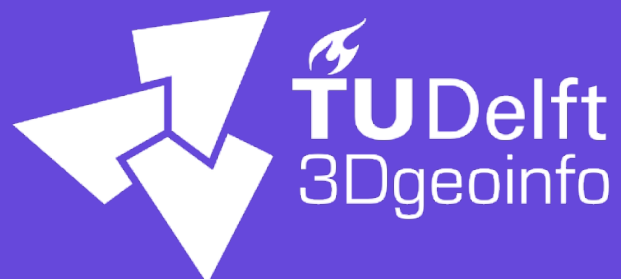


Storing the 3D topography of the Netherlands with CityJSON

Hugo Ledoux

Open Geodag
Amersfoort, the Netherlands
2019-10-02



Home - CityJSON

https://www.cityjson.org

CityJSON

Search CityJSON

Home

[What is CityJSON?](#)

[News](#)

[Specifications](#)

[Schemas](#)

[Datasets](#)

[Extensions](#)

[Software](#)

[Help](#)

CityJSON

A JSON-based encoding for 3D city models

[Getting started](#)[Specifications \(v1.0.1\)](#)

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 APIs supporting it can be quickly built, and [several](#) have been created already.

We believe that you should use CityJSON because:

- 1 its simplicity means that it is already supported by [several software](#)
- 2 you can in one-click convert CityGML files to CityJSON files, and vice versa, with the open-source tool [citygml-tools](#); we even have a [tutorial](#)
- 3 files are on average [6X more compact](#) than their CityGML equivalent
- 4 there is a [web-viewer](#) where you can drag-and-drop a file
- 5 you can easily manipulate files with [cjio](#), you can for instance merge files, remove/filter objects, change the CRS, manage the textures, etc.
- 6 you can *easily* define [Extensions](#) to the core model
- 7 its development is [open on GitHub](#), it is supported by a vibrant community, and everyone is welcome to contribute

2

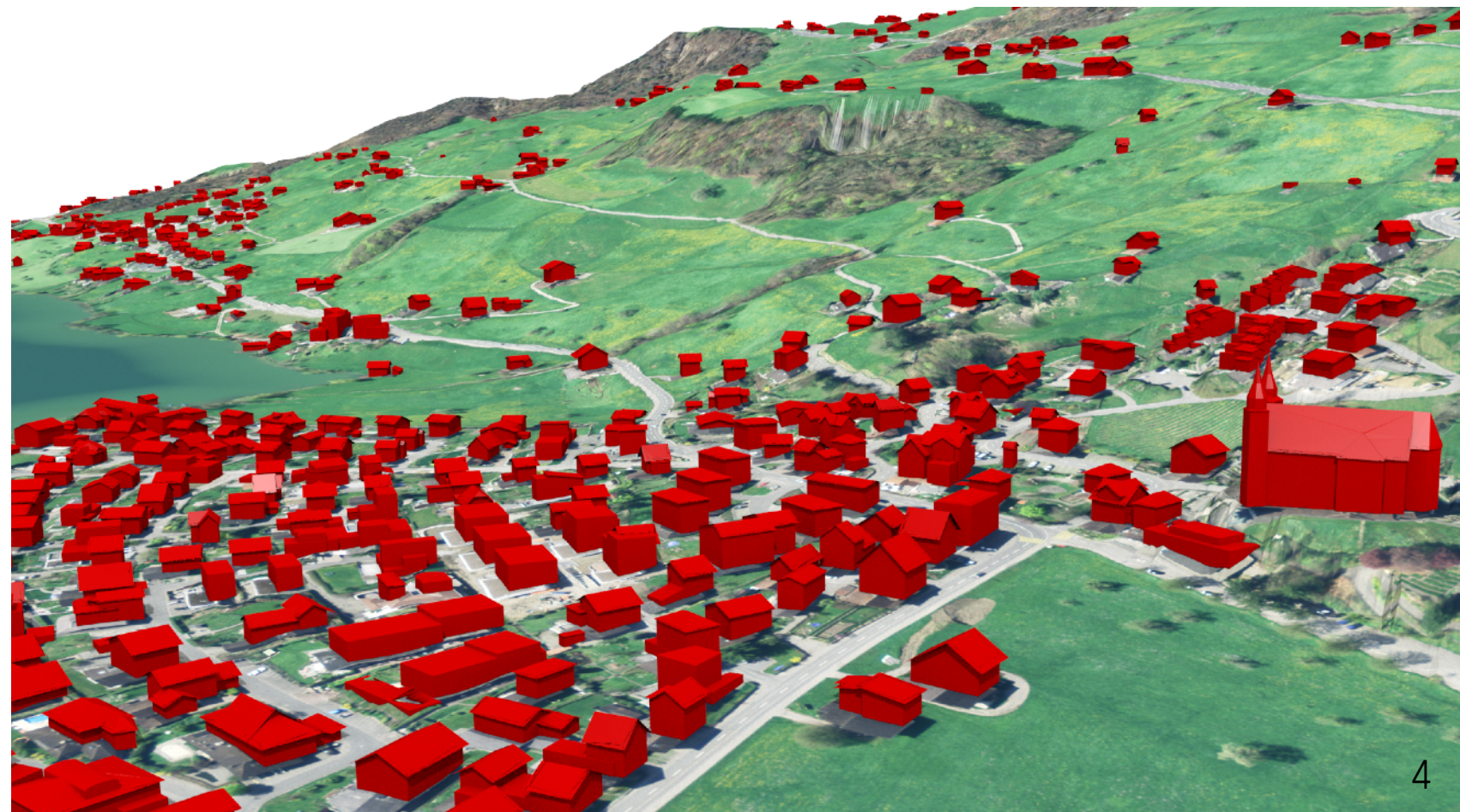
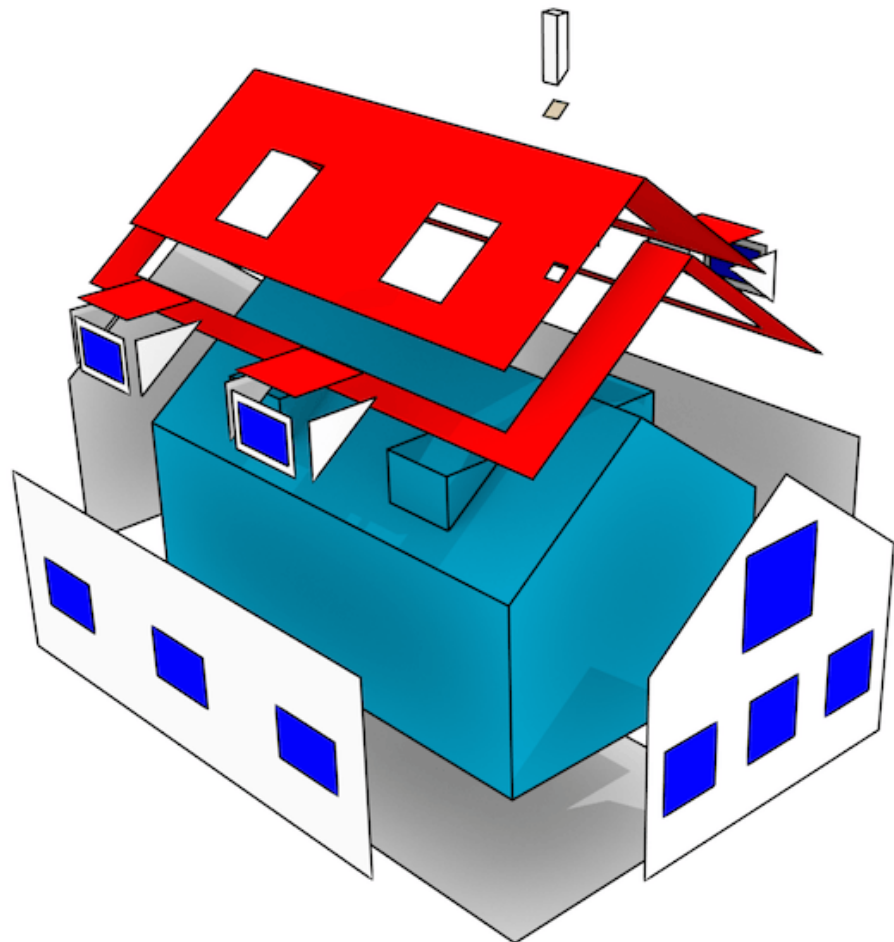
Our goal

