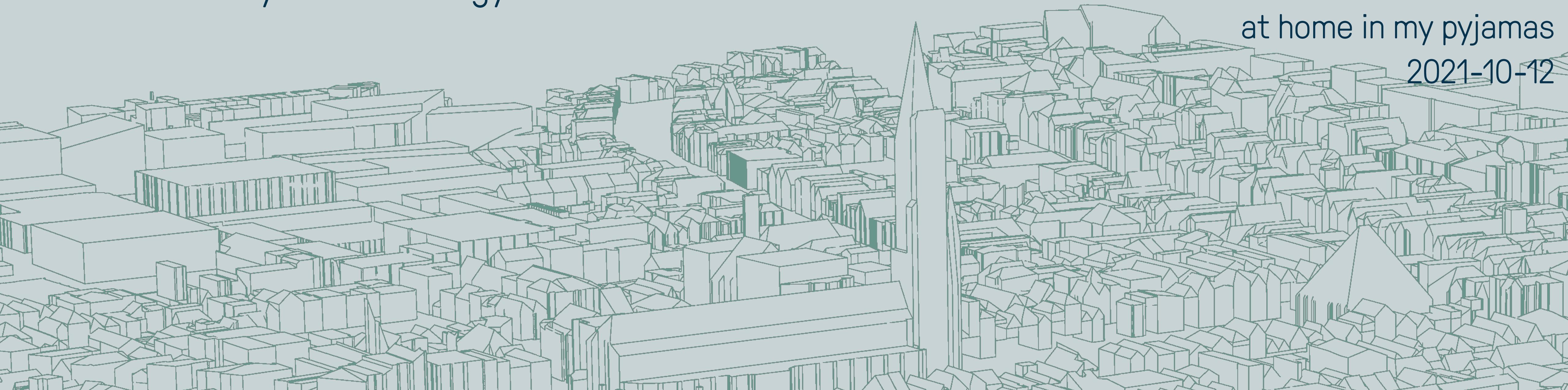


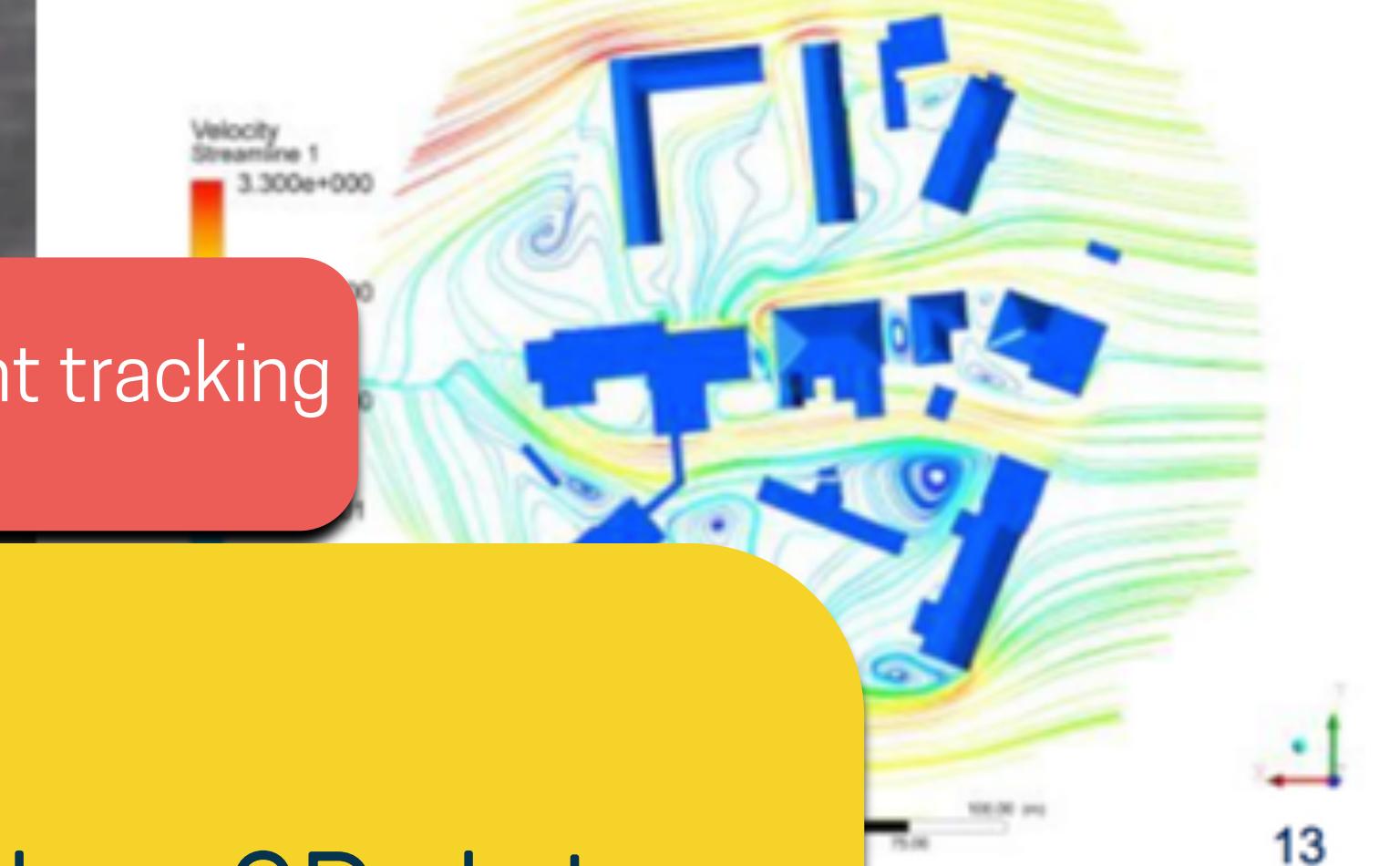
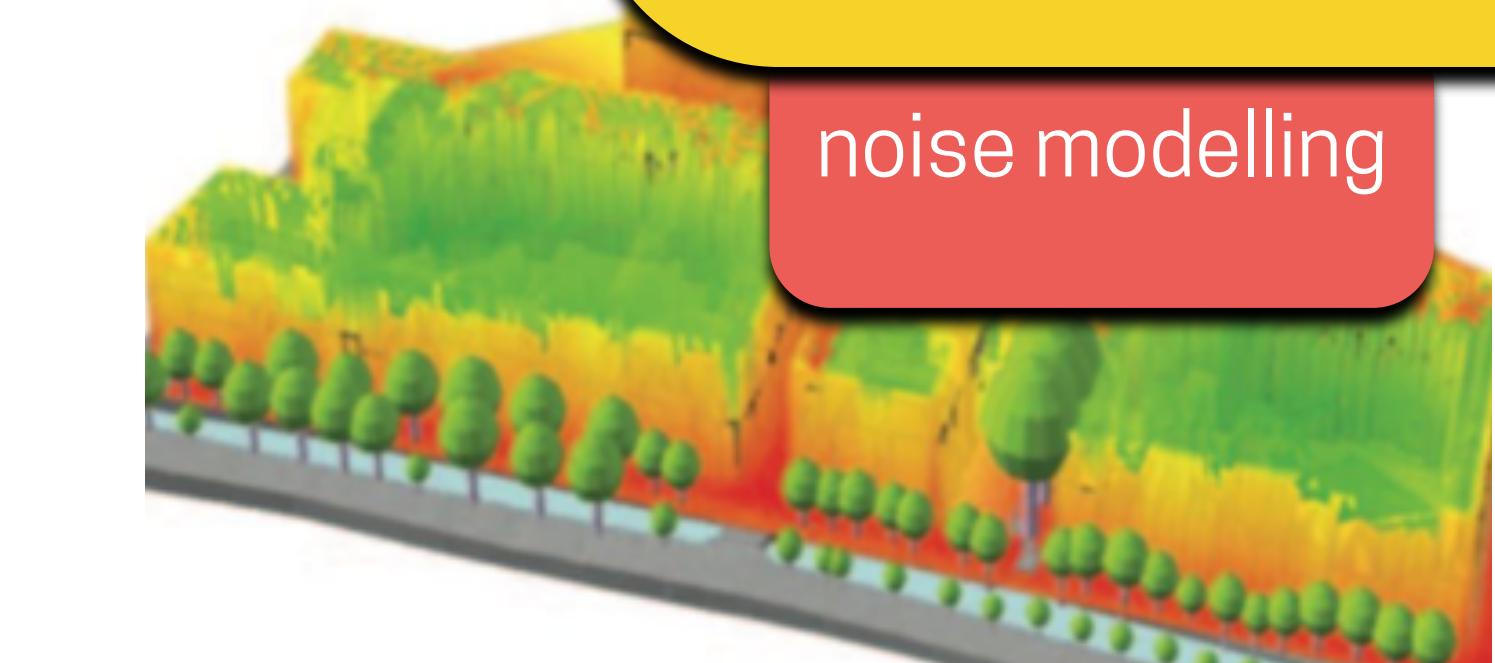
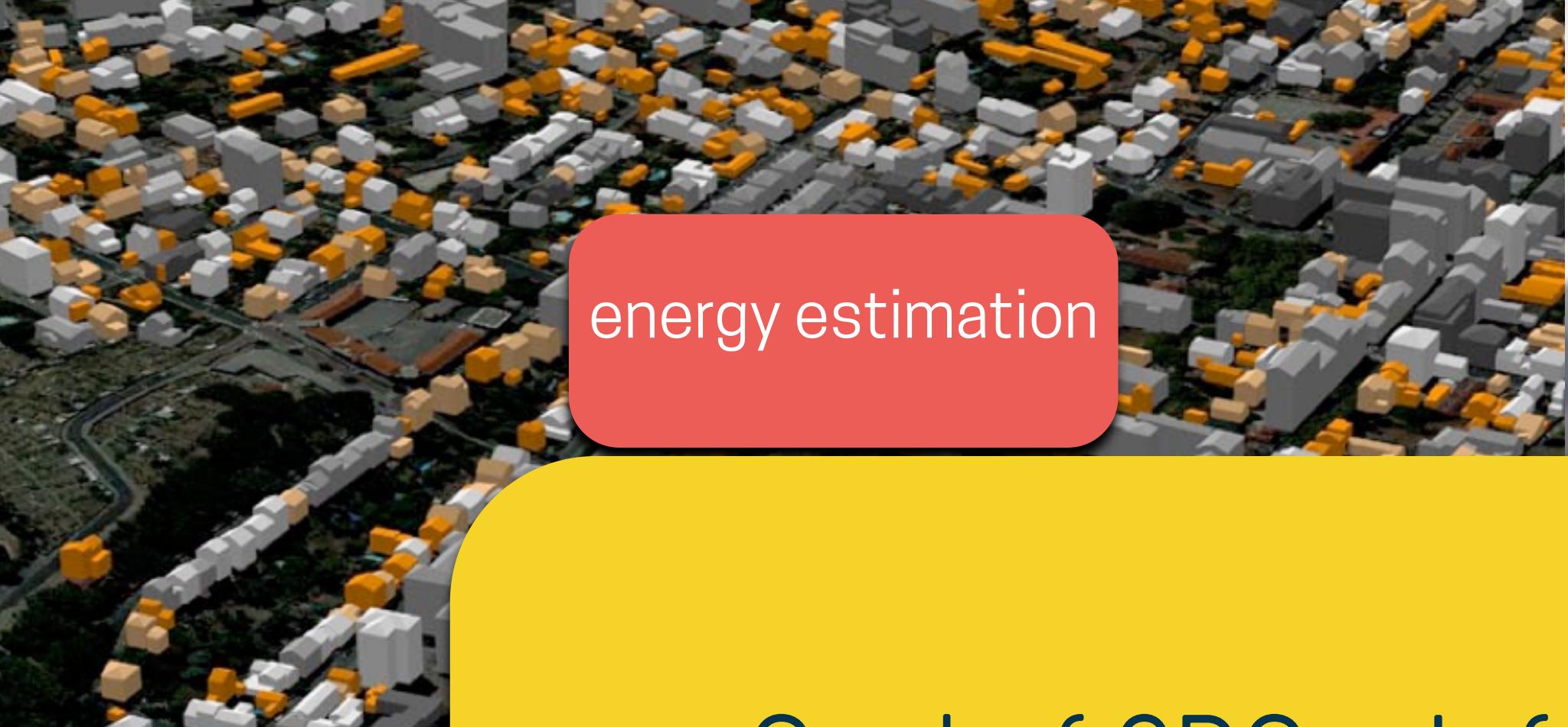
# Maybe it's time to do something with all those 3D city models?

**Hugo Ledoux**

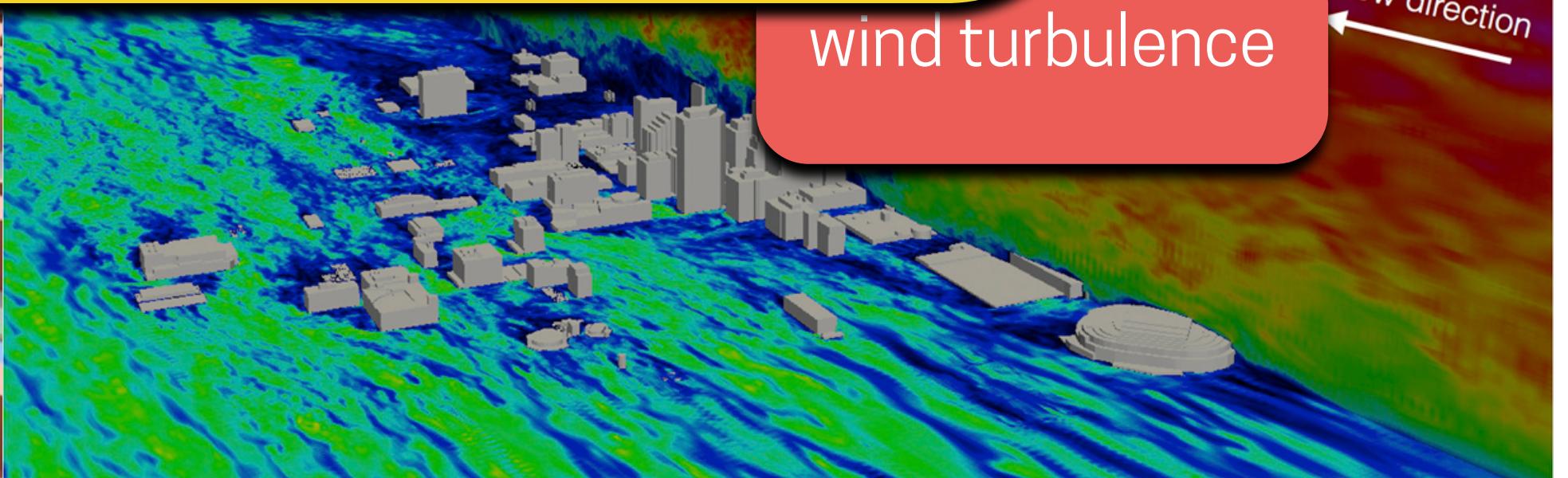
Delft University of Technology

3D GeolInfo 2021  
at home in my pyjamas  
2021-10-12



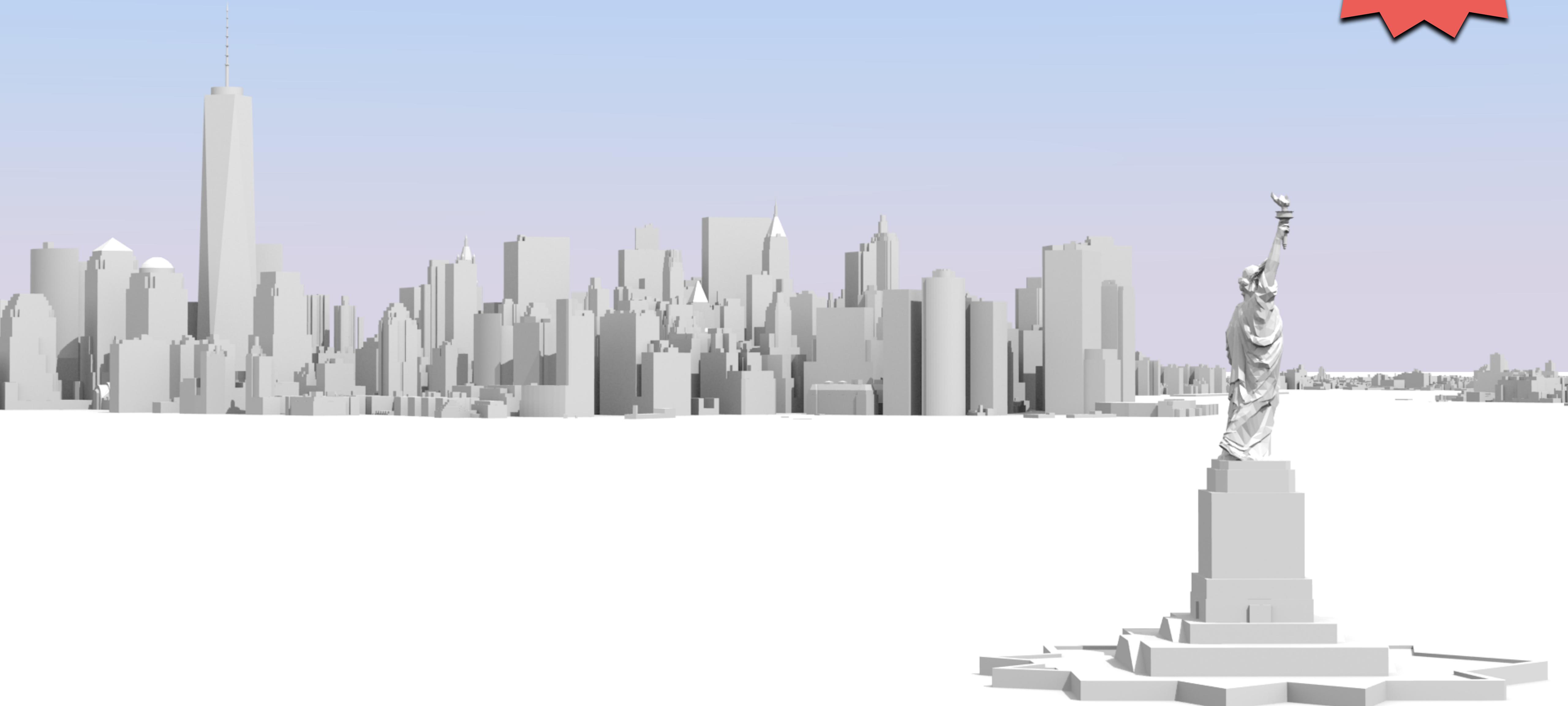


- Goal of 3DGeoInfo conferences: push further 3D data modelling
- So that 3D datasets are usable and interchangeable
-  is central as the semantic data model



