3.1. HTAP and Energy ADE data format integration

As an example of the integration between the HTAP database and CityGML Energy ADE, the Housing Technology Assessment Platform (HTAP) derived archetypes for Kelowna were mapped to the building physics and occupancy library based on the CityGML Energy Application Domain Extension (Energy ADE) specification. Following the main modules of the Energy ADE, the parameters and indicators were extracted from 24 archetypes (in H2K format) based on Kelowna (BC, Canada) energy audits provided by NRCan. The drawn-out data was utilized to organize the prototype's energy data and adjust the data in an operable XML format according to the Energy ADE specification. Two XML data libraries were provided for the building physics of 24 archetypes and the occupancy library for the single-family dwellings.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<buildingPhysicsLibrary xmlns="http://www.simstadt.eu/BuildingPhysicsLibraries">
  <name>Kelowna Building Typology Library </name>
  <description>This library is provided by Concordia University to cover low-rise housing archetype in Kelowna to support CEEMAP project ,Ca
  <source>The Canada Excellence Research Chairs</source>
  <valid>true</valid>
  <protected>false</protected>
  <windowLib>
    <name>Window library from Kelowna Building Typology Library </name>
    <description>Window library from Kelowna Building Typology Library </description>
    <source>The Canada Excellence Research Chairs</source>
    <windowType>
      <id>0</id>
      <name>Double-glazed window with plastic frame</name>
      <uValue>2.57</uValue>
      <gValue>0.76</gValue>
      <glazingNumber>2</glazingNumber>
      <frameRatio>0.3</frameRatio>
      <embodiedEnergy>577.53</embodiedEnergy>
      <embodiedCarbon>106.92</embodiedCarbon>
      <constructionDescription>double glazing with plastic frame, produced in Germany, transportation 50 km </constructionDescription>
      <disposalEnergy>4.48</disposalEnergy>
      <disposalCarbon>2.26</disposalCarbon>
      <disposalDescription>100% sent to landfill, transportation 50 km</disposalDescription>
    </windowType>
    <windowType>
      <id>1</id>
      <name>Low-E double-glazed window_1,6</name>
      <uValue>1.6</uValue>
      <gValue>0.63</gValue>
      <glazingNumber>2</glazingNumber>
      <frameRatio>0.3</frameRatio>
      <embodiedEnergy>628.89</embodiedEnergy>
      <embodiedCarbon>81.57</embodiedCarbon>
      <constructionDescription>double glazing with wooden frame, produced in Germany, transportation 50 km </constructionDescription>
      <disposalEnergy>3.97</disposalEnergy>
      <disposalCarbon>4.48</disposalCarbon>
      <disposalDescription>100% sent to landfill, transportation 50 km</disposalDescription>
    </windowType>
    <windowType>
      <id>2</id>
      <name>Low-E double-glazed window_1,3</name>
      <uValue>1.3</uValue>
      <gValue>0.6</gValue>
      <glazingNumber>2</glazingNumber>
      <frameRatio>0.3</frameRatio>
      <embodiedEnergy>628.89</embodiedEnergy>
      <embodiedCarbon>81.57</embodiedCarbon>
      <constructionDescription>double glazing with wooden frame, produced in Germany, transportation 50 km </constructionDescription>
      <disposalEnergy>3.97</disposalEnergy>
      <disposalCarbon>4.48</disposalCarbon>
      <disposalDescription>100% sent to landfill, transportation 50 km</disposalDescription>
    </windowType>
  </windowLib>
</buildingPhysicsLibrary>

```

Figure 70

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<buildingUsageLibrary xmlns="http://www.simstadt.eu/BuildingUsageLibraries_v2">
  <name>residential</name>
  <description>This library is provided by Concordia University to cover low-rise housing archetype in Kelowna to support CEE
</description>
  <source>Canada Excellence Research Chairs</source>
  <valid>true</valid>
  <protected>>false</protected>
  <mappingType>Alkis</mappingType>
  <zoneUsageType>
    <id>residential_SFD_detached</id>
    <description>SFD</description>
    <occupancy>
      <occupancyDensity>0.01</occupancyDensity>
      <usageDaysPerYear>365</usageDaysPerYear>
      <usageHoursPerDay>14</usageHoursPerDay>
      <occupancyRate>
        <description>Daily schedule extracted from ecobee dataset</description>
        <weekDayProfile>
          <source>ecobee</source>
          <values>0.89 0.89 0.89 0.89 0.89 0.89 0.89 0.89 0.89 0.43 0.38 0.38 0.38 0.39 0.4 0.42 0.43 0.45 0.46 0.48 0.48
        </weekDayProfile>
        <weekEndProfile>
          <source>ecobee</source>
          <values>0.86 0.86 0.86 0.86 0.86 0.86 0.86 0.86 0.57 0.58 0.56 0.53 0.52 0.53 0.53 0.5 0.49 0.47 0.47 0.46
        </weekEndProfile>
      </occupancyRate>
    </occupancy>
    <internGains>
      <description>people and appliance</description>
      <averageInternGainPerSqm>4.2</averageInternGainPerSqm>
      <convectiveFraction>0.3999999999999999</convectiveFraction>
      <radiantFraction>0.5000000000000001</radiantFraction>
      <latentFraction>0.10000000000000009</latentFraction>
    </internGains>
  </occupancy>
</endUses>
  <space_heating>
    <heatingSetPointTemperature>21.0</heatingSetPointTemperature>
    <heatingSetBackTemperature>18.0</heatingSetBackTemperature>
    <schedule>
      <description>Daily schedule extracted from ecobee dataset</description>
      <weekDayProfile>
        <source>ecobee</source>
        <values>18.1 18.1 18.1 18.1 18.1 18.3 19.0 19.5 19.2 19.2 19.1 19.0 19.0 19.0 19.1 19.3 19.7 19.7
      </weekDayProfile>
      <weekEndProfile>
        <source>ecobee</source>
        <values>18.1 18.1 18.1 18.1 18.1 18.3 19.0 19.5 19.2 19.2 19.1 19.0 19.0 19.0 19.1 19.3 19.7 19.7
      </weekEndProfile>
    </schedule>
  </space_heating>