storing the 3D topography of NL (IMGeo) with it

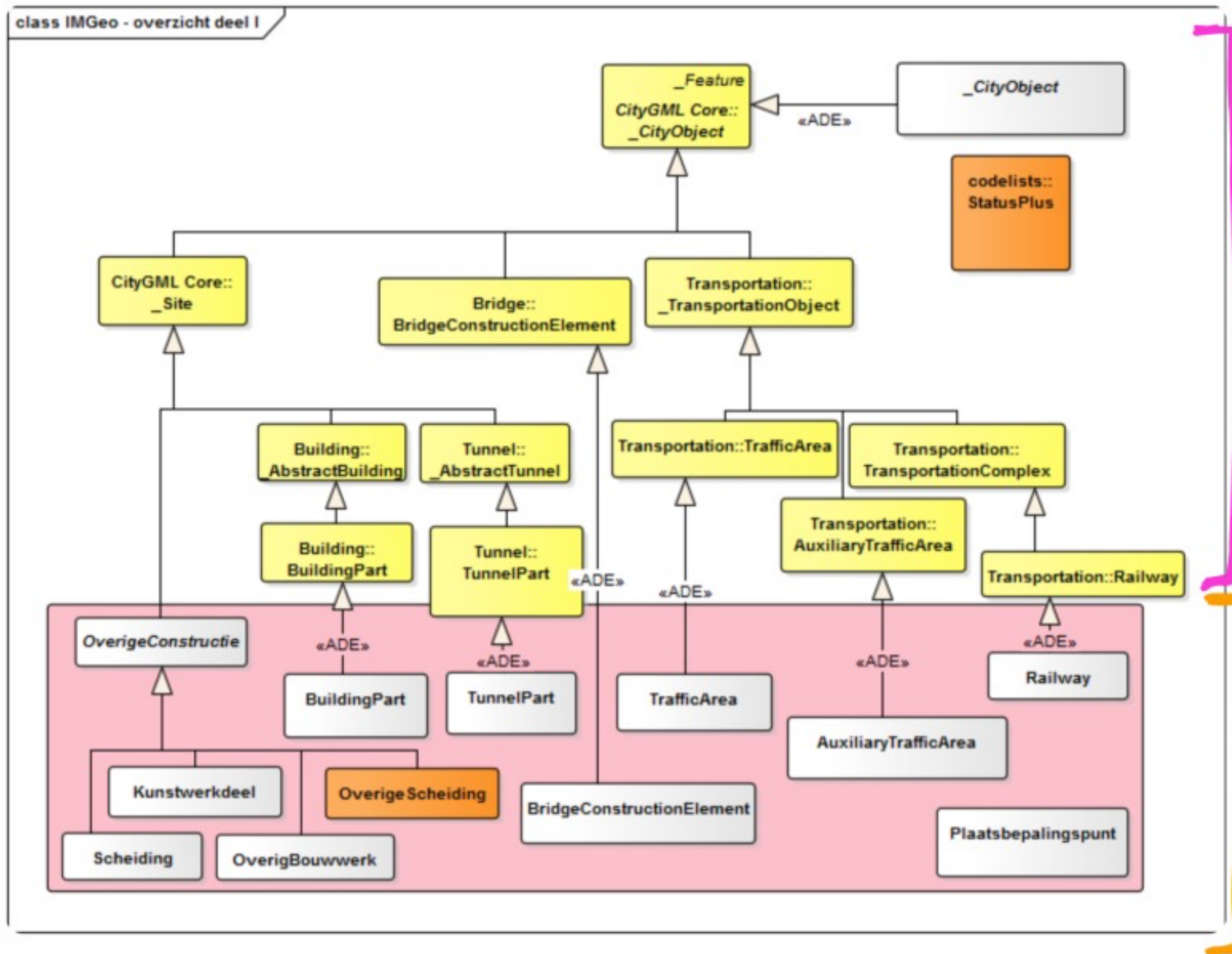
IMGeo 3D: stored since ~2012 with a CityGML ADE