# More and more 3D (open) data from cities

New York City  
(LoD2)

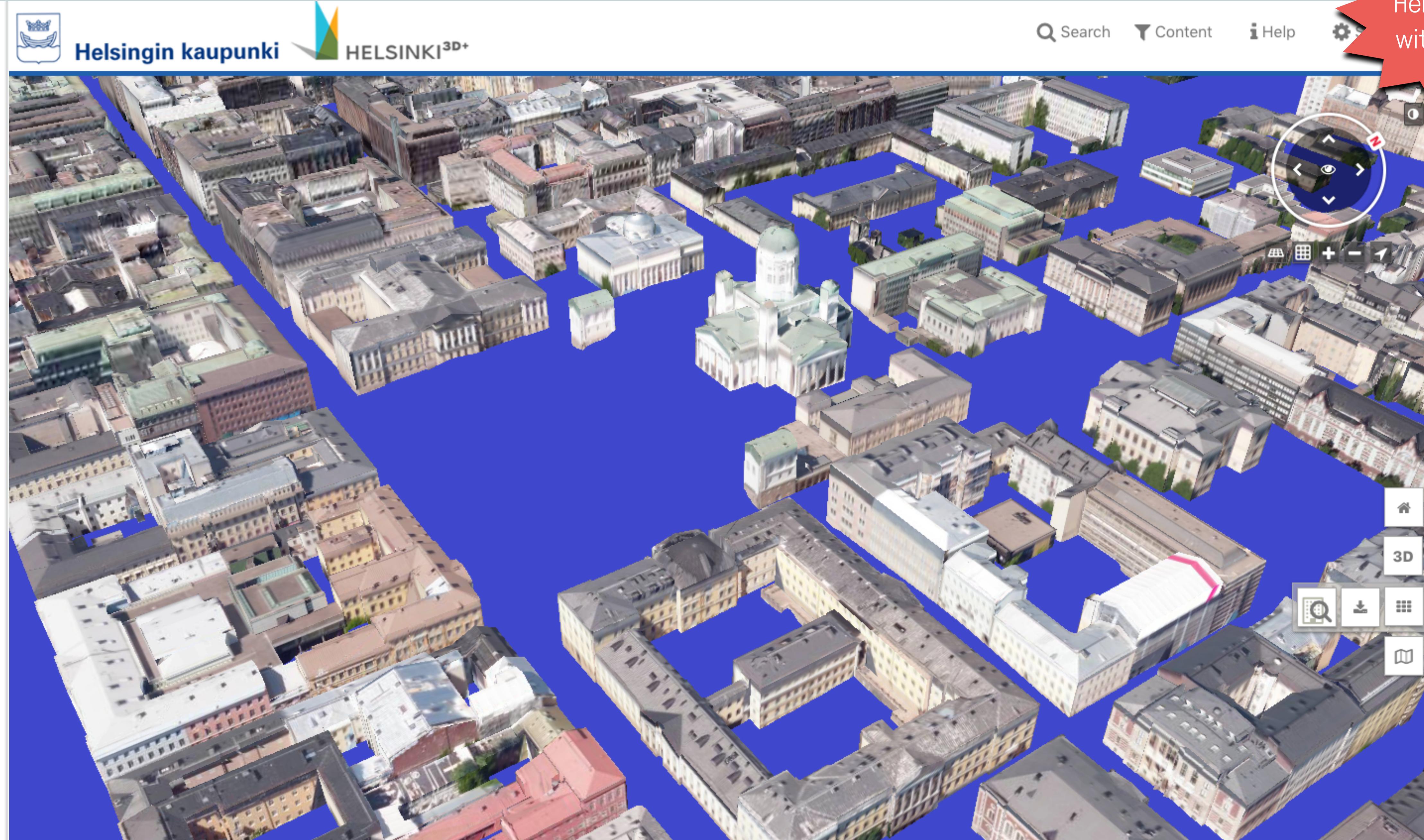


# More and more 3D (open) data from cities

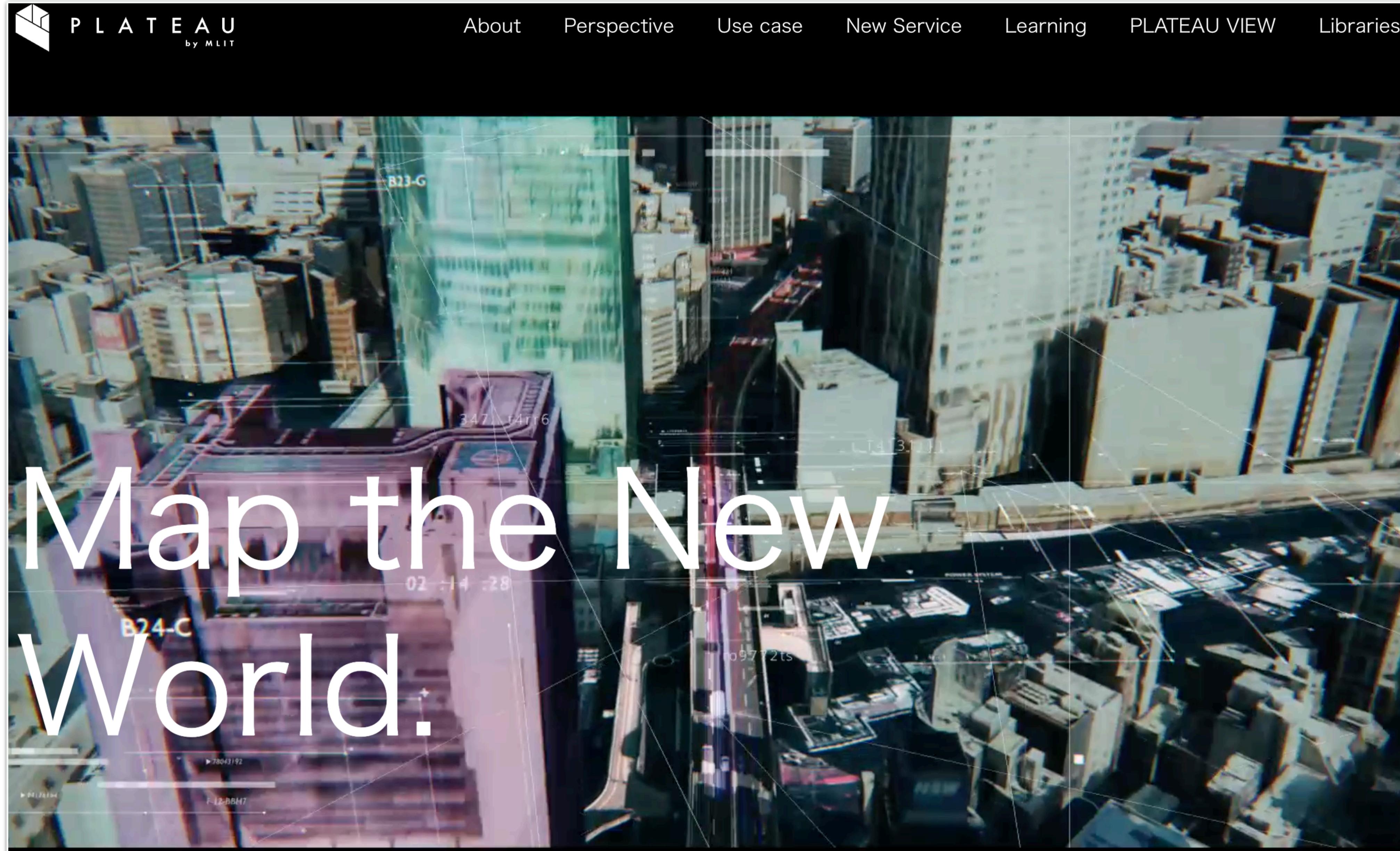
The screenshot shows a web browser window for the "Maps of Switzerland - Swiss Coat of Arms" website. The URL is <https://map.geo.admin.ch/?lang=en&topic=ech&bgLayer=ch.swisstopo.pixelkarte-farbe>. The page features a search bar at the top with the placeholder "e.g. Bundesplatz 1 Bern, 46.7 7.5, Noise map ...". To the right of the search bar are links for "Try out test.map.geo.admin.ch", "Full screen", "Report problem", "Help", "Mobile version", and language options (DE FR IT EN RM). On the left side, there is a sidebar with links for "Share", "Advanced tools", "Geocatalog" (with a "Change topic" button), and "Maps displayed". A "Close menu" button is located at the bottom of the sidebar. The main content area displays a detailed 3D map of a Swiss town, showing buildings with red roofs, green trees, and a river. A "3D" button is visible in the bottom right corner of the map area. The map is set against a background of a forested hillside.

Switzerland: 70M 3D objects are now open

# More and more 3D (open) data from cities



# More and more 3D (open) data from cities

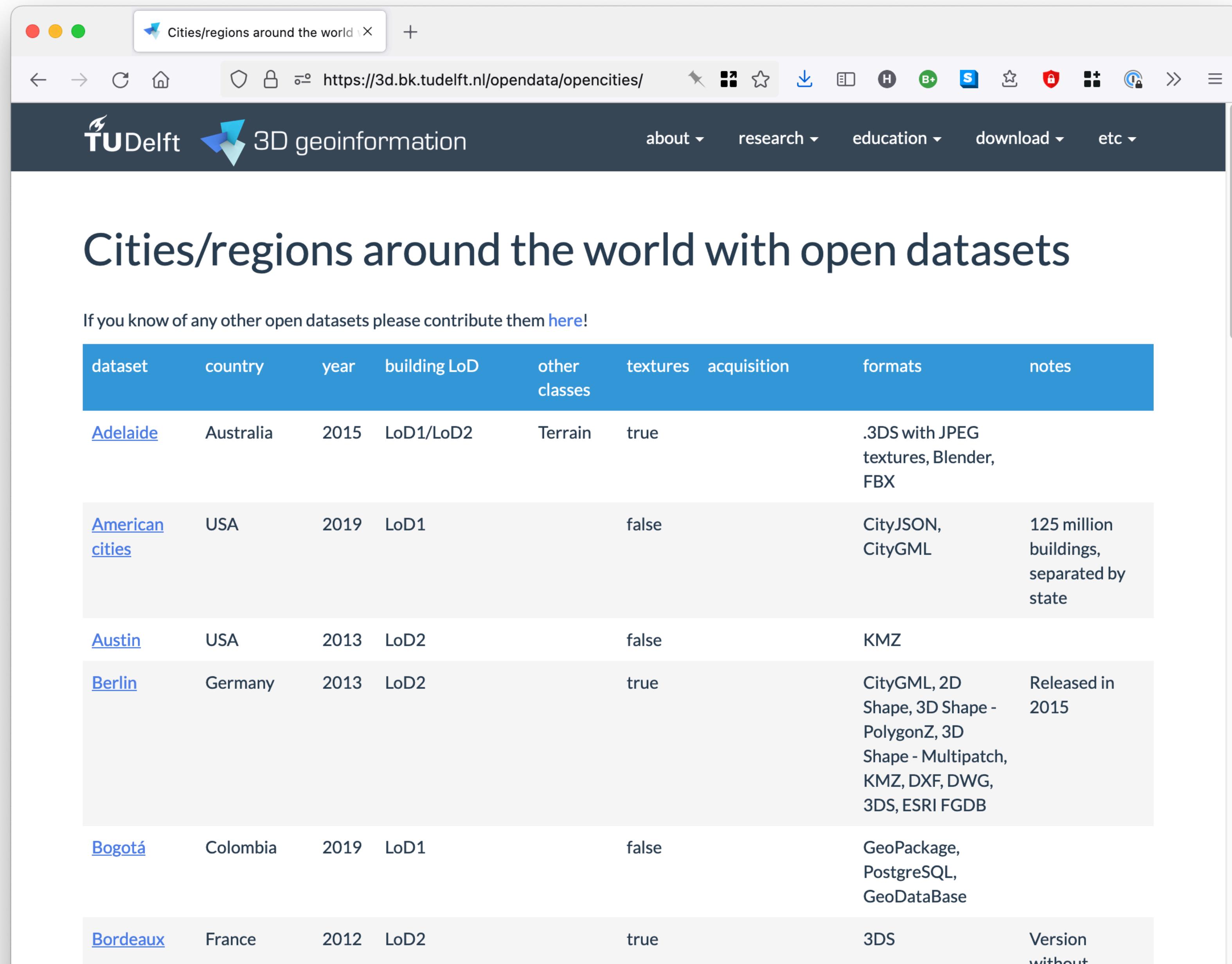


56 cities in  
Japan



About Perspective Use case New Service Learning PLATEAU VIEW Libraries

# More and more 3D (open) data from cities/countries



The screenshot shows a web browser window with the title "Cities/regions around the world" and the URL "https://3d.bk.tudelft.nl/opendata/opencities/". The page is titled "3D geoinformation" and features a navigation menu with links to "about", "research", "education", "download", and "etc". Below the menu, the main content is titled "Cities/regions around the world with open datasets". A callout box on the right side of the page contains the text "41 cities/countries!". The table lists the following data:

dataset	country	year	building LoD	other classes	textures	acquisition	formats	notes
<a href="#">Adelaide</a>	Australia	2015	LoD1/LoD2	Terrain	true		.3DS with JPEG textures, Blender, FBX	
<a href="#">American cities</a>	USA	2019	LoD1		false		CityJSON, CityGML	125 million buildings, separated by state
<a href="#">Austin</a>	USA	2013	LoD2		false		KMZ	
<a href="#">Berlin</a>	Germany	2013	LoD2		true		CityGML, 2D Shape, 3D Shape - PolygonZ, 3D Shape - Multipatch, KMZ, DXF, DWG, 3DS, ESRI FGDB	Released in 2015
<a href="#">Bogotá</a>	Colombia	2019	LoD1		false		GeoPackage, PostgreSQL, GeoDataBase	
<a href="#">Bordeaux</a>	France	2012	LoD2		true		3DS	Version without