```

Figure 71

In the developed data model for the Kelowna location, the construction class is based on the information available in the .H2K format of the prototypes. The shortcoming of construction assemblies' properties is that it is provided using the average method (not specific to each wall). The physical and construction attributes provided for each archetype were integrated into the XML format.

The occupant-related input data includes occupancy density, usage days per year, usage hours per day, internal gains (average internal gains, convective fraction, latent fraction, radiant fraction), space heating and cooling (setpoint and setback temperature), domestic hot water (average consumption, preparation temperature), and electrical appliance (average consumption). The occupant-related data was extracted from the EnerGuide occupant schedule in the archetype H2K files and edited to an XML file using an editor.

10

FUTURE WORK

The following section contains some participant-identified future work needs.

10.1. Automated workflow

The use of APIs based on standardized OGC API building blocks in the various components should enable fully automated workflows where the processing service accesses the buildings, building archetypes, observed data for buildings, weather data, etc. from APIs on demand to generate the simulated energy information for each building. This information could then be downloaded or again shared via an API for use by applications. The simulated energy data includes the building identifier, so that the simulated data can be joined with the buildings.

There was insufficient time in Testbed 18 to properly implement such a fully automated workflow. The Building Energy Processing Service already uses CityGML building data as input, but this currently requires that the data is locally available (the Nuns' Island building dataset was used in the testbed).

How a fully automated workflow can be achieved should be investigated. There are several aspects to consider and analyze.

10.1.1. Access patterns

To simulate the energy demand of a building, the simulation engine needs information about the building, geometric / physical information about the neighboring buildings, and other structures or vegetation. This implies that buildings and other geospatial data must be streamed by the APIs so that features that are close to each other are returned together. Typically, some form of tiling will be used to achieve this.

10.1.2. Data formats

Data formats such as 3D Tiles and Mapbox Vector Tiles are by design tiled data structures. They are also designed for delivering data in an efficient way. However, the priority is efficiency for visualizing the data in a map or globe – at various levels of detail. As a result the same building may be included in several zoom levels and with varying geometry information. How to find the best building representation of a building or its neighbors in a tileset is unclear.

Another limiting factor for these formats are restrictions on the representation of attribute information.