international standard (from OGC) for representing and storing 3D city models



IMGeo 3D: stored since ~2012 with a CityGML ADE

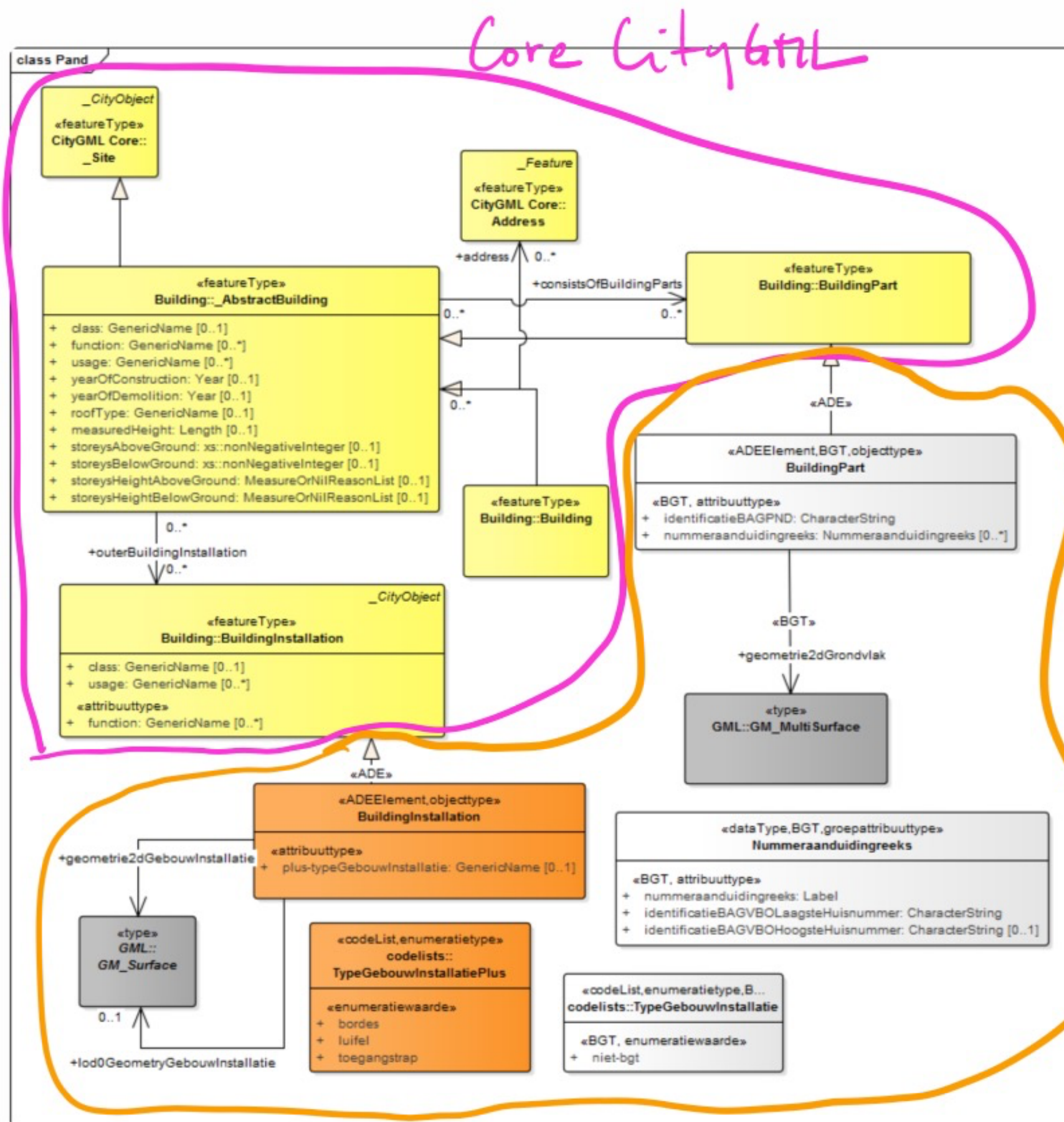


CityGML v2.0
core model

Application
Domain
Extension
ADE

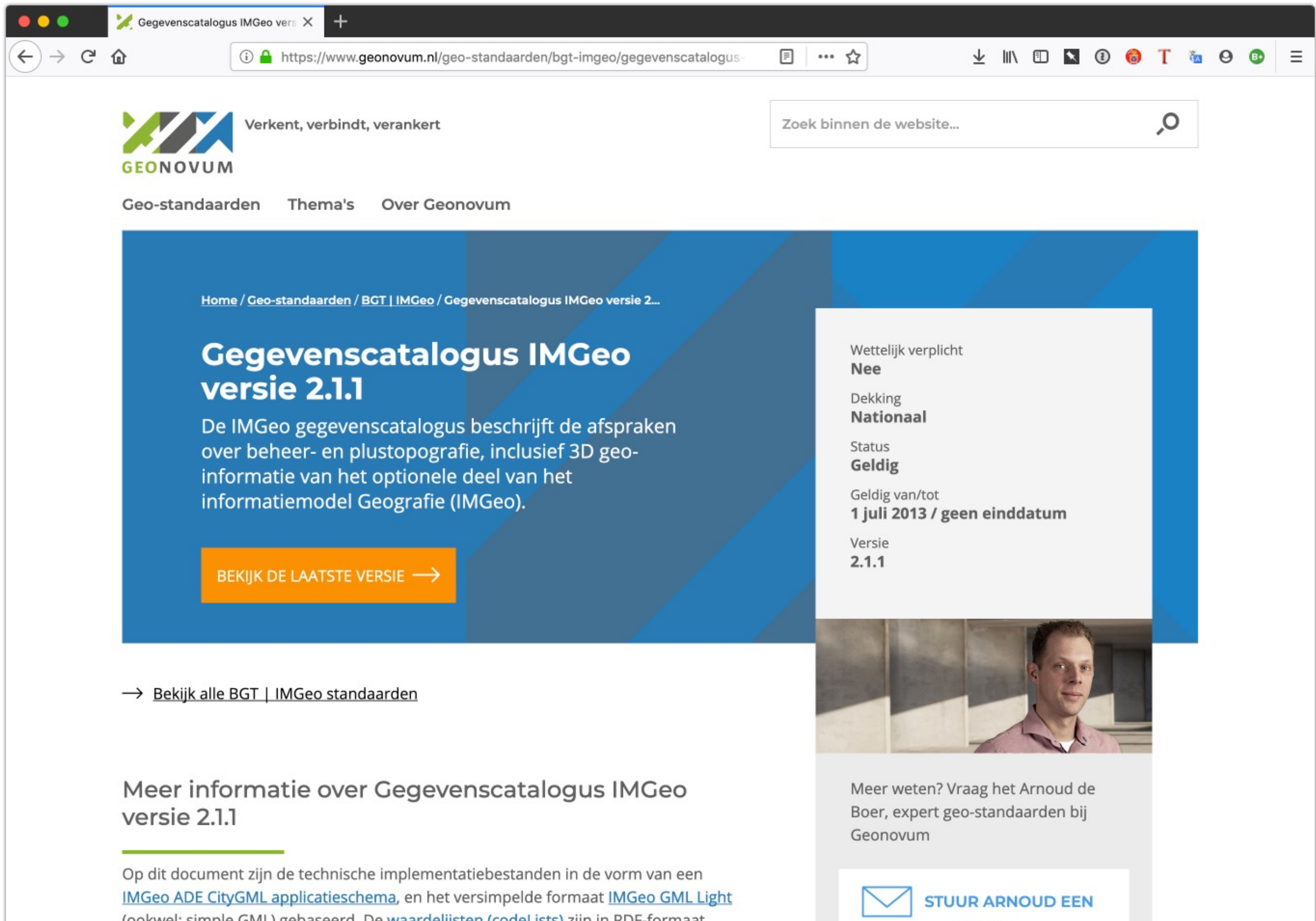
(extension to
the core model)

IMGeo 3D: stored since ~2012 with a CityGML ADE



IMGeo
ADE
for
Building

IMGeo 3D: stored since ~2012 with a CityGML ADE



Complete NL automatically with 3dfier!



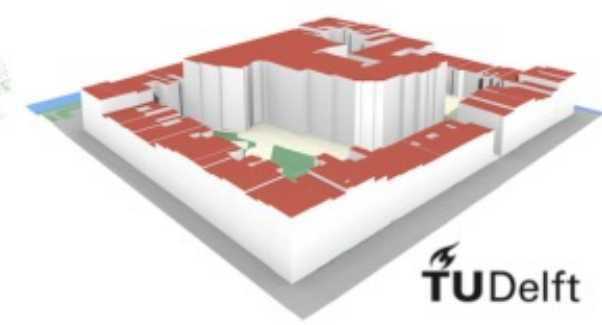
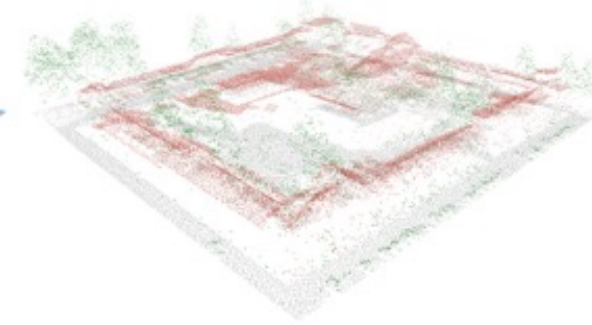
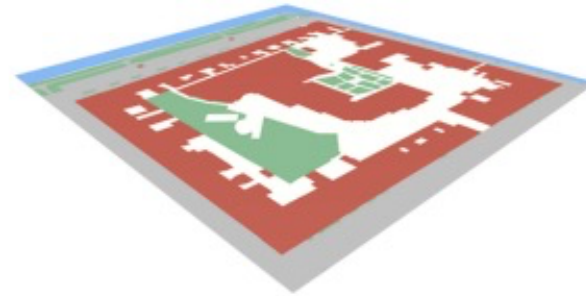
2D data (topographic map)