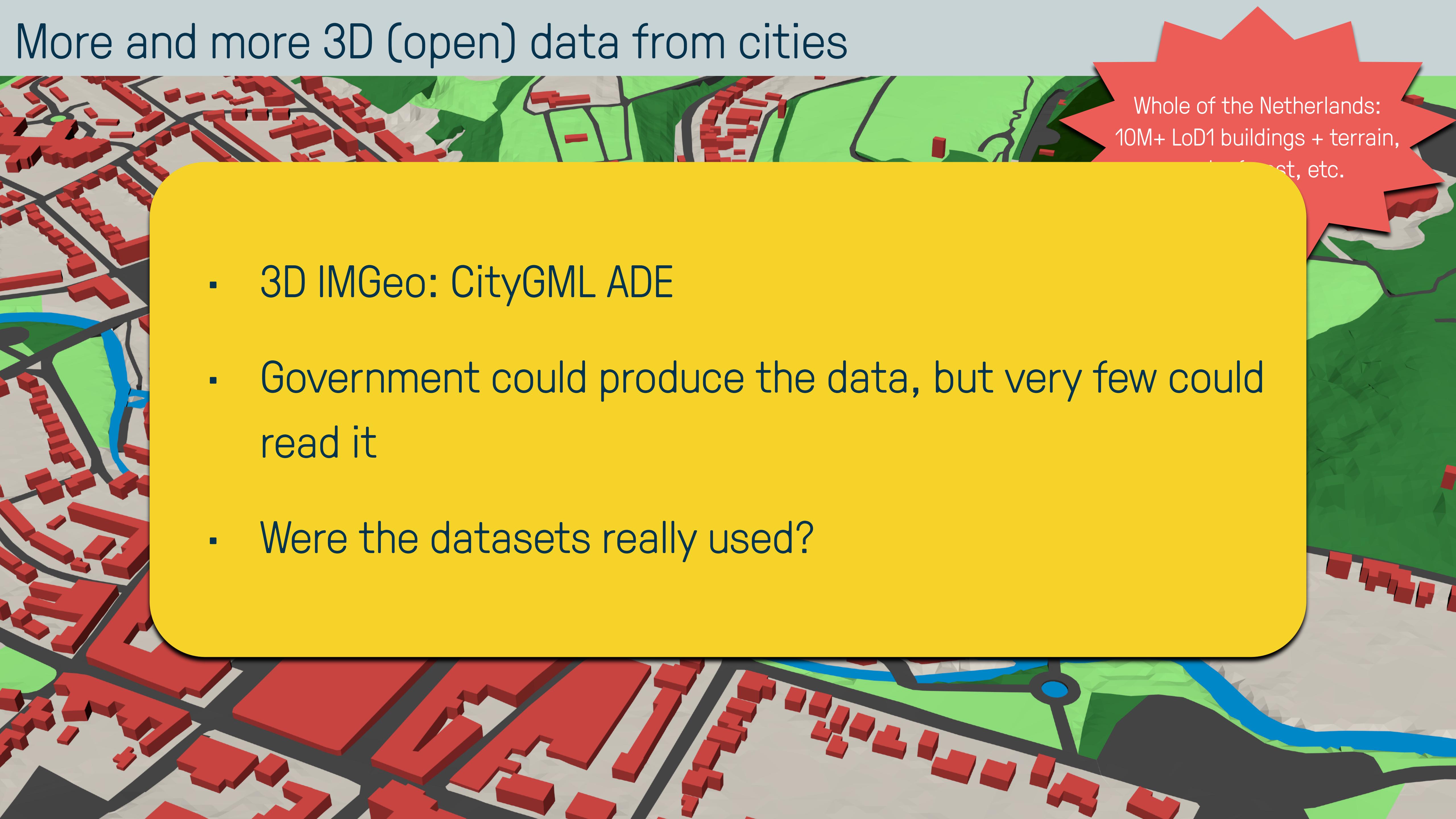
41 cities/countries!

# So, mission accomplished!?



**Yeah, not  
so sure...**

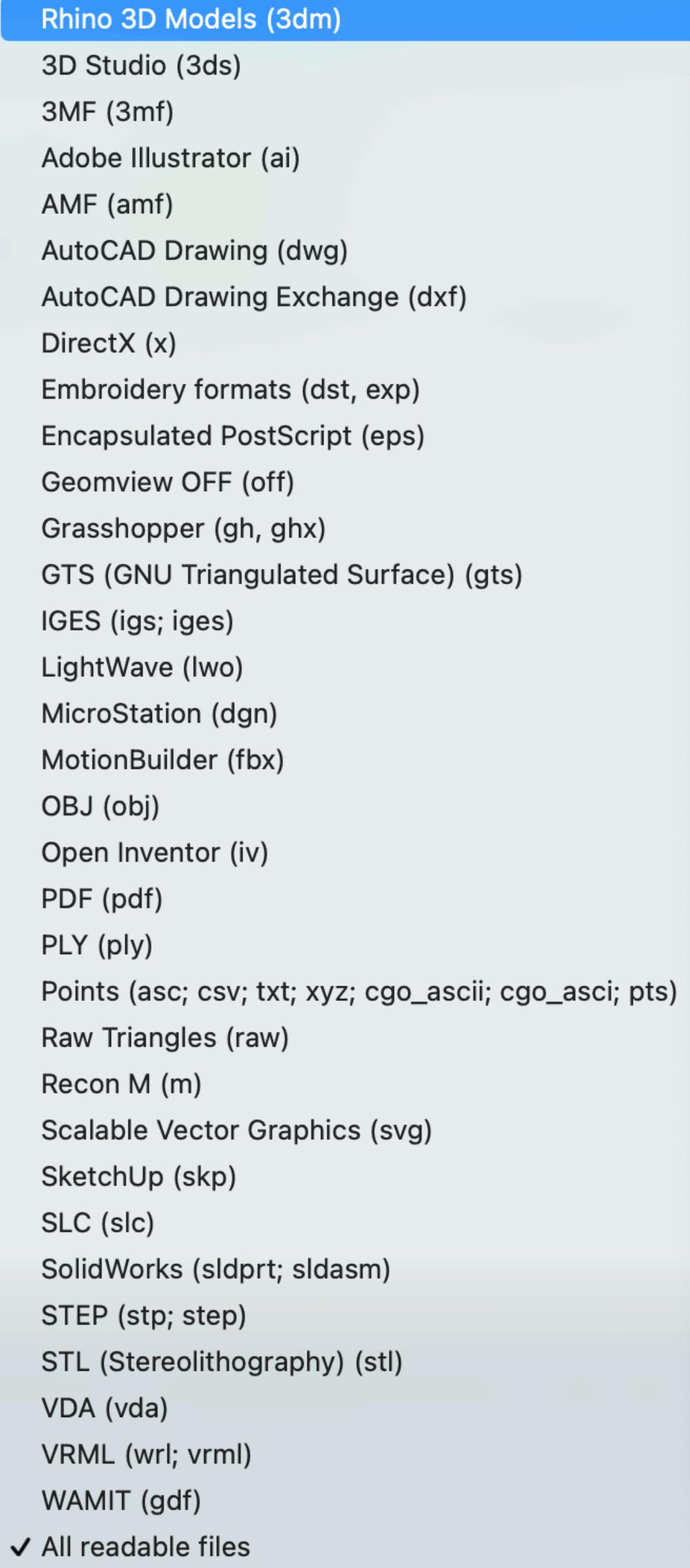
# More and more 3D (open) data from cities



Whole of the Netherlands:  
10M+ LoD1 buildings + terrain,  
roads, water, forest, etc.

- 3D IMGeo: CityGML ADE
- Government could produce the data, but very few could read it
- Were the datasets really used?

# Where is the File/Open CityGML?



Messing around with (CityGML) <https://3d.bk.tudelft.nl/svitalis/citygml/gdal/2017/07/25/me>

## Workaround for GDAL 2.2 and CityGML

While the driver works wonderfully for INSPIRE datasets (which was its original purpose), there are still some small details related more complex implementations of GML documents. CityGML is one of those schemas that really stretches some of the GML mechanisms to their limits, so we can anticipate some datasets not being so trivial to be parsed by the driver. Indeed, when I first tried to load some city models I found it didn't load all feature classes.

Thankfully, [Even Rouault](#) was there to work on some of those issues and they have already been fixed and scheduled for the next release (so expect GDAL 2.3, maybe, to work with CityGML out-of-the-box).

Meanwhile, there is a small workaround to make CityGML schema work with GDAL 2.2:

- Run `ogrinfo` once against a CityGML file (if it fails, check the solution on the fixing missing schema locations)
- Go to your home folder (e.g., `/home/your_username/` on Linux) and find the `.gdal/gmlas_xsd_cache` folder. There must be several `.xsd` files in there, including all files describing the CityGML schema.
- Open all files of CityGML (they are of `schemas.opengis.net_citygml_V.0_MODULE.xsd` format) and inside the `xs:schema` tag duplicate the first `xmns`. Then, add the prefix of this module to the second instance. So, for instance, if this is the base schema file (the core module), then the original tag should be changed from

```
<xs:schema xmlns="http://www.opengis.net/citygml/1.0" xmlns:xAL="urn:oasis:names:tc:ciiq:xsdschema:xAL:2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml" targetNamespace="http://www.opengis.net/citygml/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">
```

to

```
<xs:schema xmlns="http://www.opengis.net/citygml/1.0" xmlns:core="http://www.opengis.net/citygml/1.0" xmlns:xAL="urn:oasis:names:tc:ciiq:xsdschema:xAL:2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml" targetNamespace="http://www.opengis.net/citygml/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">
```

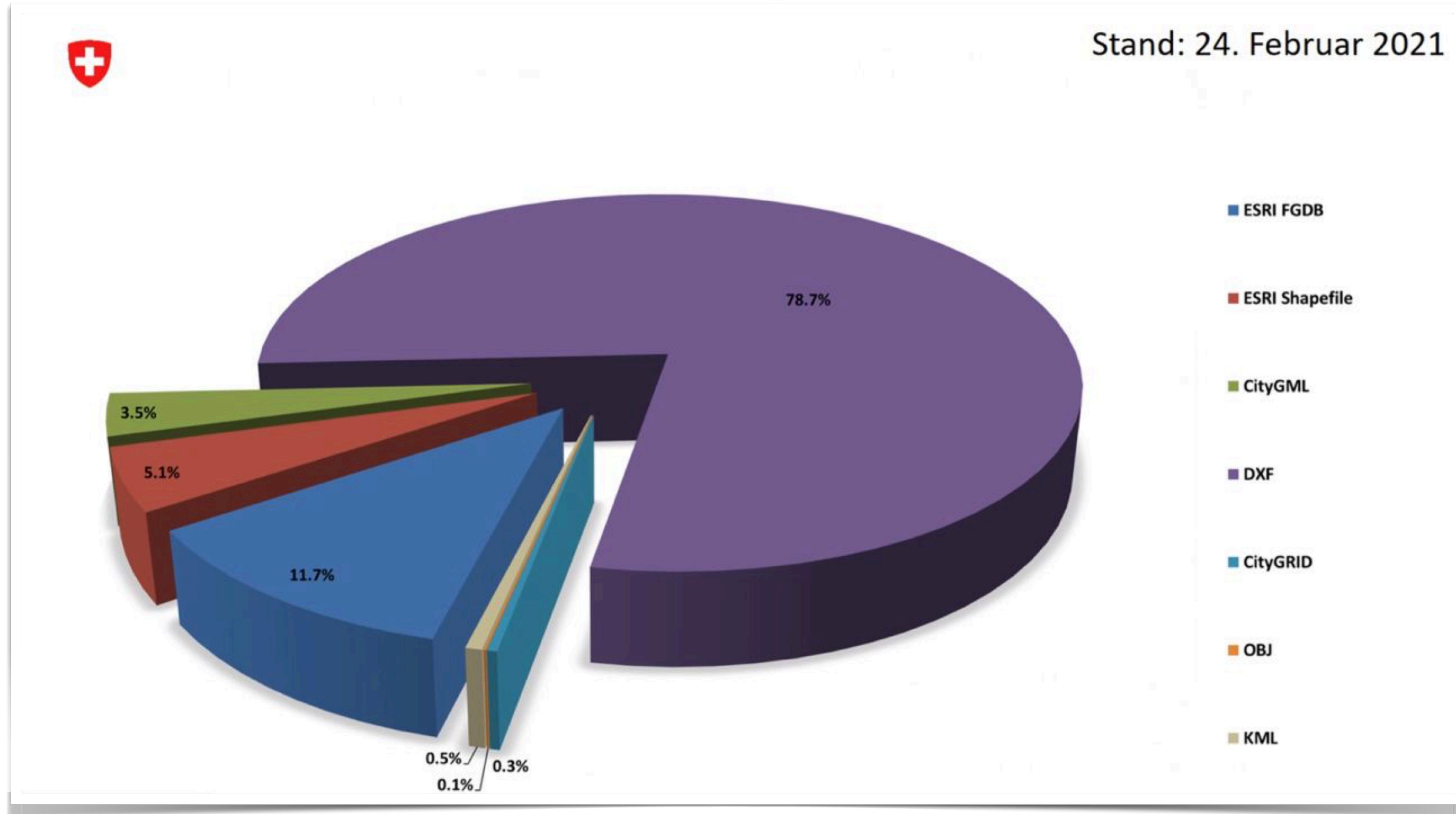
Notice, how the new `xmns:core` argument is repeating the same schema. You should do this for all modules (or at least those that your dataset uses), for instance for the building module adding the `xmns:bldg` property.

You have to do this only once and only if you are using GDAL 2.2.0 or 2.2.1. Of course, you have to repeat the same for every different CityGML version you parse in the future (again, if you are not in a newer than 2.2.1 version).

NAS - ALKIS  
Network Common Data Format

All known formats ( \*.es \*.3ds \*.ply \*.stl \*.obj \*.qobj \*.off \*.ptx \*.vmi \*.sen Script File ( \*.es )  
D-Studio File Format ( \*.3ds )  
Stanford Polygon File Format ( \*.ply )  
TL File Format ( \*.stl )  
Alias Wavefront Object ( \*.obj )  
Quad Object ( \*.qobj )  
Object File Format ( \*.off )  
TX File Format ( \*.ptx )  
CG Dump File Format ( \*.vmi )  
BX Autodesk Interchange Format ( \*.fbx )  
Reckmann File Format ( \*.bre )  
Collada File Format ( \*.dae )  
OpenCTM compressed format ( \*.ctm )  
XPE's point set (binary) ( \*.pts )  
XPE's point set (ascii) ( \*.apts )  
YZ Point Cloud (with or without normal) ( \*.xyz )  
Protein Data Bank ( \*.pdb )  
RI (photogrammetric reconstructions) ( \*.tri )  
SC (ascii triplets of points) ( \*.asc )  
XT (Generic ASCII point list) ( \*.txt )  
3D File Format - XML encoding ( \*.x3d )  
3D File Format - VRML encoding ( \*.x3dv )  
WRL 2.0 File Format ( \*.wrl )

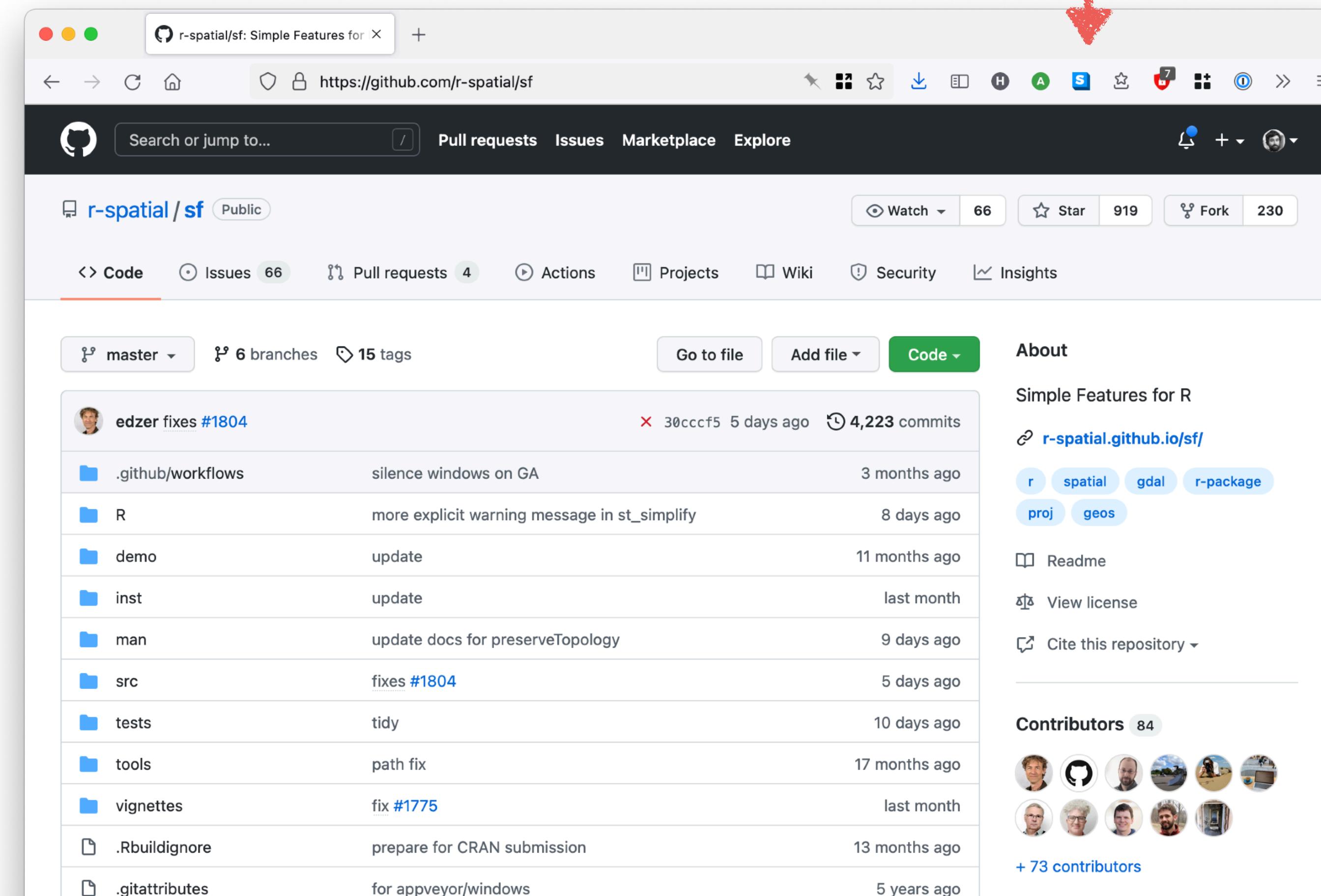
# swisstopo downloads of their 3D model



-Raphaël Bovier, from his presentation at the *3D city modelling Workshop*, Nordic Cooperation in the fields of Spatial Data and Land Administration (2021-09-29)