- Mapbox Vector Tiles only support attributes with basic types (strings, numbers, boolean), no arrays or objects.

- 3D Tiles 1.1 uses the glTF-Extension [EXT_structural_metadata](#), which primarily supports such basic types as well, but in addition supports numerical arrays. Such arrays could be used, for example, for estimated monthly heating demands from January to December.

Alternatives are feature data formats that support building data / solids. Based on the results of the tests with various feature formats for the Montreal dataset, CityJSON seems to be the most promising data format for CityGML building data and FlatGeobuf for 2D building data where only attributes with basic types are needed.

Access to such feature data could also occur using the OGC API – Tiles conformance class (building blocks) or using the OGC API – Features building blocks (and the client / processing service accesses the data in tiles, the same approach that web mapping libraries typically use when accessing feature data).

10.2. Commercial building datasets and archetypes

- Within the Testbed, the energy simulation engine, Simstadt, used non-Canadian archetypes from other jurisdictions to perform its analysis. Future work could include using appropriately designed HTAP (low rise houses) and BTAP (high rise and commercial/ industrial) formats to provide the simulation archetypes for the appropriate building topologies.
- The Testbed was unable to include utility provided or EnergyStar Portfolio Manager data into the experiments. This would be important future work to enable the utility capacity analysis aspect of the Testbed goals and advancing building rating and labeling.

10.3. Acquiring new data sources

- There is an opportunity to study how feature recognition through mass imaging techniques (LiDAR, satellite imaging, photogrammetry, drone image capture, and other techniques) combined with image recognition and machine learning algorithms can be used to acquire attributes for 3D representations of buildings and allow broader use of workflows that were demonstrated in the Testbed. The broader availability of building characteristics (including 3D) for all buildings could lead to workflows in the future that meet the Testbed goals.
- Future investigations could include working with building permit data and plans. These are the ultimate source of property assessment roll data, linked with municipal Building Code administrations that process the on-the-ground administration of provincially legislated and model national building codes. Their provision as open data would be of interest to stakeholders because their contents have been verified by a human as part of the process and are kept up to date by necessity.

10.4. Processing of datasets

- Data access and rendering of 3D data client side: how could we speed up performance when a big dataset is being used?
- Static data and dynamic data: how could we improve the relation between static data as 3D building and dynamic data such as energy values keeping good performances client side? Is it possible to introduce relations between static/dynamic datasets?
- Can the streaming performance of building energy simulations be improved? For example, [work on surrogate modeling](#) provides an instructive example of how the speed of streaming energy models can be improved (within known boundary conditions) while maintaining good accuracy.

10.5. Connection to other existing tools

- The Community Technology Assessment Platform (CTAP) is an Excel spreadsheet based tool developed by CanmetENERGY-Ottawa's Buildings and Renewables Group and built upon TRNSYS archetype models in order to support analysis of community energy technologies. CTAP operates at a higher spatial scale (FSA) but lower temporal scale for equipment analysis than the work completed within the Testbed.
- Geospatial databases could be accessed by scripts or tools that output summary statistics which could then be entered into spreadsheet based tools such as CTAP. The opportunity for future work in this case is to explore how standards-based datasets could be analyzed and used to inform spreadsheet based analysis by non-geospatial professionals such as engineers.



11

CONCLUSIONS

CONCLUSIONS

The activities conducted within the Building Energy Spatial Data Interoperability task of Testbed 18 demonstrated some of the intended work from the [OGC Testbed-18: Call for Participation \(CFP\) document](#). The participants accomplished this work through the investigative tracks discussed within this ER. The following subsections discuss the Testbed's success in achieving its initial goals.

11.1. General goals

Per the NRCan guidance, the goal of the Testbed 18 documented in this ER task was to prototype an Energy Spatial Data Infrastructure (ESDI) that could become part of the CGDI and allow for execution of building energy experiments and analysis – ideally using existing OGC Web API standards. The participants were able to accomplish all of the following goals:

- leveraging existing building energy datasets to evaluate specific energy and geospatial interoperability requirements;
- exploring draft data models and associated schemas for key building energy datasets;
- demonstrating the potential of existing geospatial standards to support access and use of building energy data, and their interoperability with other geospatial information; and
- demonstrating the potential of existing or new OGC standards to support building energy data interoperability and recommending future standards development activities to fully implement an Energy SDI.