+

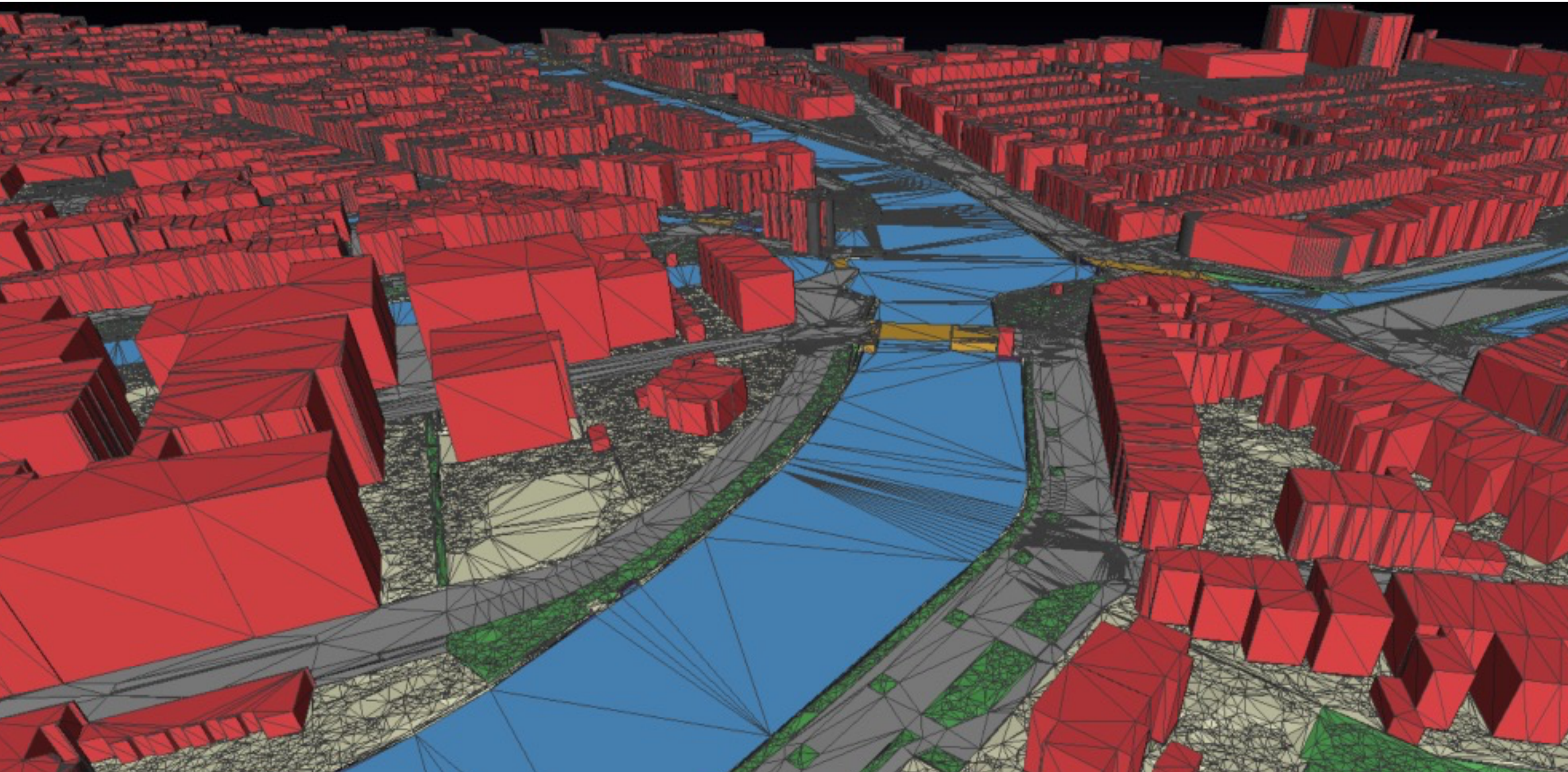
Elevation dataset (lidar)

=

3D city model



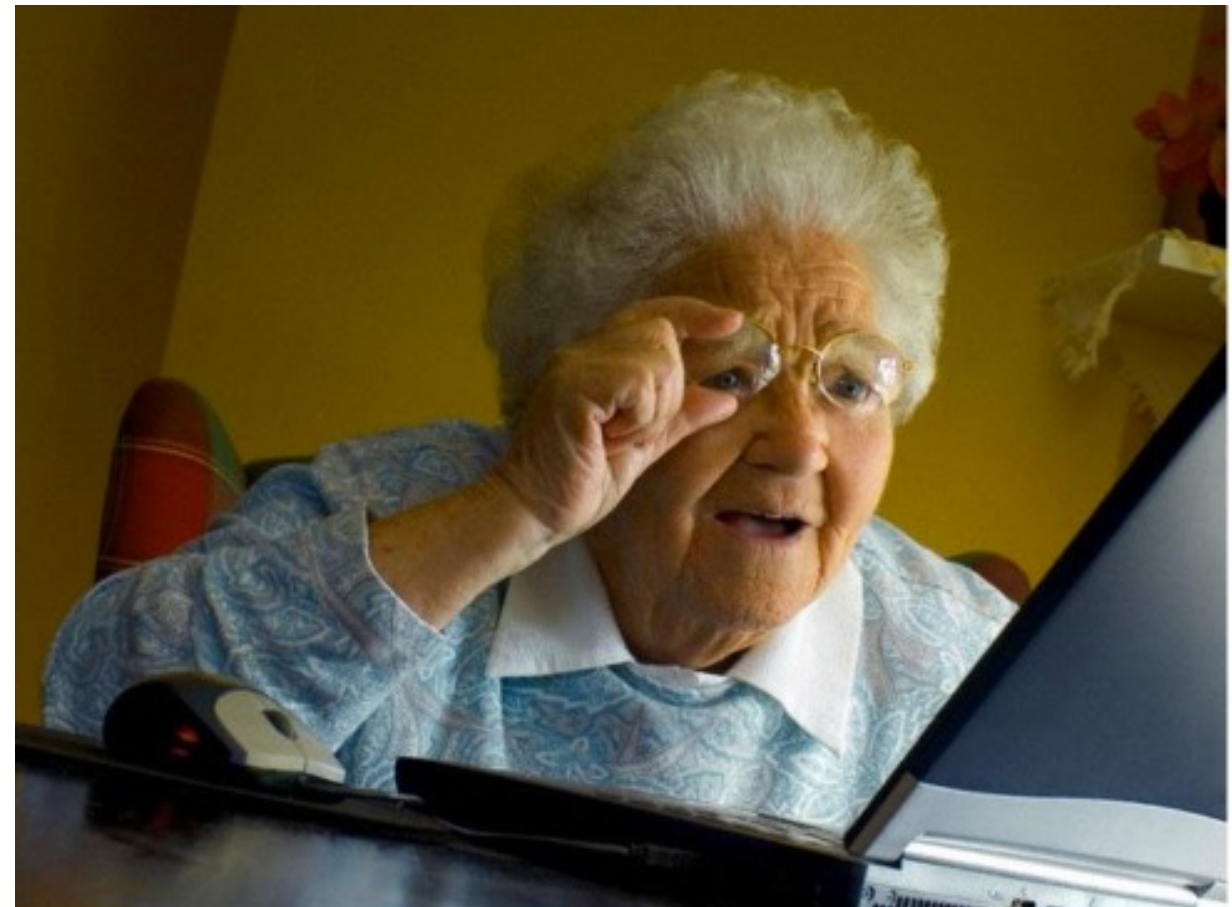
TU Delft



BUT

CityGML-ADE files are very complex and **HUGE**

- ➔ virtually no software support
- ➔ little use of the resulting files in practice
- ➔ no parsers in JavaScript (thus no web support)
- ➔ did I mention files are HUGE and complex?