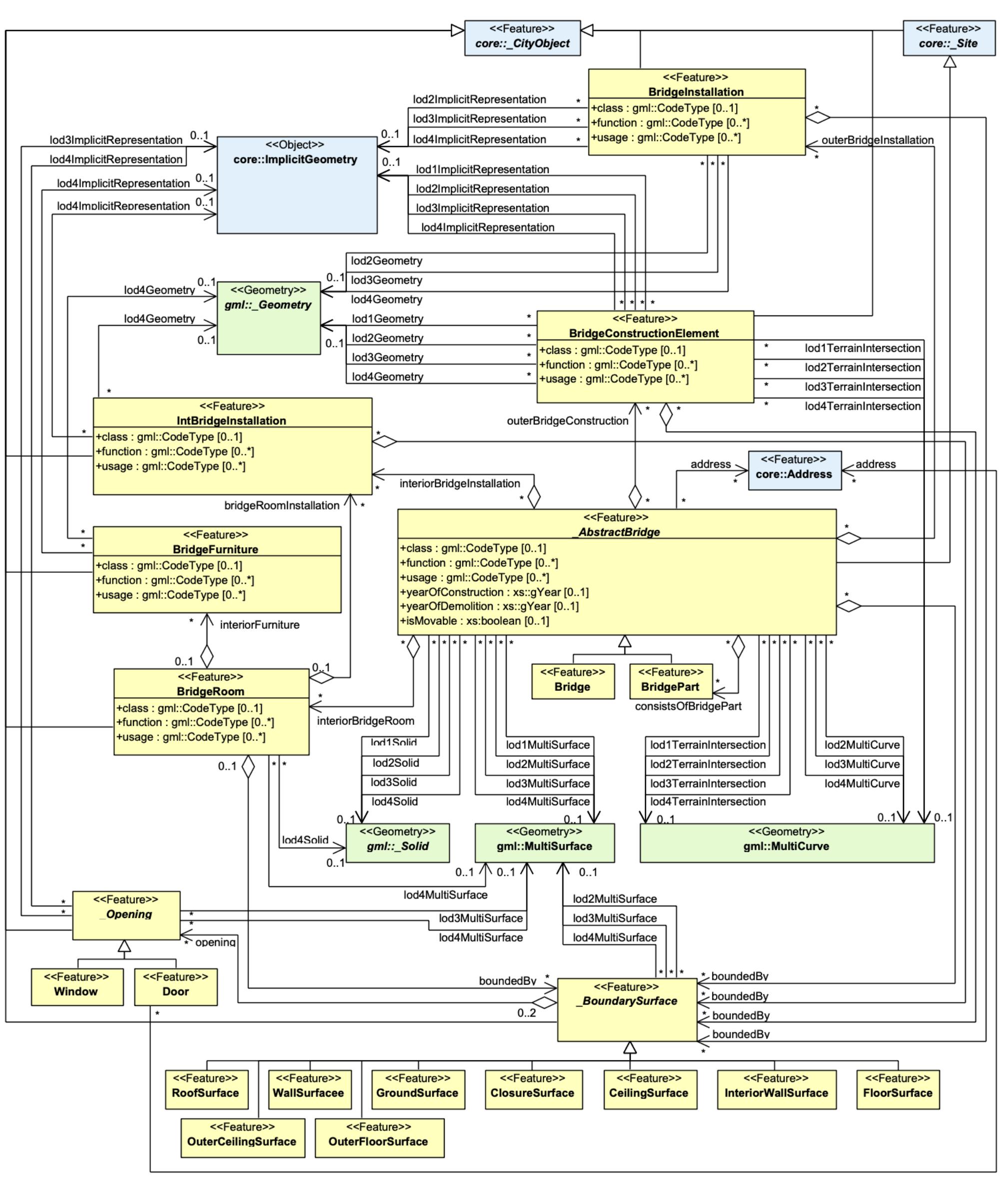
# Scientists in other communities are not really involved

- CFD/simulation community == OBJ/STL
- Computer Graphics == OBJ/STL
- Other communities have scientists involved in the ecosystem
- Why so few here?



**Why are the datasets  
seldom used?**

# UML shouldn't be the main goal



- Many papers have as result UML
- Most OGC standards focus on the data model
- “Let’s model the full reality” → increases complexity
- We stay at the theoretical level
- Users are not central

# Standards don't ensure usefulness (quite the contrary one might say...)

A screenshot of a Twitter profile for the account "Shapefile (@shapefile)". The profile picture is a white circle containing a globe icon with a green "SHP" label on top. The bio reads: "#FMEUC Lifetime Achievement Award Winner, @GeoHipster Interviewee, the tardigrade of formats, leading #TeamShapefile. I'll be here forever. #shapefile parody". A red arrow points from the text "I'll be here forever. #shapefile" to the bio. The profile has 615 Following and 3,800 Followers.

Shapefile  
7,869 Tweets

**Country outlines from shapefile**

**Shapefile**  
@shapefile

#FMEUC Lifetime Achievement Award Winner, @GeoHipster Interviewee, the tardigrade of formats, leading #TeamShapefile. I'll be here forever. #shapefile parody

All over your file system esri.com/library/whitep... Joined June 2016

615 Following 3,800 Followers

# Quiz: you need to store 2D polygons, which one you choose?

## GML 3.2.2 specs

(427 pages)

OGC 07-036r1

**NOTE** The definition of this group may be used as a pattern in the definition of geometric primitives instead of using this group definition directly. The main change will typically be a change in the multiplicity of the referenced group. A LineString, for example, requires at least two positions.

Also, to support deprecated elements, i.e. `gml:coordinates` (superceded by `gml:posList`) and `gml:pointRep` (superceded by `gml:pointProperty`), the current encodings of point arrays in GML, e.g. in curve segments, uses this group as a pattern and adds the deprecated elements.

### 10.1.4.5 VectorType, Vector

```
<complexType name="VectorType">
  <simpleContent>
    <restriction base="gml:DirectPositionType"/>
  </simpleContent>
</complexType>
```

```
<element name="vector" type="gml:VectorType" />
```

`gml:vector` implements ISO/TS 19103 Vector (see D.2.3.2 and ISO/TS 19103:2005, 6.5.2.6).

For some applications the components of the position may be adjusted to yield a unit vector.

**NOTE** This definition allows `VectorType` to be used elsewhere when appropriate — e.g. for `offsetVector` in `grids.xsd`, and `vector` to be used directly when appropriate — e.g. in `DirectionVector` in `direction.xsd`.

### 10.1.4.6 EnvelopeType, Envelope

```
<complexType name="EnvelopeType">
  <choice>
    <sequence>
      <element name="lowerCorner" type="gml:DirectPositionType"/>
      <element name="upperCorner" type="gml:DirectPositionType"/>
    </sequence>
    <element ref="gml:pos" minOccurs="2" maxOccurs="2"/>
    <element ref="gml:coordinates"/>
  </choice>
  <attributeGroup ref="gml:SRSReferenceGroup"/>
</complexType>
```

```
<element name="Envelope" type="gml:EnvelopeType" substitutionGroup="gml:AbstractObject"/>
```

`gml:Envelope` implements ISO 19107 GM\_Envelope (see D.2.3.4 and ISO 19107:2003, 6.4.3).

Envelope defines an extent using a pair of positions defining opposite corners in arbitrary dimensions. The first direct position is the "lower corner" (a coordinate position consisting of all the minimal ordinates for each dimension for all points within the envelope), the second one the "upper corner" (a coordinate position consisting of all the maximal ordinates for each dimension for all points within the envelope).

*The use of the properties "coordinates" and "pos" in Envelope has been deprecated. The explicitly named properties "lowerCorner" and "upperCorner" shall be used instead.*

## GeoJSON specs

(28 pages )

https://datatracker.ietf.org/

For example, a line extending from 45 degrees N, 170 degrees E across the antimeridian to 45 degrees N, 170 degrees W should be cut in two and represented as a MultiLineString.

```
{
  "type": "MultiLineString",
  "coordinates": [
    [
      [170.0, 45.0], [180.0, 45.0]
    ], [
      [-180.0, 45.0], [-170.0, 45.0]
    ]
  ]
}
```

Butler, et al.

Standards Track

[Page 10]

RFC 7946

GeoJSON

August 2016

A rectangle extending from 40 degrees N, 170 degrees E across the antimeridian to 50 degrees N, 170 degrees W should be cut in two and represented as a MultiPolygon.

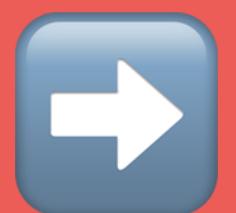
```
{
  "type": "MultiPolygon",
  "coordinates": [
    [
      [
        [180.0, 40.0], [180.0, 50.0], [170.0, 50.0],
        [170.0, 40.0], [180.0, 40.0]
      ]
    ],
    ...
  ]
},
```

# ADEs: where are the schemas?

**Table 1** List of identified ADEs

	ADE <sup>a</sup>	Purpose	Website <sup>d</sup>	XSD <sup>b</sup>	UML	Origin <sup>c</sup>	Further reading
1	Energy ADE	Application	URL-1	•	•	Europe	[14–17]
2	Energy Efficiency ADE	Application		•	•	Italy	[20]
3	Energy Efficiency ADE (ii)	Application			•	Spain	[21–23, 42]
4	Noise ADE	Application	URL-2	•	•	Germany	[1, 24]
5	Extended Noise ADE						
6	Road Traffic Noise ADE						
7	Robotics ADE						
8	UtilityNetworkADE						
9	CAFM ADE						
10	Immovable Property Taxation ADE						
11	Cadastre ADE						
12	CityGML-LADM ADE						
13	Cultural Heritage ADE						
14	Cultural Heritage ADE (ii)						
15	Cultural Heritage ADE (iii)						
16	Heritage house ADE						
17	Intervention ADE						
18	BCH Management ADE						
19	Indoor N&P ADE						
20	Indoor ADE						
21	i-SCOPE						
22	HydroADE						
23	AR ADE						
24	Collada FX ADE						
25	ENC ADE						
26	Air Quality ADE						
27	IMGeo ADE						
28	CityGML-TRKBIS						
29	INSPIRE ADE						
30	ACRoofADE						
31	CityGML iTINs ADE						
32	Vegetation Objects ADE						
33	Dynamizers						
34	Dynamic ADE						
35	Geodata Join ADE	Generic			Germany	[72]	
36	Topo ADE	Generic			China	[73]	
37	Transport ADE	Generic			Netherlands	[74]	
38	Traffic Sign ADE	Generic			Spain	[75]	
39	3D-GEM	Generic		•	Netherlands	[76]	
40	New LOD ADE	Generic		•	Netherlands	[77]	
41	Semantic City Model	Generic		•	China	[79]	
42	GeoBIM	Generic	URL-9	•	Netherlands	[80]	
43	PANTURA ADE	Generic			Netherlands	[81]	
44	3D Metadata ADE	Generic	URL-10	•	•	Netherlands	
			Total:	18	30		

Files with ADEs are way more difficult  
to read than “vanilla” ones



custom code

Biljecki F, Kumar K, and Nagel C (2018). *CityGML Application Domain Extension (ADE): overview of developments*. Open Geospatial Data, Software and Standards (2018) 3:13

# Little focus on *usable* datasets

```
410405      ~gml:doubleAttribute name="lod2TerrainIntersection" ~gml:value>0.075</gml:value>
410406      ~gml:doubleAttribute name="lod2TerrainHeight" ~gml:value>0.075</gml:value>
410407      ~gml:doubleAttribute name="lod2TerrainWeight" ~gml:value>0.075</gml:value>
410408
410409
410410      <app:appearance> ←
410411          <app:Appearance>
410412              <app:surfaceDataMember>
410413                  <app:X3DMaterial>
410414                      <app:ambientIntensity>0.2</app:ambientIntensity>
410415                      <app:diffuseColor>0.922 0.922 0.922</app:diffuseColor>
410416                      <app:target>#GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_1</app:target>
410417                      <app:target>#GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_2</app:target>
410418                      <app:target>#GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_3</app:target>
410419                      <app:target>#GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_4</app:target>
410420                      <app:target>#GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_5</app:target>
410421                  </app:X3DMaterial>
410422          </app:surfaceDataMember>
410423          <app:surfaceDataMember>
410424              <app:X3DMaterial>
410425                  <app:ambientIntensity>0.2</app:ambientIntensity>
410426                  <app:diffuseColor>0.502 0 0</app:diffuseColor>
410427                  <app:target>#GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_6</app:target>
410428              </app:X3DMaterial>
```

```
410409      ~gml:multicurve>
410410      </bldg:lod2TerrainIntersection>
410411      <bldg:boundedBy>
410412          <bldg:WallSurface>
410413              <bldg:lod2MultiSurface> ←
410414                  <gml:MultiSurface>
410415                      <gml:surfaceMember>
410416                          <gml:Polygon gml:id="GUID_6E02C743-872C-4B43-A45E-46D1BDCB57A0_3_1">
410417                              <gml:exterior>
410418                                  <gml:LinearRing>
410419                                      <gml:pos>78853.849 457917.572 9.873</gml:pos>
410420                                      <gml:pos>78856.397 457913.899 9.873</gml:pos>
410421                                      <gml:pos>78856.397 457913.899 9.873</gml:pos>
```

- Assumed encoding (GML) can be automatically derived from schemas
- Datasets are often hierarchical and have annoying quirks

# “GML madness”: how many ways to store a simple square?

The screenshot shows a blog post titled "Geo tips & tricks" from "erouault.blogspot.com". The post is dated "dimanche 6 avril 2014" and is titled "GML madness". It discusses the various ways to encode a simple square polygon in GML. A diagram of a unit square is shown on a coordinate system with axes labeled 0 and 1. The post also includes a sidebar with member profiles and a blog archive.

**Membres**

**Volgers (39) Volgende**

**Archives du blog**

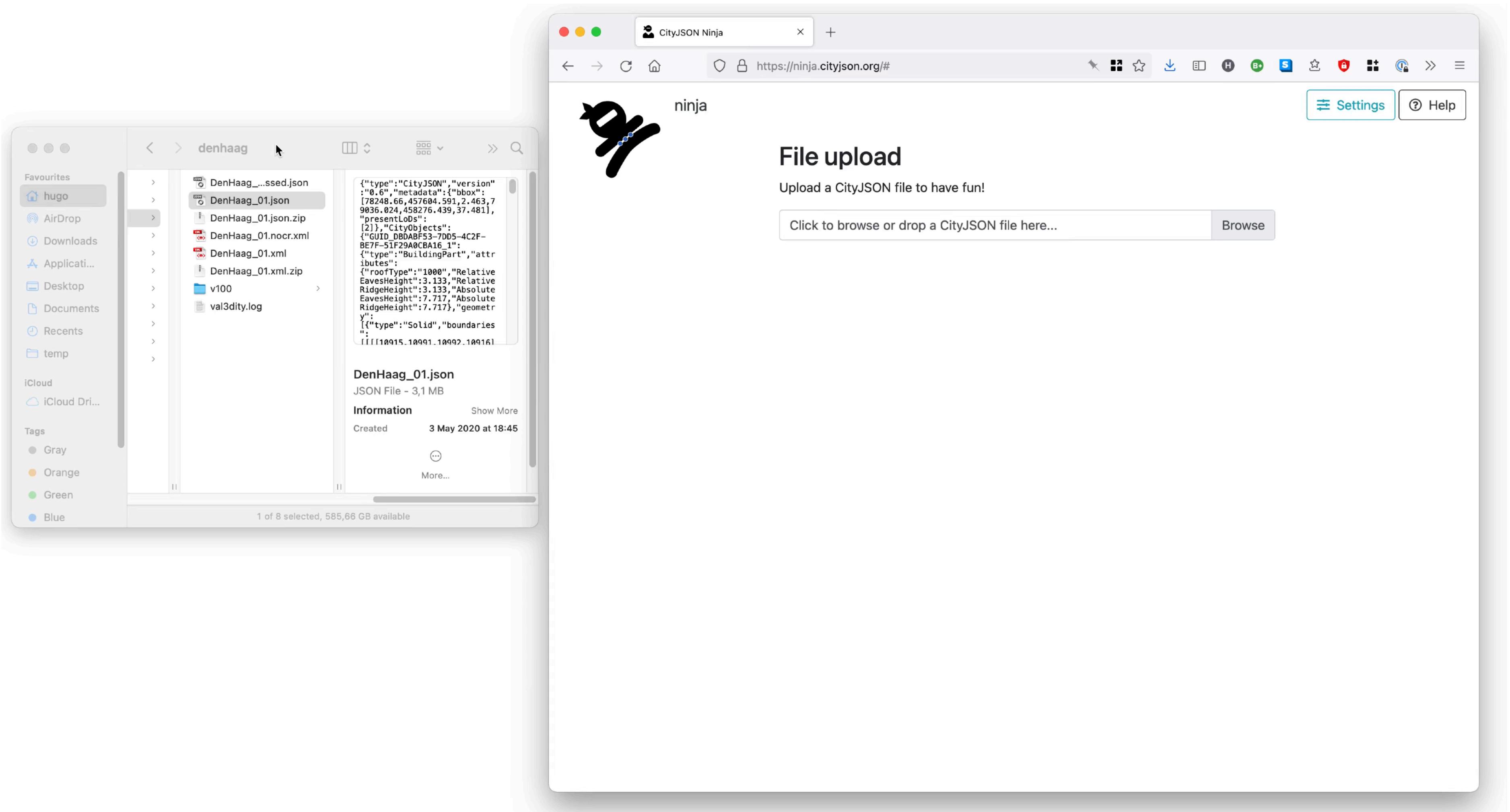
- 2021 (3)
- 2019 (2)
- 2018 (8)
- 2017 (3)
- 2016 (5)
- 2015 (7)
- ▼ 2014 (12)
  - décembre (2)
  - novembre (1)
  - octobre (3)
  - septembre (1)
  - ▼ avril (3)

GDAL/OGR 1.11.0 released



# Complex formats usually have fewer software support

No known JavaScript parser for CityGML files!?