However, these goals were not completed in full for all of the intended data sources. It was clear from the Testbed activities that OGC standards compliance enabled use of some datasets (such as the CityGML data from Montreal) and lack of compliance and documentation hindered the use of others (NRCan provided building energy datasets).

11.2. Data model goals

At the outset of the Testbed, the development of a generalized data model was anticipated based on the following potential scenarios:

- extracting information from building energy models for map attribution or conversely; and
- extracting available attributes from a map for a number of dwellings in the stock to feed into building energy simulations.

Of those two goals, the latter was met by the Testbed and potential was shown in the challenges and lessons learned from the Testbed on how the other goal could be met in the future.

11.3. Detailed use case goals

The Testbed's targeted stakeholders had a need for understanding of the building stock – the age, number of stories, floor areas, and other characteristics of various building archetypes and how they impact energy usage towards informing programs and policies. Of the previously mentioned needs, the data models worked on within Testbed 18 intended to enable the following use cases.

- Housing Retrofit Program Planning
- Utility Hosting Capacity Analysis

Through the adaptation of existing OGC Standards, in particular OGC Web APIs, to the available energy and geospatial data, Testbed participants were able to explore the Housing Retrofit Program Planning and Provision of building stock statistics use case goals in support of contributing towards a Canadian Energy SDI. The Testbed participants were not able to address the Utility Hosting Capacity Analysis use case within the Testbed due to time and data constraints. The work done in the Testbed can be considered foundational for mapping buildings and modeled energy data, to which electricity distribution topology and loads could be added later. In the future, this could bring energy efficiency opportunities to hosting capacity assessment, augmenting what is typically a load analysis activity.

11.4. Final Notes

Testbed 18 was able to show promise in the application of standards to the problem of geospatial, building characteristics, and building energy data interoperability for the use cases put forward in the Testbed. The Testbed also demonstrated that standardized Web APIs foster an innovative and open environment where software providers from different domains (e.g., geospatial, building energy) developed modular software services and client applications that exchanged data and formed parts of large and feature rich applications. This standardized, modular application format encourages economic development and innovation by allowing smaller, often subject matter expert developers, to interact with a vast ecosystem of other stakeholders to create innovative and useful software platforms and grow their technical and organizational capacities. This was shown in the Testbed where organizations drew upon their strengths and contributed modularized components to the full stack of services and clients needed to reach the Testbed 18 goals. The standardized, modular application format also provides the ability to isolate and verify the components of larger software platforms allowing easier process validation and leading to more accuracy and transparency within the vertical

geospatial and building energy software stack. The Testbed therefore formed a foundation for future investigation into the converging fields of geospatial and building energy analyses.

A

ANNEX A (INFORMATIVE) REVISION HISTORY



ANNEX A (INFORMATIVE) REVISION HISTORY

DATE	RELEASE	AUTHOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2023-02-09	1	L. St. Hilaire	all	initial version
2023-04-28	2	L. St. Hilaire	all	DWG review version
2023-06-30	3	L. St. Hilaire	all	final version



BIBLIOGRAPHY





BIBLIOGRAPHY

- [1] 2022 Global Status Report for Buildings and Construction. <https://www.unep.org/resources/publication/2022-global-status-report-buildings-and-construction>
- [2] CO2 emissions from buildings and construction hit new high, leaving sector off track to decarbonize by 2050: UN. <https://www.unep.org/news-and-stories/press-release/co2-emissions-buildings-and-construction-hit-new-high-leaving-sector>
- [3] The Canada Green Buildings Strategy. <https://natural-resources.canada.ca/public-consultations-and-engagements/the-canada-green-buildings-strategy/25009>
- [4] Green buildings. <https://natural-resources.canada.ca/energy-efficiency/green-buildings/24572>
- [5] Frequently Asked Questions (FAQs) – U.S. Energy Information Administration (EIA). <https://www.eia.gov/tools/faqs/faq.php?id=86&t=1>
- [6] Advanced Hosting Capacity Analysis. <https://www.nrel.gov/solar/market-research-analysis/advanced-hosting-capacity-analysis.html>