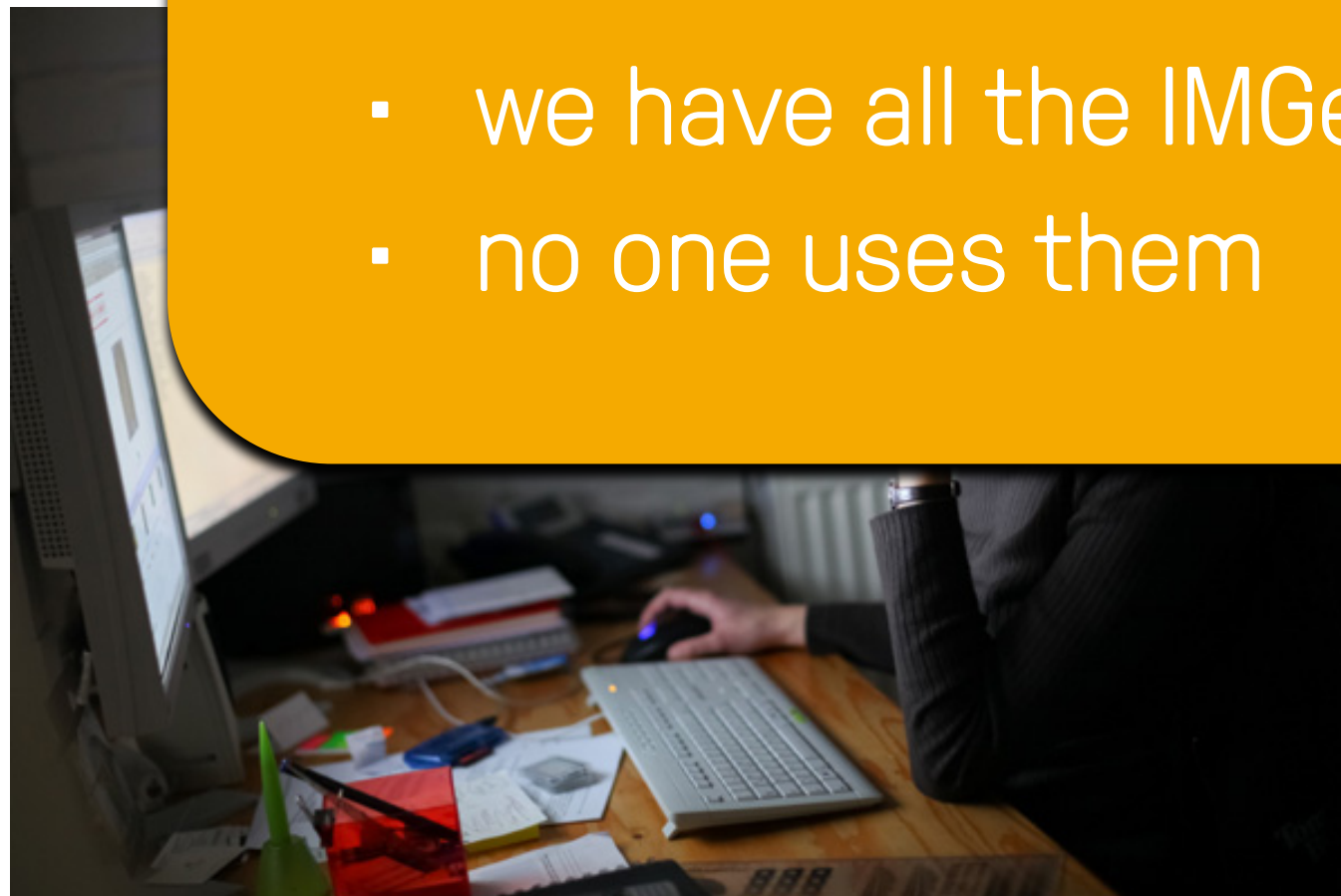


CityGML-ADE files are very complex and **HUGE**

- ➔ virtually no software support
- ➔ little use of the resulting files in practice
- ➔ no parsers in JavaScript (thus no web support)

In brief

- we have all the IMGeo 3D files (CityGML)
- no one uses them



Home - CityJSON

https://www.cityjson.org

CityJSON

Search CityJSON

Home

What is CityJSON?

News

Specifications

Schemas

Datasets

Extensions

Software

Help

{}

CityJSON

A JSON-based encoding for 3D city models

Getting started

Specifications (v1.0.1)

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 APIs supporting it can be quickly built, and [several](#) have been created already.

We believe that you should use CityJSON because:

1

its simplicity means that it is already supported by [several software](#)

2

you can in one-click convert CityGML files to CityJSON files, and vice versa, with the open-source tool [citygml-tools](#); we even have a [tutorial](#)

3

files are on average [6X more compact](#) than their CityGML equivalent

4

there is a [web-viewer](#) where you can drag-and-drop a file

5

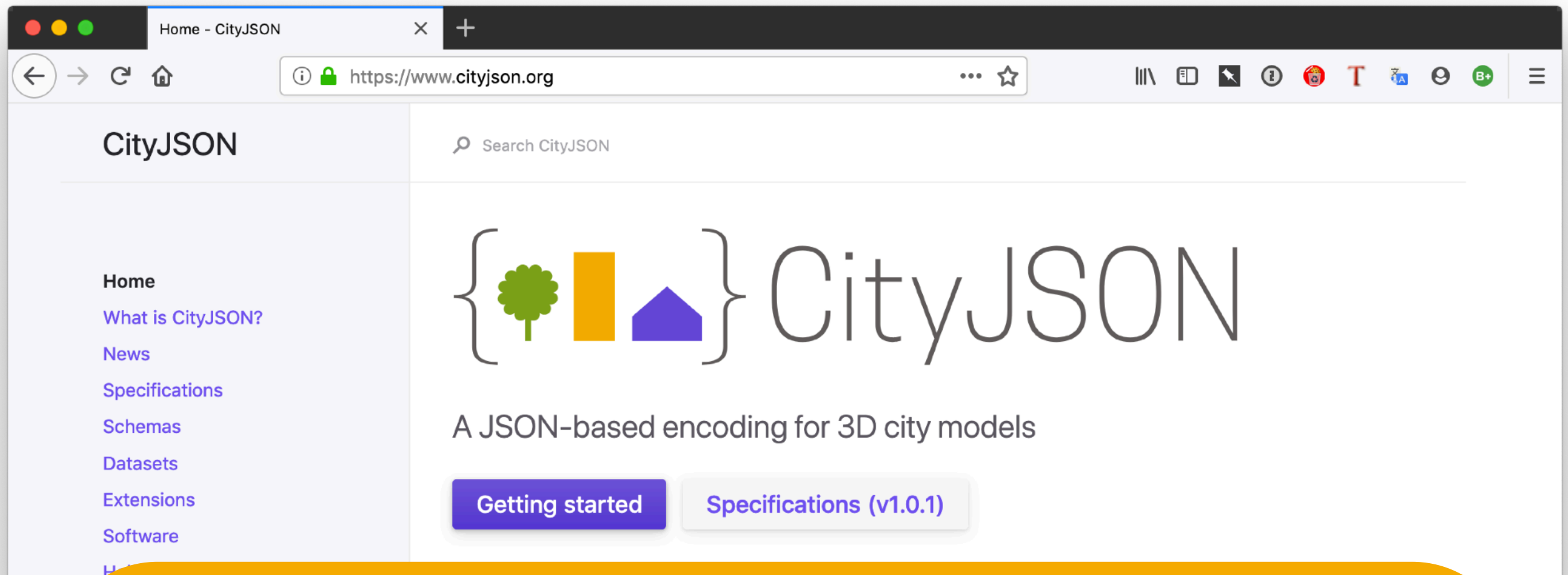
you can easily manipulate files with [cjio](#), you can for instance merge files, remove/filter objects, change the CRS, manage the textures, etc.