# Real-life reactions of students in my course

When learning about semantic  
3D city models



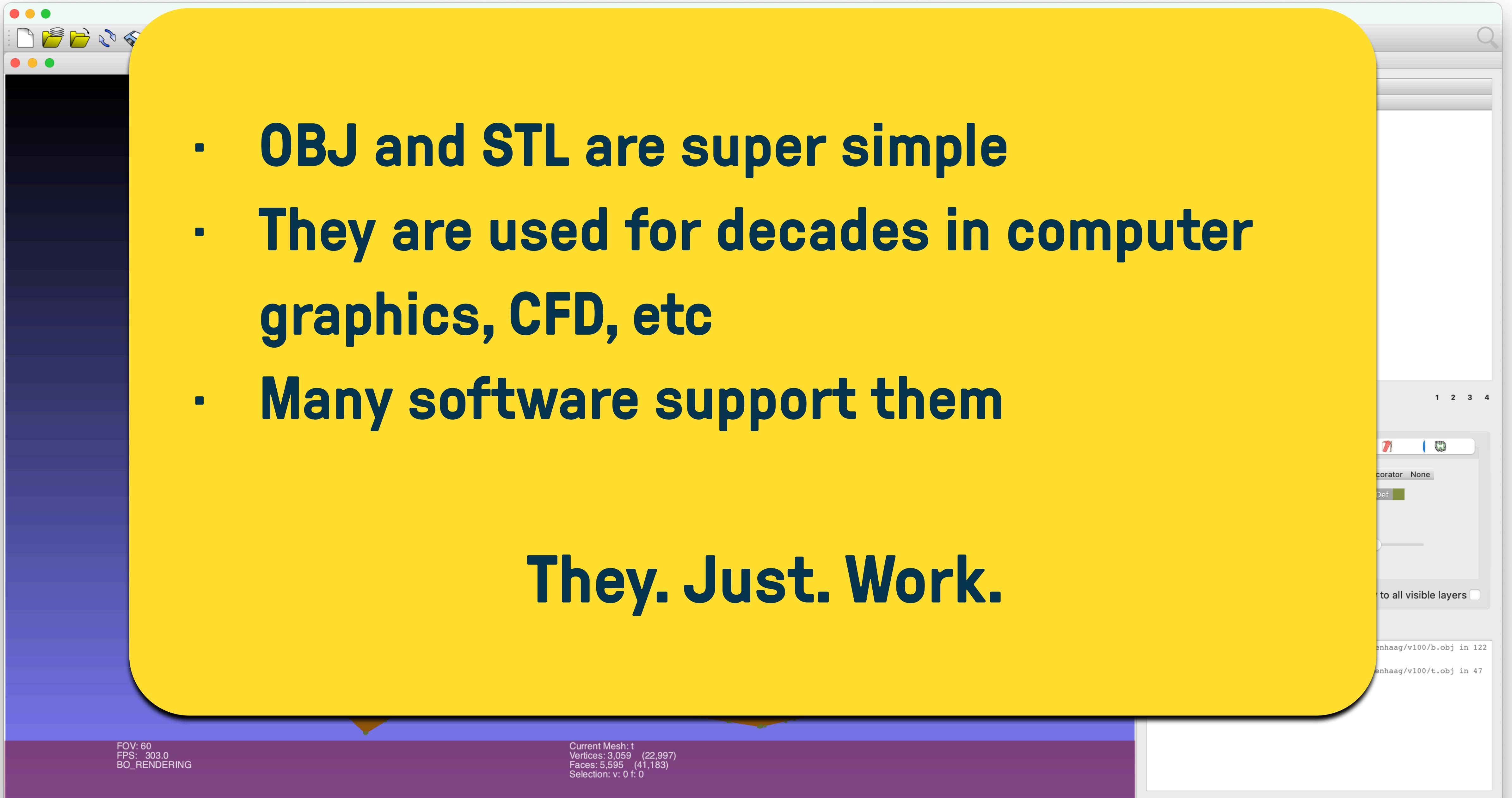
Trying to read a CityGML file  
with Python



# Many of us promote semantic formats but use... OBJ/STL

- **OBJ and STL are super simple**
- **They are used for decades in computer graphics, CFD, etc**
- **Many software support them**

**They. Just. Work.**



FOV: 60  
FPS: 303.0  
BO\_RENDERING

Current Mesh: t  
Vertices: 3,059 (22,997)  
Faces: 5,595 (41,183)  
Selection: v: 0 f: 0

**What can we do  
about it?**

# Shouldn't we apply the Rule of Least Power?

W3 The Rule of Least Power

←

**Principle**

*Principle: Powerful languages inhibit information reuse.*

**Good Practice**

*Good Practice: Use the least powerful language suitable for expressing information, constraints or programs on the World Wide Web.*

## Editors:

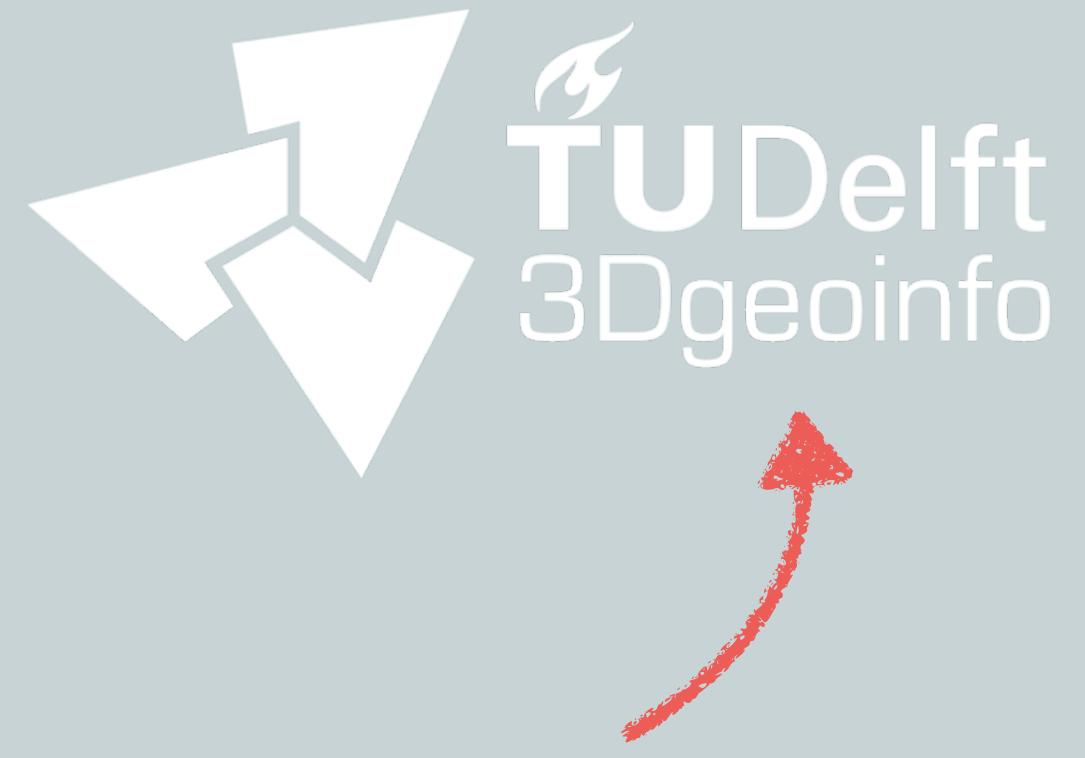
Tim Berners-Lee, World Wide Web Consortium [timbl@w3.org](mailto:timbl@w3.org)

Noah Mendelsohn, IBM Corporation [noah\\_mendelsohn@us.ibm.com](mailto:noah_mendelsohn@us.ibm.com)

This document is also available in these non-normative formats: [XML](#).

# OK, but concretely here?

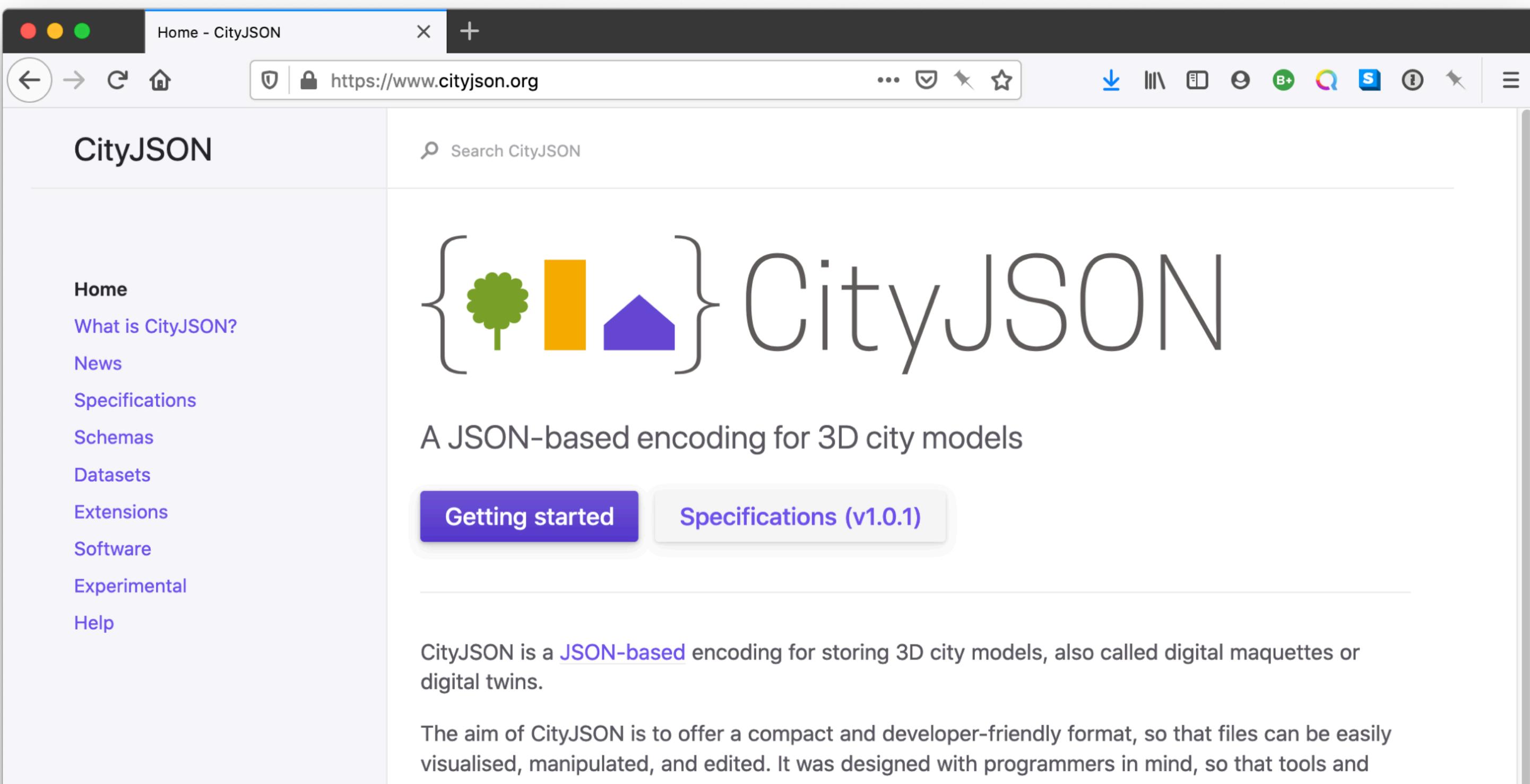
1. Make *one* usable **profile** of CityGML that supports most important parts (similar to *Simple Features*)
2. Please **no GML** for the encoding, no developers want this
3. Write **parsers** for the format (Python, JS, C++) so that others can just reuse/copy. Maybe an API to manipulate the data?
4. Think “like a designer” (how will the product be used?)
5. **“Eat your own dog food”**



**Two projects where we  
applied those principles**

# CityJSON v1.0

-  community standard
- all modules of CityGML v2.0 are mapped 💪
- Subset of CityGML (~97% of features)
- software for full conversion CityGML <-> CityJSON
- several software support it



The screenshot shows the homepage of the CityJSON website at <https://www.cityjson.org>. The page has a dark header with the OGC logo and a navigation bar with links like Home, What is CityJSON?, News, Specifications, Schemas, Datasets, Extensions, Software, Experimental, and Help. The main content area features a large "CityJSON" logo with icons of a tree, a bar chart, and a house. Below the logo is the text "A JSON-based encoding for 3D city models". There are two buttons at the bottom: "Getting started" (in purple) and "Specifications (v1.0.1)". A footer note states: "CityJSON is a [JSON-based](#) encoding for storing 3D city models, also called digital maquettes or digital twins. The aim of CityJSON is to offer a compact and developer-friendly format, so that files can be easily visualised, manipulated, and edited. It was designed with programmers in mind, so that tools and".

1. "Building"
2. "BuildingPart"
3. "BuildingInstallation"
4. "Road"
5. "Railway"
6. "TransportSquare"
7. "TINRelief"
8. "WaterBody"
9. "PlantCover"
10. "SolitaryVegetationObject"
11. "LandUse"
12. "CityFurniture"
13. "GenericCityObject"
14. "Bridge"
15. "BridgePart"
16. "BridgeInstallation"
17. "BridgeConstructionElement"
18. "Tunnel"
19. "TunnelPart"
20. "TunnelInstallation"
21. "CityObjectGroup"

# Same information as CityGML, but in JSON format

```
{  
  "type": "CityJSON",  
  "version": "1.0",  
  "metadata": {  
    "referenceSystem": "urn:ogc:def:crs:EPSG::7415",  
  },  
  "CityObjects": {  
    "id-1": {  
      "type": "Building",  
      "attributes": {  
        "measuredHeight": 22.3,  
        "roofType": "gable",  
        "owner": "Elvis Presley"  
      },  
      "geometry": [  
        {  
          "type": "MultiSurface",  
          "boundaries": [  
            [[0, 3, 2, 1]], [[4, 5, 6, 7]], [[0, 1, 5, 4]]  
          ]  
        }  
      ]  
    },  
    "vertices": [  
      [23.1, 2321.2, 11.0],  
      [111.1, 321.1, 12.0],  
      ...  
    ],  
    "appearance": {  
      "materials": [],  
      "textures": [],  
      "vertices-texture": []  
    }  
  }  
}
```

human-readable file

computers prefer this over XML

~6X compacter than CityGML

# Python parsing is very easy



```
import json

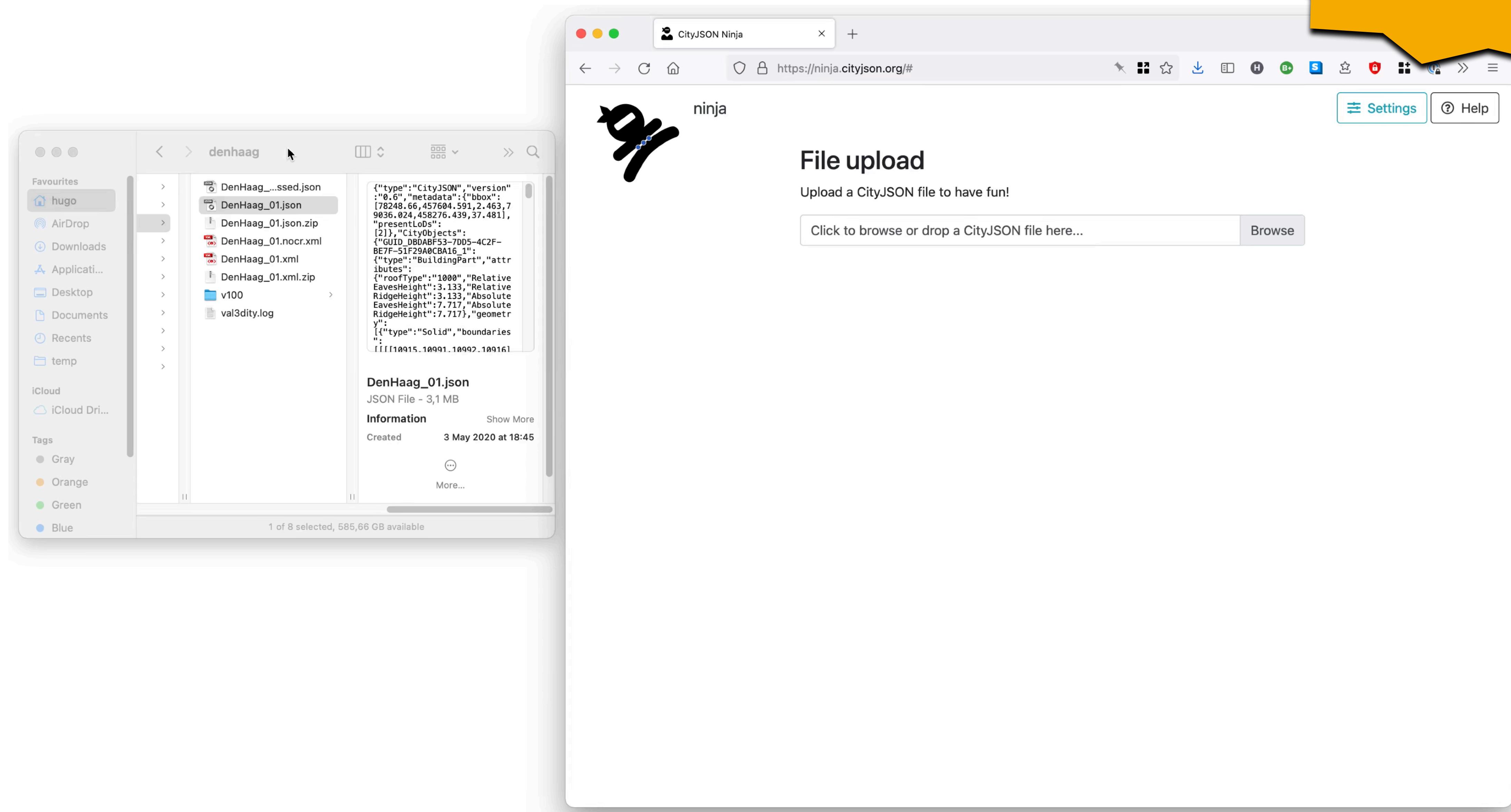
fin = open('mycity.json')
cm = json.loads(fin.read())

print "There are", len(cm['CityObjects']), "CityObjects"

# list all ids
for id in cm['CityObjects']:
    print "\t", id
```

# CityJSON can be easily parsed with JavaScript

Developed by MSc  
students



# CityJSON is “ready” for machine learning

Analysing CityModels with pandas X +

cgio, or CityJSON.io Tutorials Reference

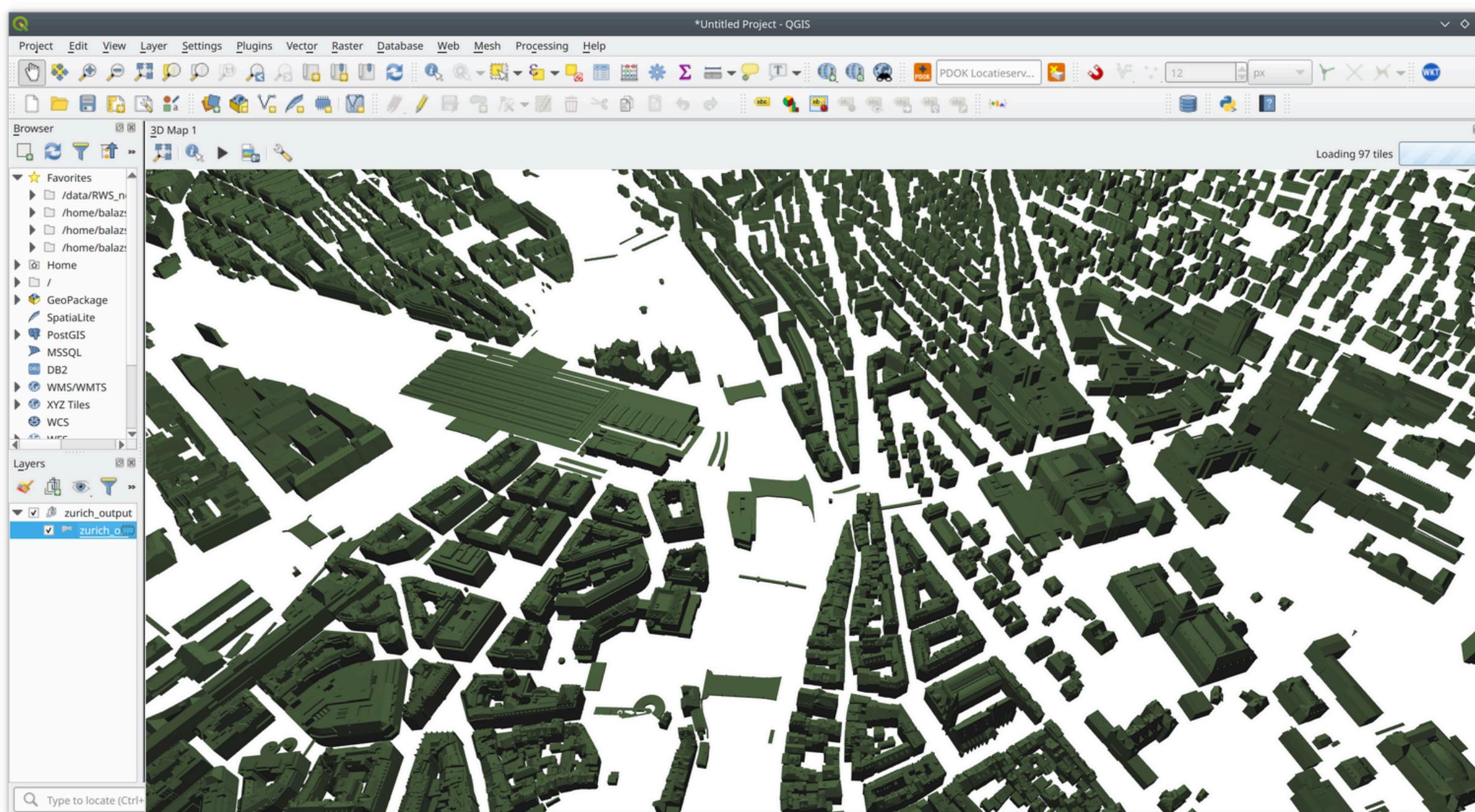
Search the docs ...

cgio API tutorial

**Analysing CityModels with pandas and scikit-learn**

Creating city models and objects

## Analysing CityModels with pandas and scikit-learn



In the following I show how to compute some attributes from CityObject geometry and use these attributes as input for machine learning. For this we use the LoD2 model of Zürich.

**new!**

# 3D BAG: all 10M+ buildings in the Netherlands

<https://3dbag.nl>

3D BAG Viewer

3D BAG by tudelft3d v21.03.1 beta

Baselayer LoD Search for a place

Attribute Value

Tile number	5910
identificatie	NL. IMBAG. Pand. 050310000032914
h_maaiveld	0.002
h_dak_70p	21.499681
dak_type	slanted
pw_bron	ahn3
pw_datum	2013-12-01
val3dity_codes	[ 302 ]

Attribute descriptions you can find in the [documentation](#).

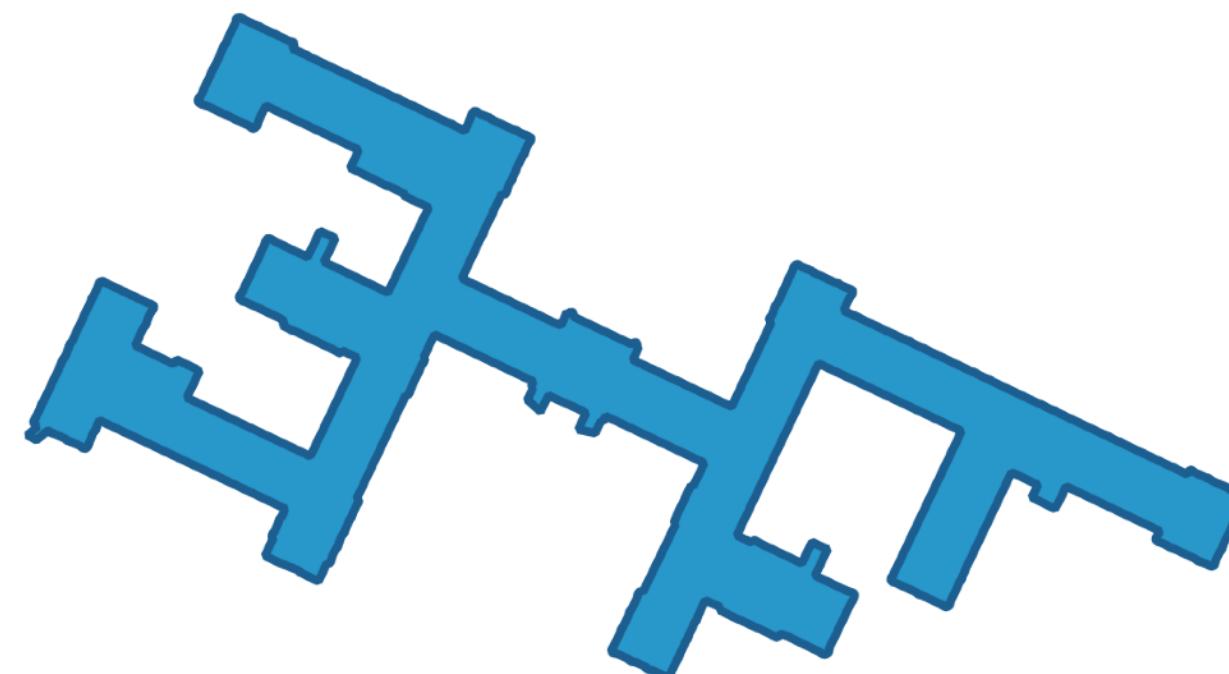
Report a problem with this building

Attributes 11.1 m 90.0 °

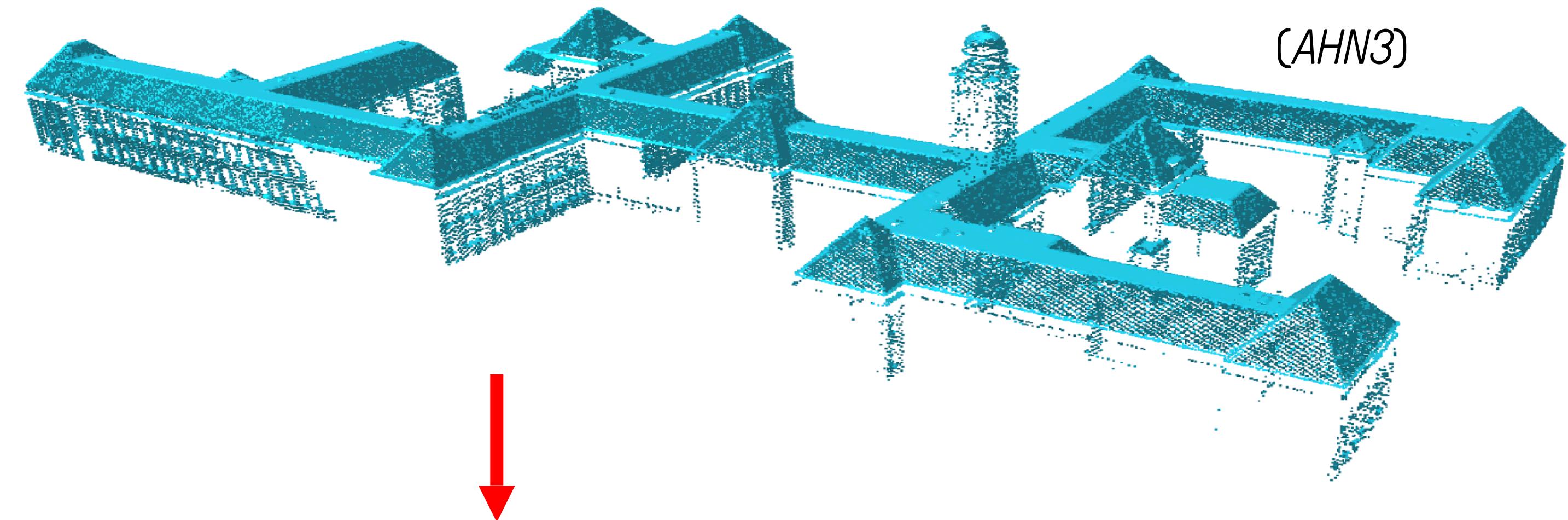
Baselayer from PDOK © 3D BAG by tudelft3d

# 3D BAG is based on open datasets

Footprint (*BAG*)



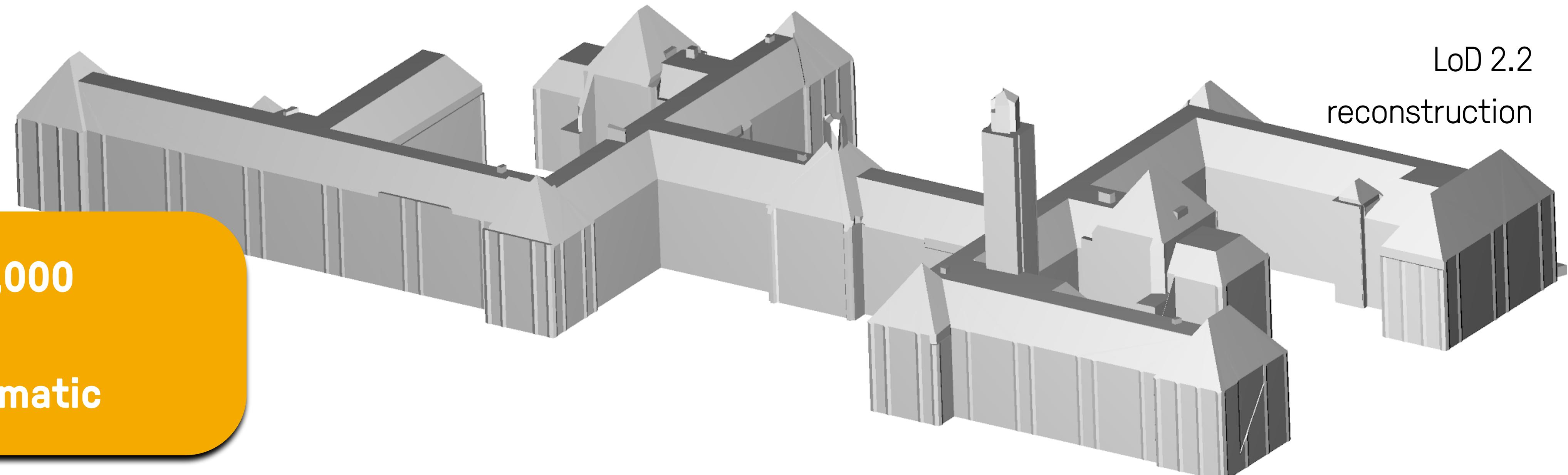
+



Point cloud  
(*AHN3*)