6

you can *easily* define [Extensions](#) to the core model

7

its development is [open on GitHub](#), it is supported by a vibrant community, and everyone is welcome to contribute



1. JSON encoding for 3D city models
2. designed with programmers in mind
3. ready for the web
4. 6X more compact than CityGML
5. Extensions can be easily defined and used

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

all programming language support
JSON “natively”

Python parser is simple



```
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

Drop your file here,
or click to open

show
edges

▼

DenHaag_01

GUID_DBDABF53-7DD5-4C2F-BE7F-51F29A0CBA16_1

GUID_8CE54418-E2F7-49A7-9A8D-C3D172BA62C4_2

GUID_13974D93-CB4F-4B5A-AB1E-577DD9928CF2_3

GUID_8CE54418-E2F7-49A7-9A8D-C3D172BA62C4_1

GUID_47061B3E-C490-4882-806A-EC084006045F

GUID_47061B3E-C490-4882-806A-EC084006045F_1

GUID_47061B3E-C490-4882-806A-EC084006045F_2

GUID_13974D93-CB4F-4B5A-AB1E-577DD9928CF2_2

GUID_13974D93-CB4F-4B5A-AB1E-577DD9928CF2_1

GUID_3D7D60B9-8F3A-4D3B-A3E5-CD9B5565A5B2

GUID_F784906B-FD5A-4F81-BACF-79C3566B2789

GUID_F784906B-FD5A-4F81-BACF-79C3566B2789_1

GUID_F784906B-FD5A-4F81-BACF-79C3566B2789_2

GUID_F784906B-FD5A-4F81-BACF-79C3566B2789_3

GUID_1280E44F-7EFD-4BEA-8E79-3124934BD6A0_4

GUID_1280E44F-7EFD-4BEA-8E79-3124934BD6A0_3

GUID_1280E44F-7EFD-4BEA-8E79-3124934BD6A0_2

GUID_1280E44F-7EFD-4BEA-8E79-3124934BD6A0_1

GUID_DCAE0ACB-5F38-4AEA-80D2-7242B193AE70

Attribute

Value

id

GUID_EA492E2E-6138-422C-B674-230CCD5E4DF0_2

roofType

1000

RelativeEavesHeight

7.472

RelativeRidgeHeight

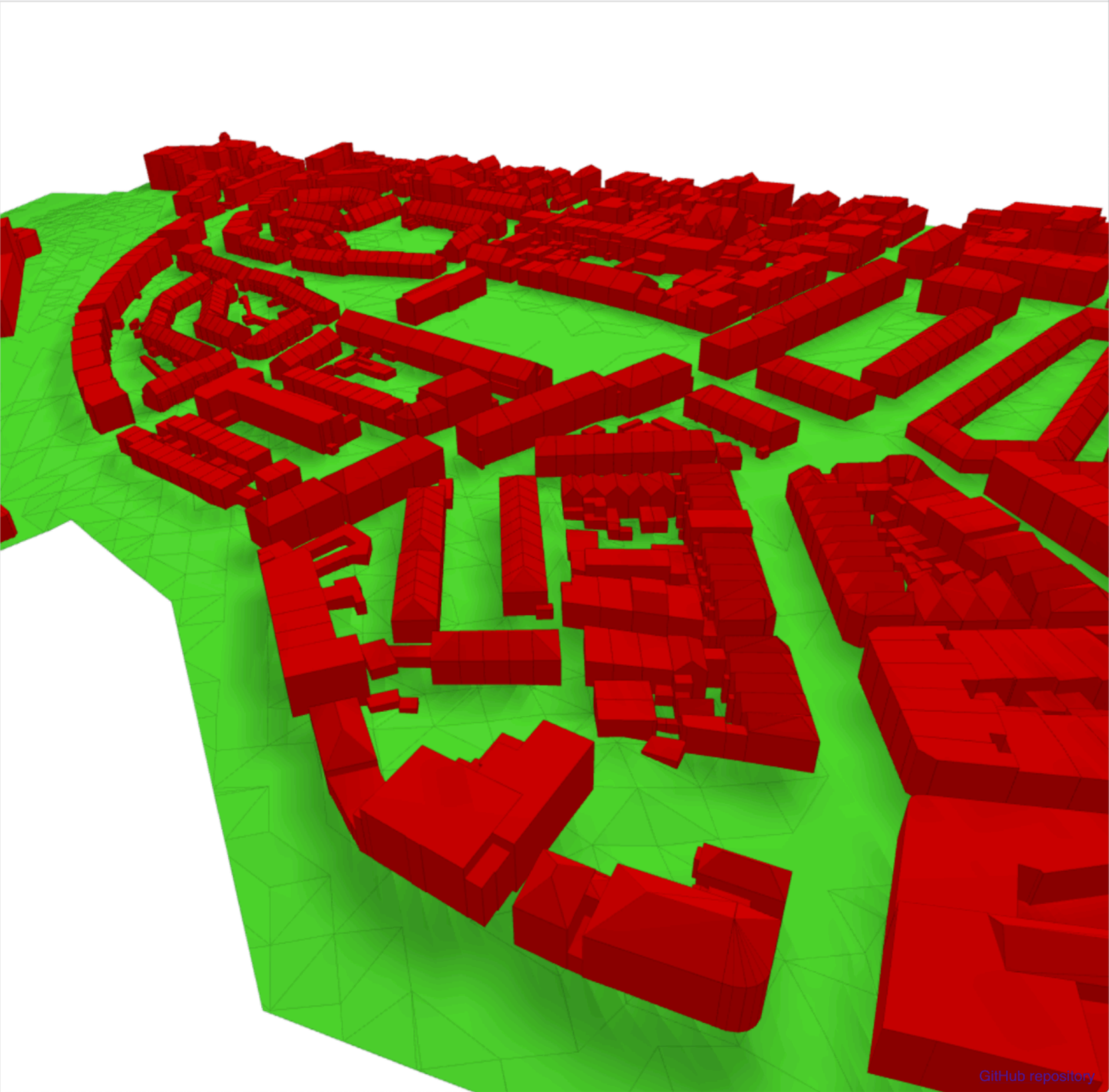
7.472

AbsoluteEavesHeight

14.297

AbsoluteRidgeHeight

14.297



3D visualization of a city model. The buildings are represented as red rectangular blocks of varying heights and orientations, clustered together. They are situated on a green, textured terrain that appears to be a map of a city area. The perspective is from an elevated angle, looking down at the city layout.

GitHub repository

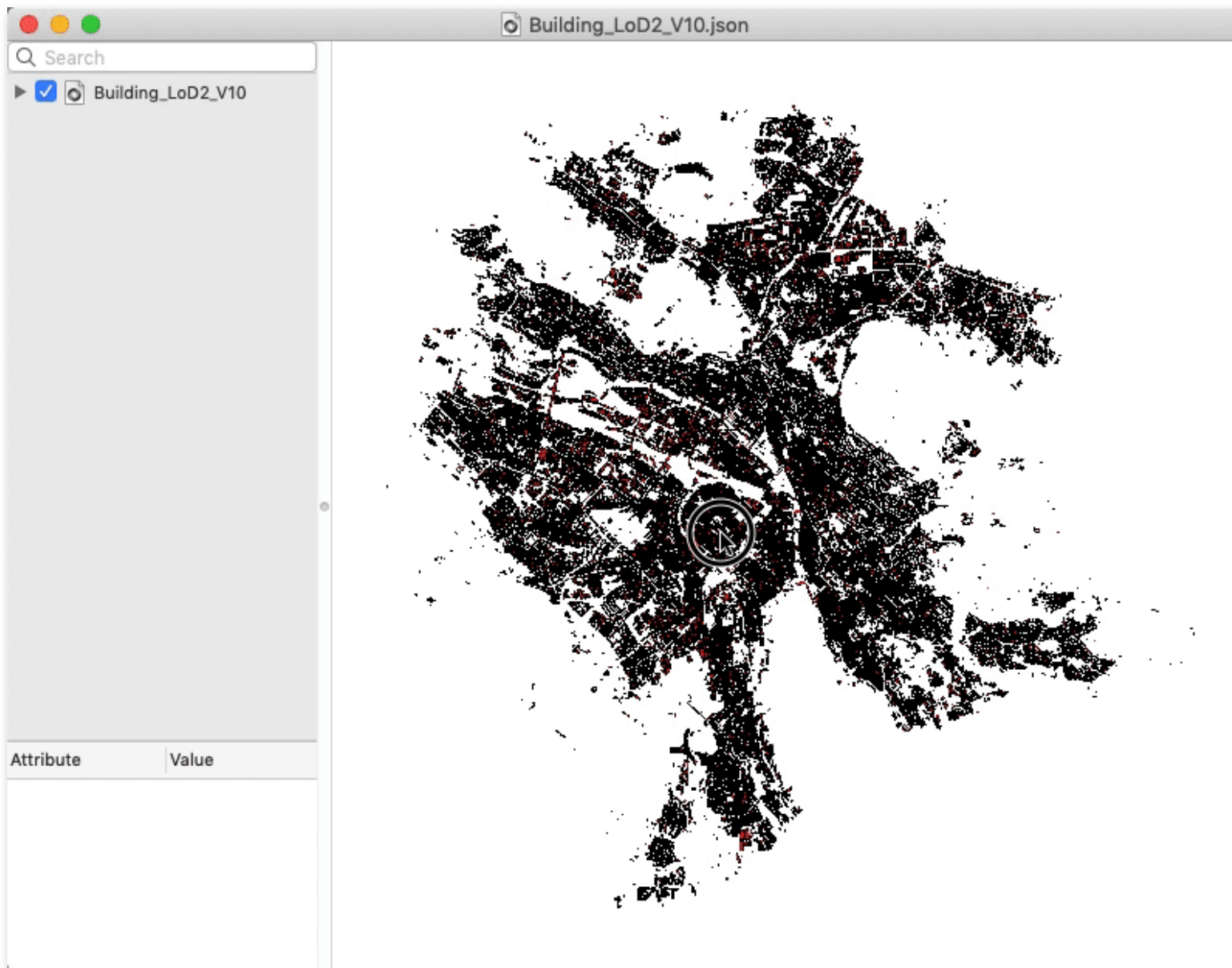
Compression example: Zürich LoD2 buildings

CityGML = 3.0GB

(but 1GB of spaces/CRs/tabs!)

CityJSON = 292MB

Compression == 7.1X



**IMGeo-3D with
CityJSON Extension?**

CityJSON Extension

CityJSON

🔍 Search CityJSON

[Home](#)
[What is CityJSON?](#)
[News](#)
[Specifications](#)
[Schemas](#)
[Datasets](#)
[Extensions](#)
[Software](#)
[Help](#)

Extensions

CityJSON uses [JSON Schemas](#) to document and validate the data model, schemas should be seen as basically validating the syntax of a JSON document.

A CityJSON *Extension* is a JSON file that allows us to document how the core data model of CityJSON may be extended, and to validate CityJSON files containing new objects and/or attributes. This is conceptually akin to the *Application Domain Extensions* (ADEs) in CityGML; see Section 10.13 of the [official CityGML documentation](#).

See the [Extensions specifications for v1.0.1](#)

Registry of current Extensions

Noise

Conversion of the CityGML Noise ADE
v1.0 | [schema](#) | [3D geoinformation TU Delft](#)

Linear Cell Complex

An extension that allows topological information of a city model to be stored as a Combinatorial Map based Linear Cell Complex
v0.2 | [schema](#) | [scientific article](#) | [3D geoinformation TU Delft](#)

Extension

a JSON file documenting how the core data model can be extended

CityJSON with an Extensions