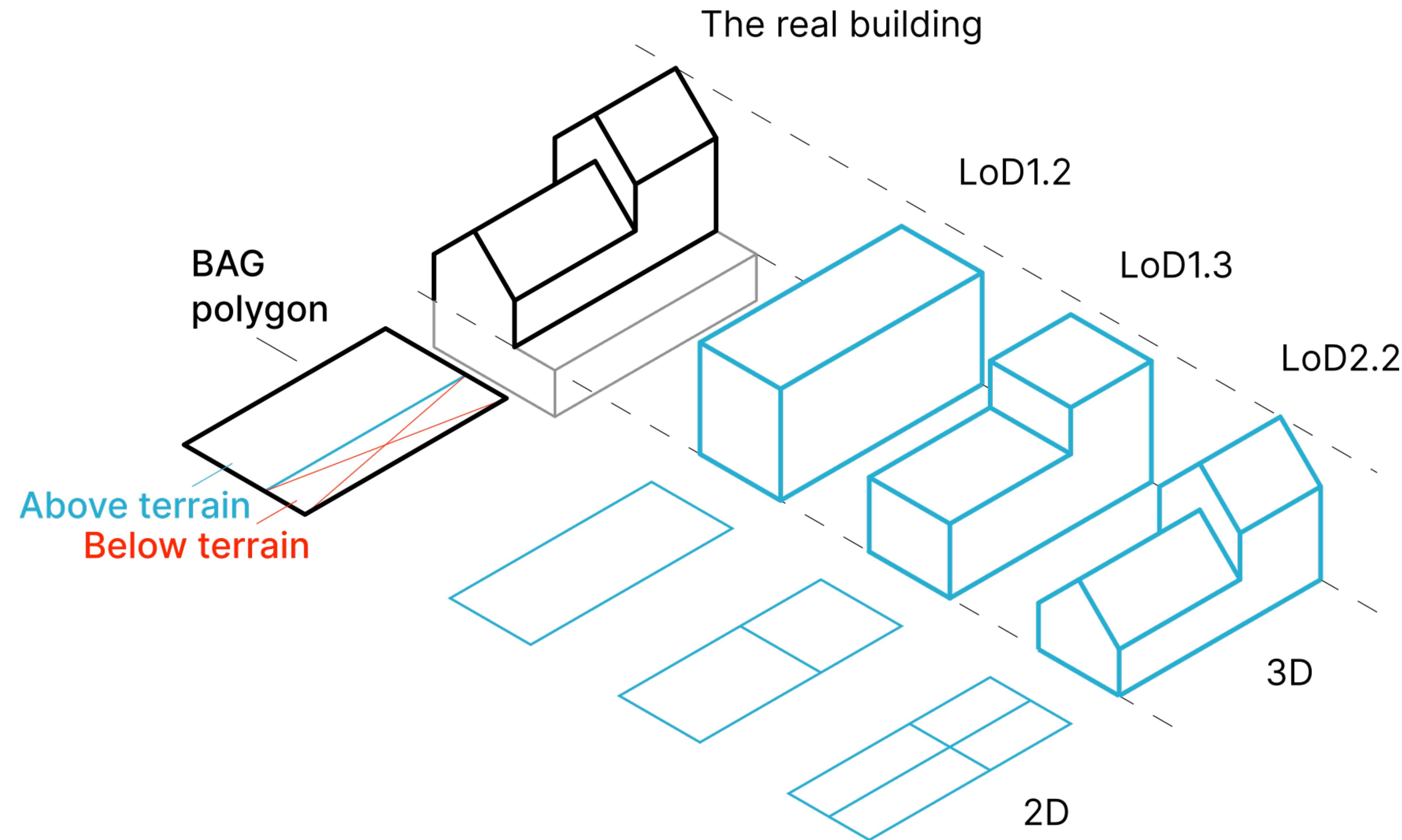
LoD 2.2  
reconstruction



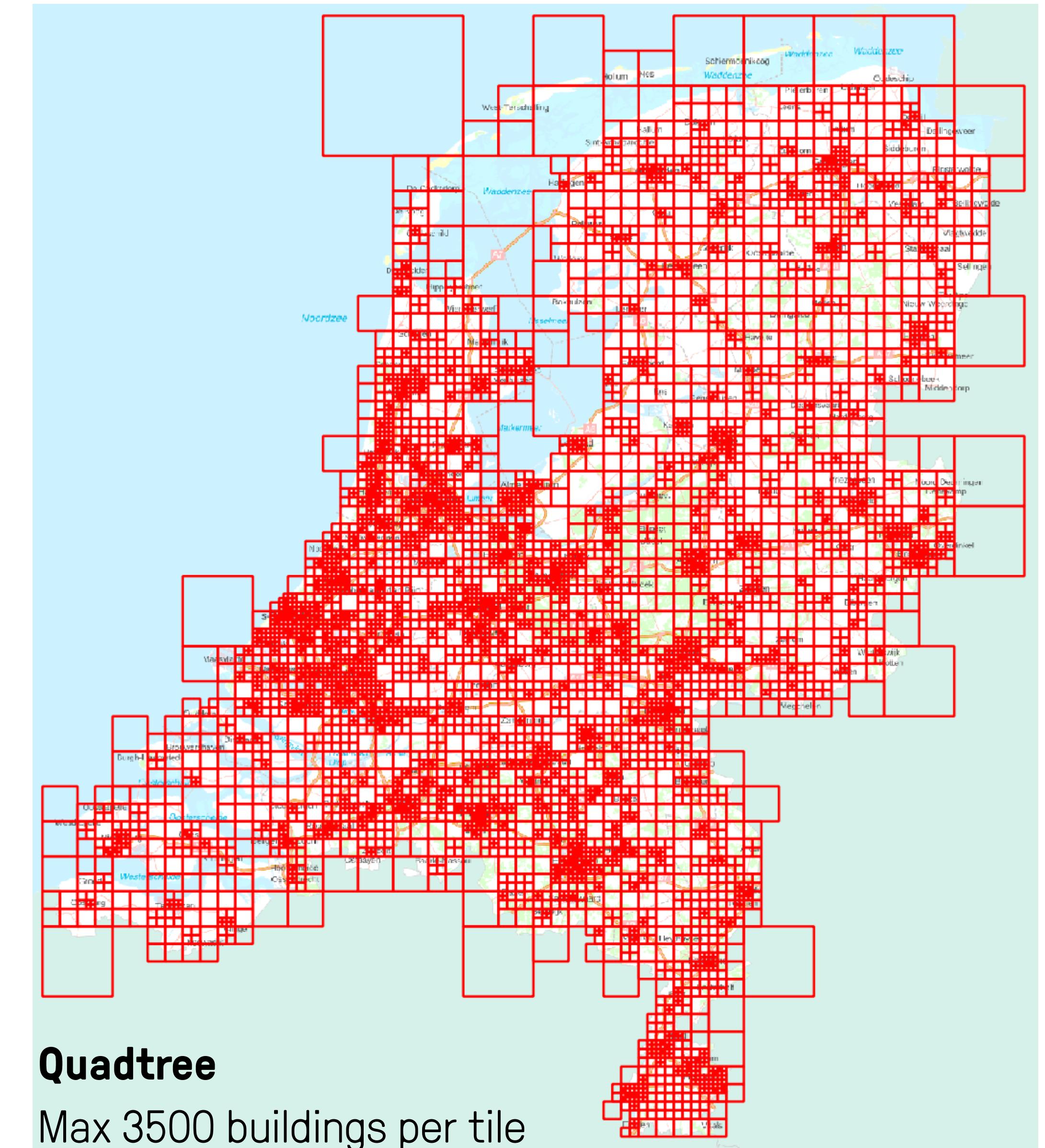
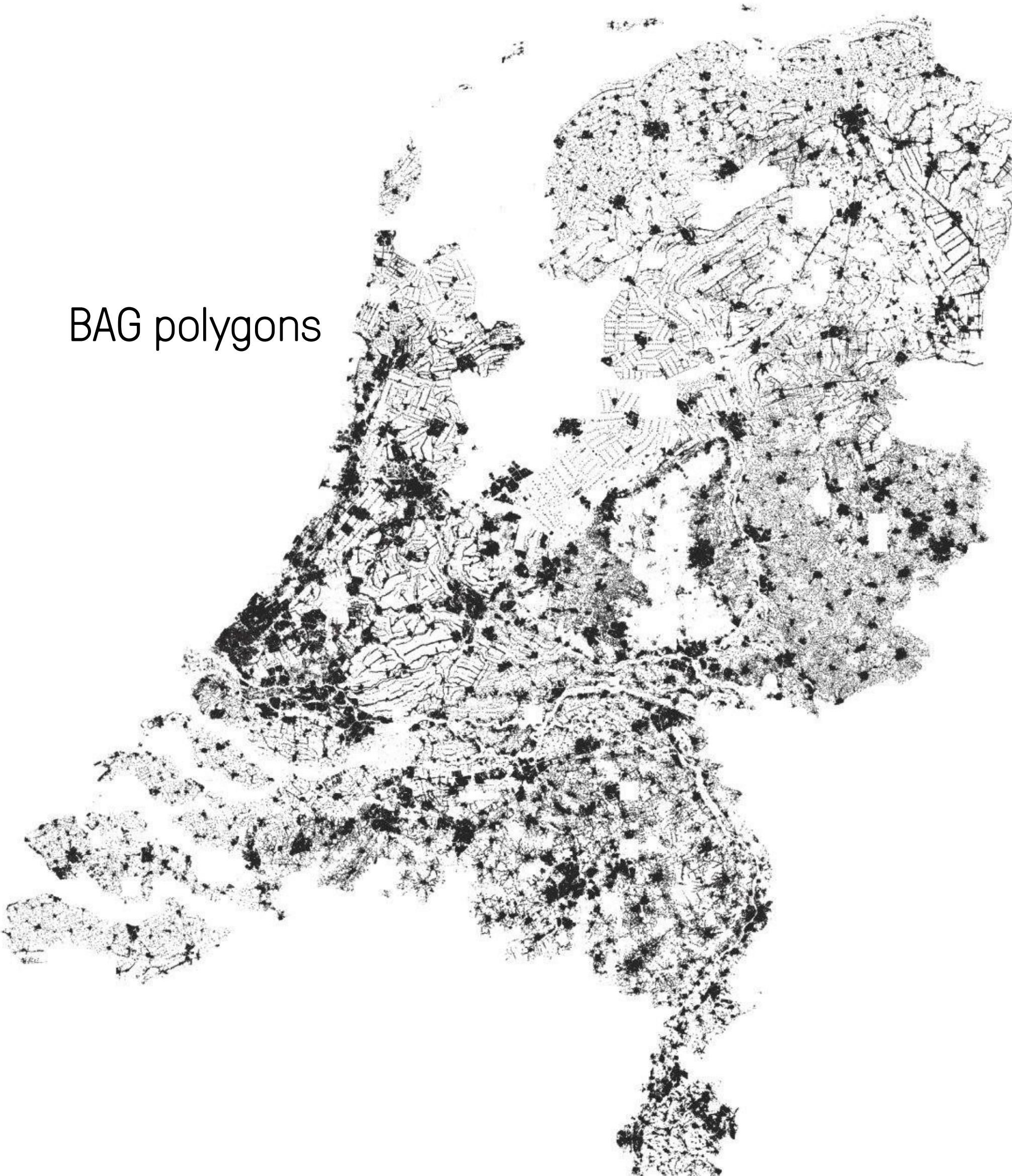
X 10,000,000

100% automatic

# 3 different LoDs



# 3D BAG is tiled into \*small\* and \*manageable\* datasets



# Download is simple and fast

The screenshot shows a web browser window for '3D BAG Viewer' at <https://3dbag.nl/en/download>. A red callout box highlights the download statistics: 'Stats last 2 weeks (~32000 downloads)'.

**Downloads per tile**  
To keep filesizes manageable the 3D BAG data is split into tiles. Click below to select the tile of interest to see its download stats.  
[Pick a tile](#)

**Webservices**  
These allow you to explore the entire dataset without having to download it.

Type	URL
WMS ?	<a href="https://data.3dbag.nl/wms">https://data.3dbag.nl/wms</a>
WFS ?	<a href="https://data.3dbag.nl/wfs">https://data.3dbag.nl/wfs</a>

**PostgreSQL data dump**  
The PostgreSQL file below contain the raw 3D BAG data for the whole of the Netherlands, including geometry and attributes. Beware, this is a very large file to download.

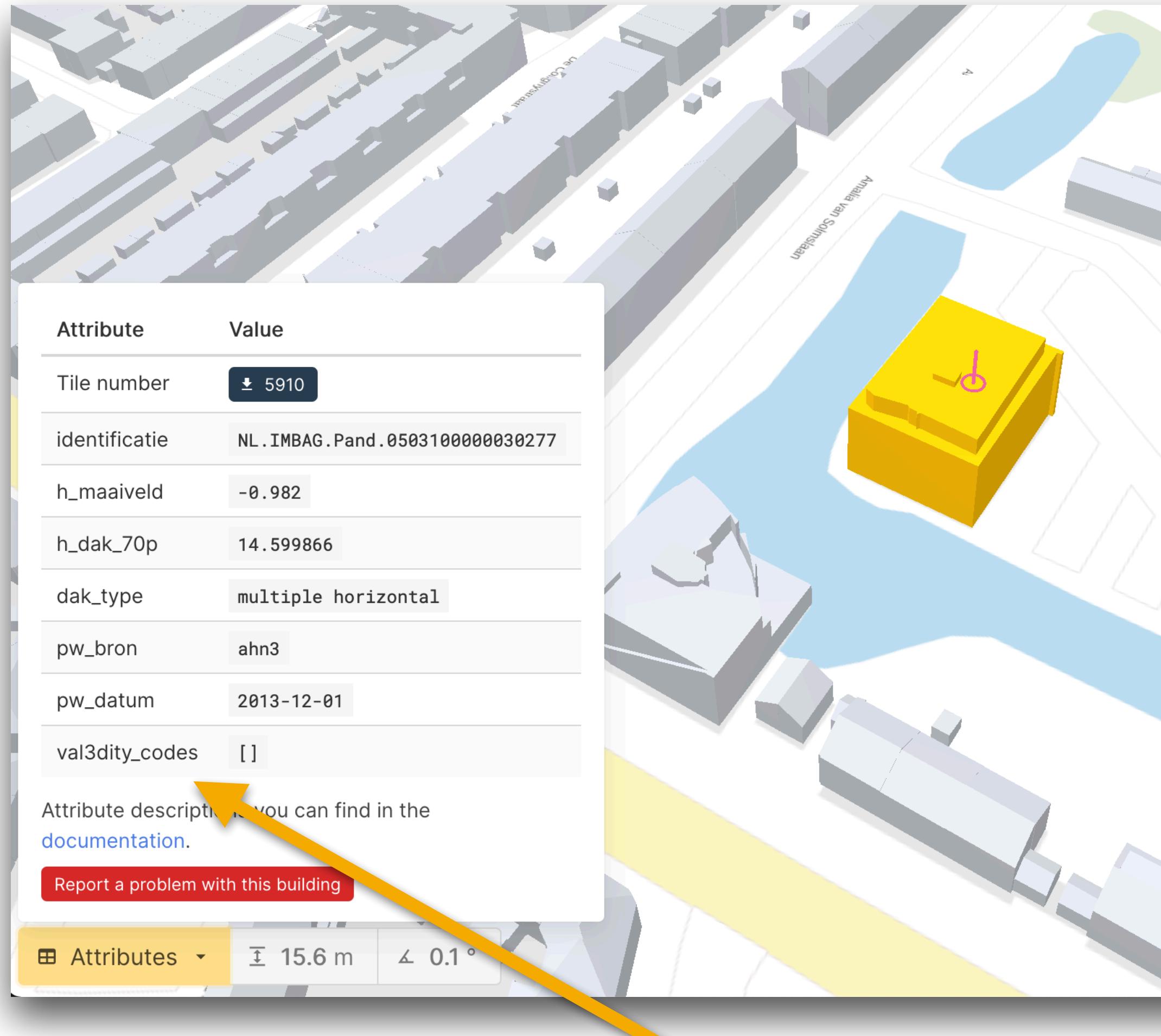
File	Format	Size	Version
<a href="3dbag_v21031_7425c21b.zip">3dbag_v21031_7425c21b.zip</a>	PostgreSQL ?	>20GB (>75GB in PostgreSQL)	v21.03.1

**Stats last 2 weeks (~32000 downloads)**

CityJSON	56%
OBJ	23%
GPKG	21%
PostgreSQL	<1%

# Very careful about geometric validity

~98% valid

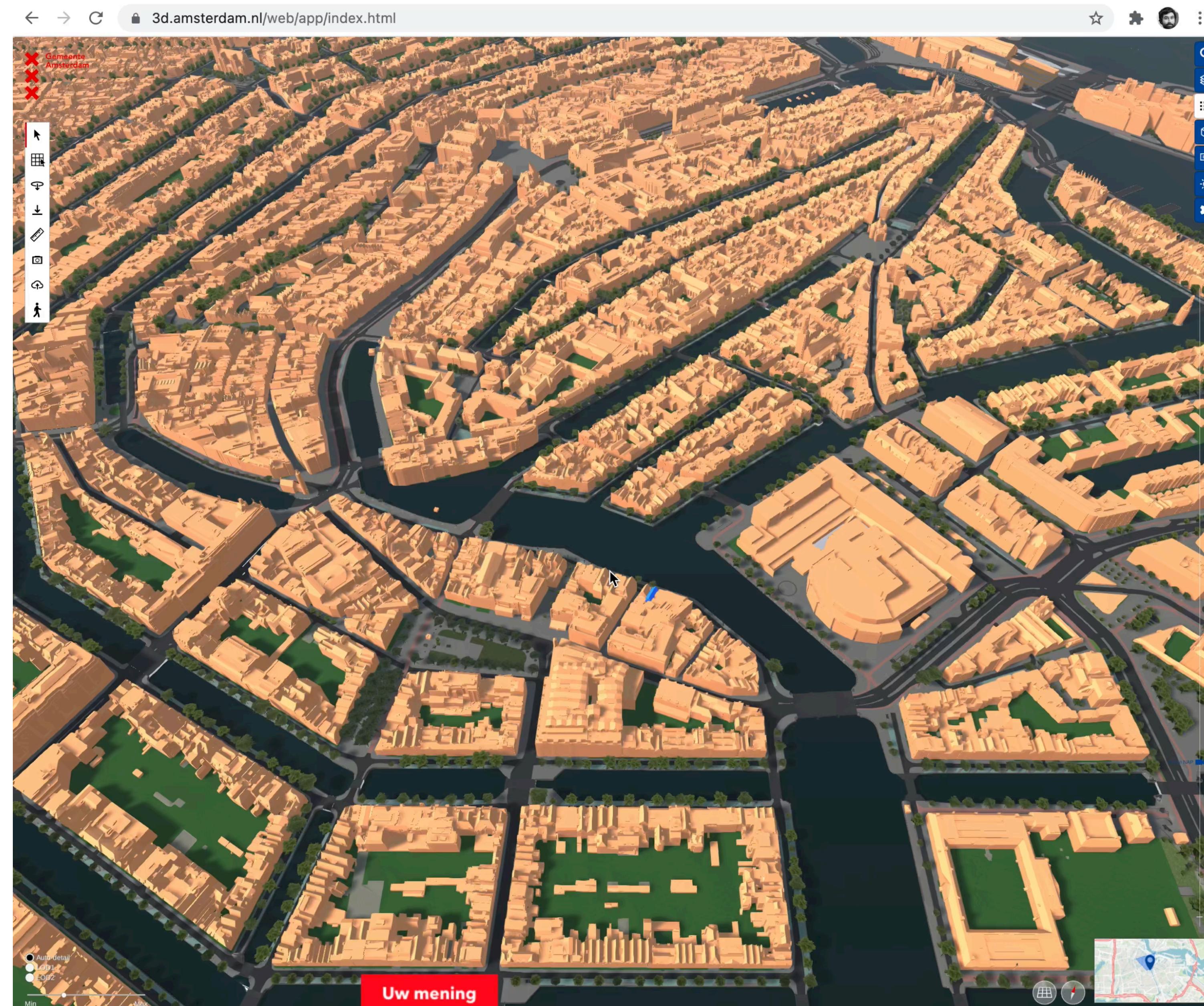


A screenshot of a GitHub repository page for "tudelft3d/val3dity". The repository is described as "Validation of 3D primitives according to the international standard ISO19107". It has 12 pulls, 45 stars, and 13 forks. The "Code" tab is selected, showing a list of commits:

- hugoledoux Add brew install + reformat code ... on 18 May 2020 (1,412)
- cmake Revert "update visual studio files 2" 4 years ago
- data Update CityJSON files for pytest to vers... 2 years ago
- docs Reformat code: now all errors in a std::m... 17 months ago
- misc Add termcast in svg, looks neater, innit? 17 months ago
- report\_browser Add warning to web-report when CityG... 17 months ago
- src Bump version to v2.2.0 🎉 17 months ago
- tests test: yaml.FullLoader is deprecated 17 months ago
- thirdparty Revert "update visual studio files 2" 4 years ago
- tools/python Revert "update visual studio files 2" 4 years ago
- vs\_build update vs\_build files 3 years ago
- .appveyor.yml fix eigen3 include dir 17 months ago
- .gitignore Update gitignore file 2 years ago
- .travis.yml Fix docker build on travis 2 years ago

The repository has 18 releases, with the latest being "new report, n..." on 14 May 2020. There are also sections for "About", "Readme", "GPL-3.0 License", "Releases", "Packages", and "Publish your first package".

# Amsterdam put our data in their Unity online platform within days



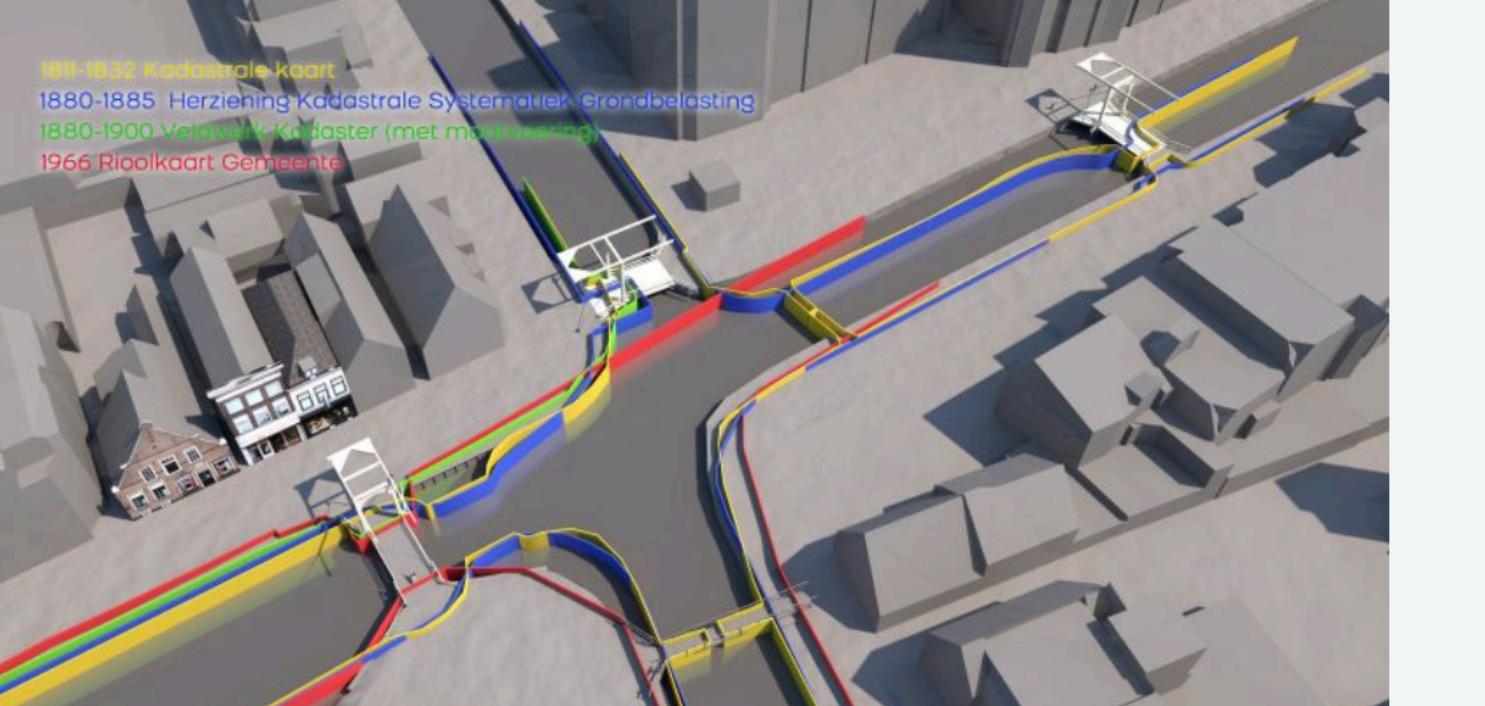
# People outside our field used the data

## Historical cadastre

**LinkedIn** Personen  ... ...

 **Andy Benjamins**  
3D Modelleur / 3D Designer / Game Developer/ Gis Medewerker at Kadaster  
3 w · Bewerkt

Historische Kadastrale grenzen in 3D:  
Voor mijn project Virtueel Hoogeveen ben ik bezig om oude kadastrale grenzen vast te leggen en te digitaliseren. Hoe krijg je nu alles goed op de juiste locatie?, dit is een grote uitdaging.  
Georefereren van de oude kaarten blijkt toch lastig, eigenlijk onmogelijk omdat de tekeningen toch behoorlijk afwijken en niet maatvast zijn. Gelukkig zijn er veel oude panden bewaard gebleven en met behulp van de LOD2.2 dataset van de TU delft kon ik toch vrij nauwkeurig (bij benadering) de kaarten op de plek leggen. Nu nog uitvogelen welke dicht bij de waarheid ligt.  
pijplijn: FME, ArcGispro, CityEngine, Cinema4D en Unreal Engine.



68 · 10 commentaren

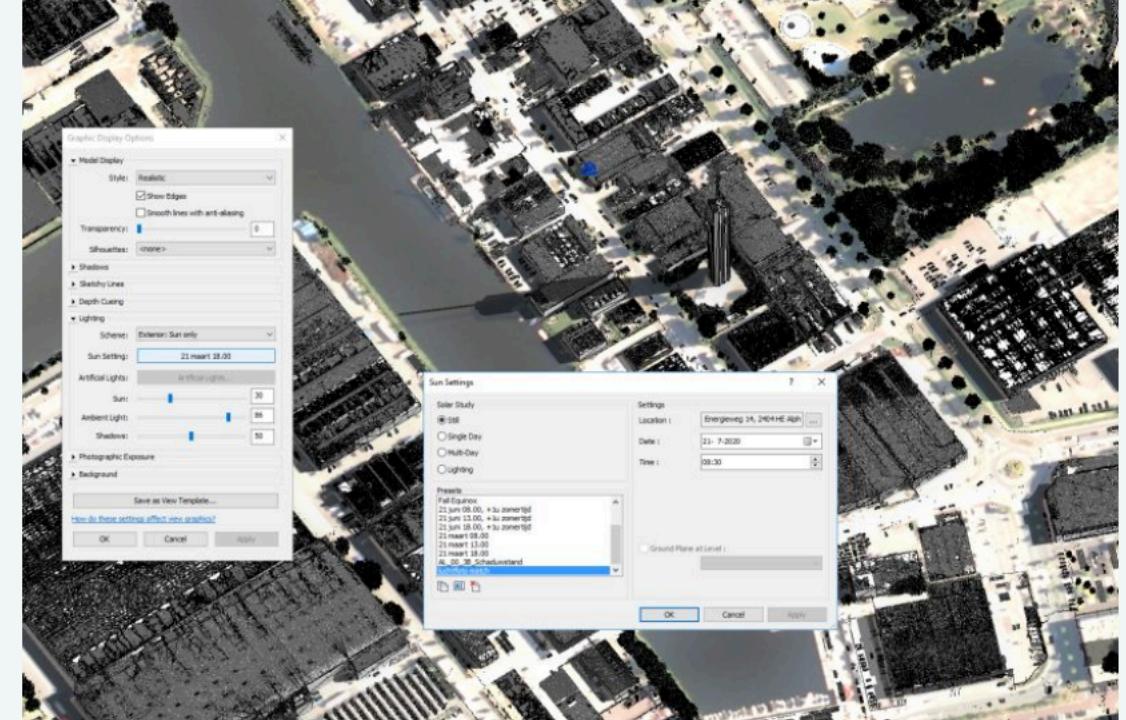
## Shadow analysis

**LinkedIn** Personen  ... ...

 **Ekko Nap**  
Architect, BIM R&D, Revit Coach at Groosman  
6 mnd

Vandaag even gespeeld met de nieuwe #3DBAG LOD 2.2. De schaduw van de Alphense zendmast om half 9 's ochtends klopt heel aardig, in luchtfoto 2020, AHN3 Pointcloud en 3D BAG, in #Revit.  
**Krijn Geevers** Vilmane Daulenskyte **Nick Tettero** niet te hard van stapel lopen, het komt nog binnen als complete Mesh Tiles. Zodra ik de CityJSON in ons kavelpaspoort verwerkt heb krijgen we gewoon gebouwobjecten met alle data die jullie gewend zijn.

Credits voor **Balázs Dukai**, **Ravi Peters**, **Jantien Stoter**, **Stelios Vitalis**, Jordi van Liempt ziet er erg bruikbaar uit!



49 · 8 commentaren

... ... ...

 **Gerwin Korpershoek**  
Maarten Vroegeindeweij gaan ze je nu inhalen :)  
Interessant Beantwoorden | 2 markeringen als interessant

5 mnd ...

# We reached non-geo people

Een kaart met 'body': alle gebouwen in Nederland in 3d +1

Door Julian Huijbregts  
Nieuwsredacteur  
Feedback • 09-04-2021 11:45 81

Een onderzoeks groep van TU Delft heeft een 3d-dataset en bijbehorende viewer van vrijwel alle tien miljoen gebouwen in Nederland gepubliceerd. Tweakers sprak met Ravi Peters, die het reconstructie-algoritme ontwikkelde om open data te vertalen naar 3d-modellen.

3D BAG is volgens de makers de eerste [nationale open dataset van 3d-gebouwen](#) op dit detailniveau. Niet alleen de blokstructuur van gebouwen is opgenomen, maar ook schuine dakvlakken zijn gemodelleerd. De makers noemen dat LoD 2.2, wat staat voor *level of detail*. Het is ook mogelijk om voor de lagere detailniveaus LoD 1.2 en LoD 1.3 te kiezen.

De 3d-modellen zijn volledig automatisch gegenereerd aan de hand van open data afkomstig uit de [Basisregistratie Adressen en Gebouwen](#) en het [Actueel Hoogtebestand Nederland](#). De BAG is een 2d-kaart van Nederland met daarop alle gebouwen weergegeven en het AHN is een 3d-puntenwolk, die vanuit vliegtuigen met laserscanners is ingewonnen.

Door de data van beide te combineren, is 3D BAG tot stand gekomen. Het is een project van zes onderzoekers van de [3D geoinformation-onderzoeks groep](#) van TU Delft. Dat is een multidisciplinaire groep, bestaande uit informatici, geomatica-ingenieurs en geografen.

De onderzoeks groep [werkte eerder al samen met het Kadaster](#) aan de publicatie van een 3d-bestand van Nederland. Daarbij werd ook gebruikgemaakt van luchtfoto's waar ook hoogteinformatie uit afgeleid kan worden, maar de 3d-data die het Kadaster vorig jaar publiceerde, heeft een lager detailniveau en er is geen 3d-webviewer om die gemakkelijk te bekijken.



<https://www.cyclomedia.co...t/data-visualization/3dnl>

jdh009 @Robonus • 9 april 2021 13:09 • Rapporteer +1  
Gezien Cyclomedia een commercieel bedrijf is zal deze dataset waarschijnlijk (lees: helaas) niet als open data beschikbaar komen.

Robonus @jdh009 • 9 april 2021 13:10 • Rapporteer +1  
Nee dat klopt, hier zal een betaalde licentie voor nodig zijn, net als alle jaarlijkse "streetview" opnames en hun landsdekkende lidarmodel vanaf straatniveau.

CopyCatz 9 april 2021 12:01 • Rapporteer +2  
Ik zou zeggen even de database doorzetten naar Asobo Studio, dit is perfect voor flight simulator 😊

rclinden @CopyCatz • 9 april 2021 15:17 • Rapporteer 0  
Dat zou geweldig zijn.

ThePendulum @CopyCatz • 9 april 2021 22:40 • Rapporteer 0  
Ik dacht al, dit komt precies te laat. Volgende week komt er een World Update voor Frankrijk en de Benelux uit, dus zal het wel een hele poos duren voordat we weer aan de beurt zijn.

DFKT T 9 april 2021 12:20 • Rapporteer +2  
Echt heel tof dat dit beschikbaar is! Voor architecten fantastisch om hun ontwerpen in een realistische omgeving te kunnen renderen. Voor andere adviseurs betrokken bij het (nieuw)bouwproces om allerhande gebouwaspecten en zaken gerelateerd aan de bebouwde omgeving te beoordelen: dient de omgeving als bebouwd of onbebouwd te worden geklassificeerd? Rekenen aan te verwachten windhinder in/voor de omgeving. Bezonningsstudies in en om gebouwen, etc.  
  
Off-topic: al die reacties over of een woning of gebouw wel of niet goed in het model staat, zijn echt niet relevant bij dit artikel. Het gaat hier om de beschikbaarheid en verwerking van deze data, niet de accuraatheid (die wordt bepaald door de input voor dit project, uit BAG en AHN)  
[Reactie gewijzigd door DFKT op 9 april 2021 12:22]

# thank you.

Hugo Ledoux

[h.ledoux@tudelft.nl](mailto:h.ledoux@tudelft.nl)  
[3d.bk.tudelft.nl/hledoux](http://3d.bk.tudelft.nl/hledoux)



<https://cityjson.org>

3D BAG by tudelft3d

<https://3dbag.nl>