```
{
  "type": "CityJSON",
  "version": "1.0",
  "extensions": {
    "Noise": {
      "url": "https://someurl.org/noise.json",
      "version": "1.0"
    }
  },
  "CityObjects": {
    "1234": {
      "type": "Building",
      "geometry": [
        {
          "type": "Solid",
          "lod": 2,
          "boundaries": [
            [ [0, 3, 2, 1]], ..., [[3, 0, 4, 7]] ]
          ]
        }
      ],
      "attributes": {
        "roofType": "pointy",
        "class": "22",
        "+noise-buildingReflectionCorrection": {
          "value": 4.123,
          "uom": "dB"
        },
        "+noise-buildingLNightMax": {
          "value": 43.123,
          "uom": "dB"
        }
      }
    }
  },
}
```

```
"thesegment_1": {
  "type": "+NoiseCityFurnitureSegment",
  "toplevel": false,
  "geometry": [
    {
      "type": "MultiLineString",
      "lod": 0,
      "boundaries": [
        [2, 3, 5], [7, 5, 12]
      ]
    }
  ],
  "attributes": {
    "reflectionCorrection": {
      "value": 4.123,
      "uom": "dB"
    }
  },
  "parents": ["a_noisy_bench"]
},
"thesegment_2": {
  "type": "+NoiseCityFurnitureSegment",
  "toplevel": false,
  "geometry": [
    {
      "type": "MultiLineString",
      "lod": 0,
      "boundaries": [
        [2, 3, 5], [7, 2, 12]
      ]
    }
  ],
  "attributes": {
    "reflectionCorrection": {
      "value": 2.666,
      "uom": "dB"
    }
  }
}
```

CityJSON with an Extensions

```
{
  "type": "CityJSON",
  "version": "1.0",
  "extensions": {
    "Noise": {
      "url": "https://someurl.org/noise.json",
      "version": "1.0"
    }
  },
  "CityObjects": {
    "1234": {
      "type": "Building",
      "geometry": [
        {
          "type": "Solid",
          "lod": 2,
          "boundaries": [
            [ [0, 3, 2, 1]], ..., [[3, 0, 4, 7]] ]
          ]
        }
      ],
      "attributes": {
        "roofType": "pointy",
        "class": "22",
        "+noise-buildingReflectionCorrection": {
          "value": 4.123,
          "uom": "dB"
        },
        "+noise-buildingLNightMax": {
          "value": 43.123,
          "uom": "dB"
        }
      }
    }
  }
},
```

```
    "thesegment_1": {
      "type": "+NoiseCityFurnitureSegment",
      "toplevel": false,
      "geometry": [
        {
          "type": "MultiLineString",
          "lod": 0,
          "boundaries": [
            [2, 3, 5], [7, 5, 12]
          ]
        }
      ],
      "attributes": {
        "reflectionCorrection": {
          "value": 4.123,
          "uom": "dB"
        }
      },
      "parents": ["a_noisy_bench"]
    },
    "thesegment_2": {
      "type": "+NoiseCityFurnitureSegment",
      "toplevel": false,
      "geometry": [
        {
          "type": "MultiLineString",
          "lod": 0,
          "boundaries": [
            [2, 3, 5], [7, 2, 12]
          ]
        }
      ],
      "attributes": {
        "reflectionCorrection": {
          "value": 2.666,
          "uom": "dB"
        }
      }
    }
  }
},
```

For IMGeo? We started a pilot with Geonovum

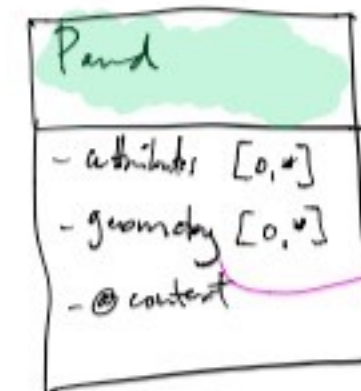
```
{  
  "type": "CityJSON_Extension",  
  "name": "NL3D",  
  "uri": "https://github.com/hugoledoux/nl3d/blob/master/schemas/extensions/nl_3d.json",  
  "version": "0.1",  
  "description": "NL3D data model as a CityJSON Extension.",  

```

```
  "extraCityObjects": {  
  
    "+Pand": {  
      "allOf": [  
        { "$ref": "../cityobjects.schema.json#/_AbstractCityObject" },  
        {  
          "properties": {  
            "type": { "enum": [ "+Pand" ] },  
            "toplevel": { "type": "boolean" },  
            "@context": {  
              "type": "string",  
              "format": "uri-reference"  
            },  
            "attributes": {  
              "properties": {  
                "yearOfConstruction": { "type": "integer" },  
                "gebouwnummer": { "type": "string" },  
                "statusOmschr": { "type": "string" },  
                "hoogste_bouwlaag": { "type": "integer" },  
                "hoogste_bouwlaag": { "type": "integer" },  
                "laagste_bouwlaag": { "type": "integer" },  
                "aantalBouwlagen": { "type": "integer" }  
              }  
            },  
            "geometry": {  
              "type": "array",  
              "items": {  
                "oneOf": [  
                  { "$ref": "../geomprimitives.schema.json#/Solid" },  
                  { "$ref": "../geomprimitives.schema.json#/CompositeSolid" },  
                  { "$ref": "../geomprimitives.schema.json#/MultiSurface" }  
                ]  
              }  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```



include roots but
"bundled separately"



geom always
has several
LODs @
semantic
surfaces



roots are a separate
geom object, but not
semantically labelled

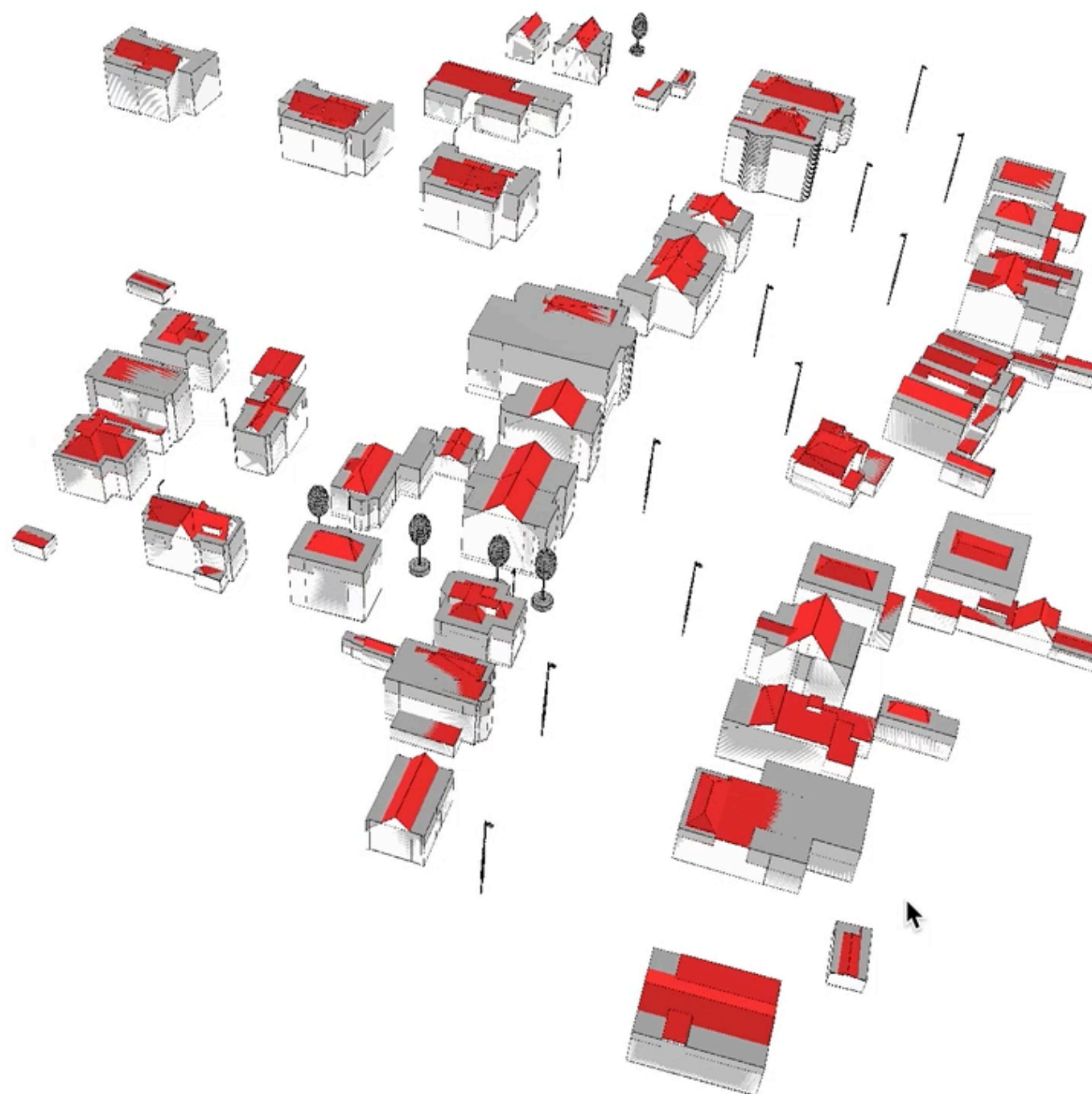
Search

all

- ▼ ☒ ☒ +Pand (ID_0599100000678129)
 - ☒ LoD (0)
 - ☒ LoD (1)
 - ▶ ☒ LoD (2)
- ▶ ☒ +Pand (ID_0599100000059771)
- ▶ ☒ +Pand (ID_0599100000376512)
- ▶ ☒ +Pand (ID_0599100010060731)
- ▶ ☒ +Pand (ID_0599100010003441)
- ▶ ☒ +Pand (ID_0599100000285796)
- ▶ ☒ +Pand (ID_0599100010005917)
- ▼ ☒ +Pand (ID_0599100000052206)
 - ☒ LoD (0)
 - ☒ LoD (1)
 - ▶ ☒ LoD (2)
- ▶ ☒ +Pand (ID_0599100000192858)
- ▶ ☒ +Pand (ID_0599100000110772)
- ▶ ☒ +Pand (ID_0599100010058319)
- ▶ ☒ +Pand (ID_0599100000133608)
- ▶ ☒ +Pand (ID_0599100000678099)
- ▶ ☒ +Pand (ID_0599100010010850)
- ▶ ☒ +Pand (ID_0599100010053785)
- ▶ ☒ +Pand (ID_0599100000264780)
- ▶ ☒ +Pand (ID_0599100010043999)
- ▶ ☒ +Pand (ID_0599100010002082)

Attribute

Value



Our (alpha) code/models/tests are open

The screenshot shows the GitHub repository page for `tudelft3d/nl3d`. The repository has 5 stars, 0 forks, and 0 pull requests. The main content area shows a list of files and folders, including `code`, `data/rotterdam`, `jsonld`, `misc`, `schemas`, `semantics`, and `README.md`. The `README.md` file is selected, showing the title "Draft data model for NL3D project" and the text "Example dataset all merged is /data/rotterdam/citvison/all.json".

Repository: `tudelft3d / nl3d`

Unwatch 5 Star 0 Fork 0

Code Issues 6 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. [Edit](#)

[Manage topics](#)

25 commits 1 branch 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

hugoledoux Improve schema of the extension Latest commit 3df94c9 4 days ago

<code>code</code>	Fix the trees	5 days ago
<code>data/rotterdam</code>	All versions of the new files	5 days ago
<code>jsonld</code>	First import	2 months ago
<code>misc</code>	New UML for what is done	5 days ago
<code>schemas</code>	Improve schema of the extension	4 days ago
<code>semantics</code>	first try at diagrams for semantic 1st layer	last month
<code>README.md</code>	New UML for what is done	5 days ago

README.md

Draft data model for NL3D project

Example dataset all merged is `/data/rotterdam/citvison/all.json`

thank you.

Hugo Ledoux

`h.ledoux@tudelft.nl`
`3d.bk.tudelft.nl/hledoux`

{} CityJSON

`https://cityjson.org`