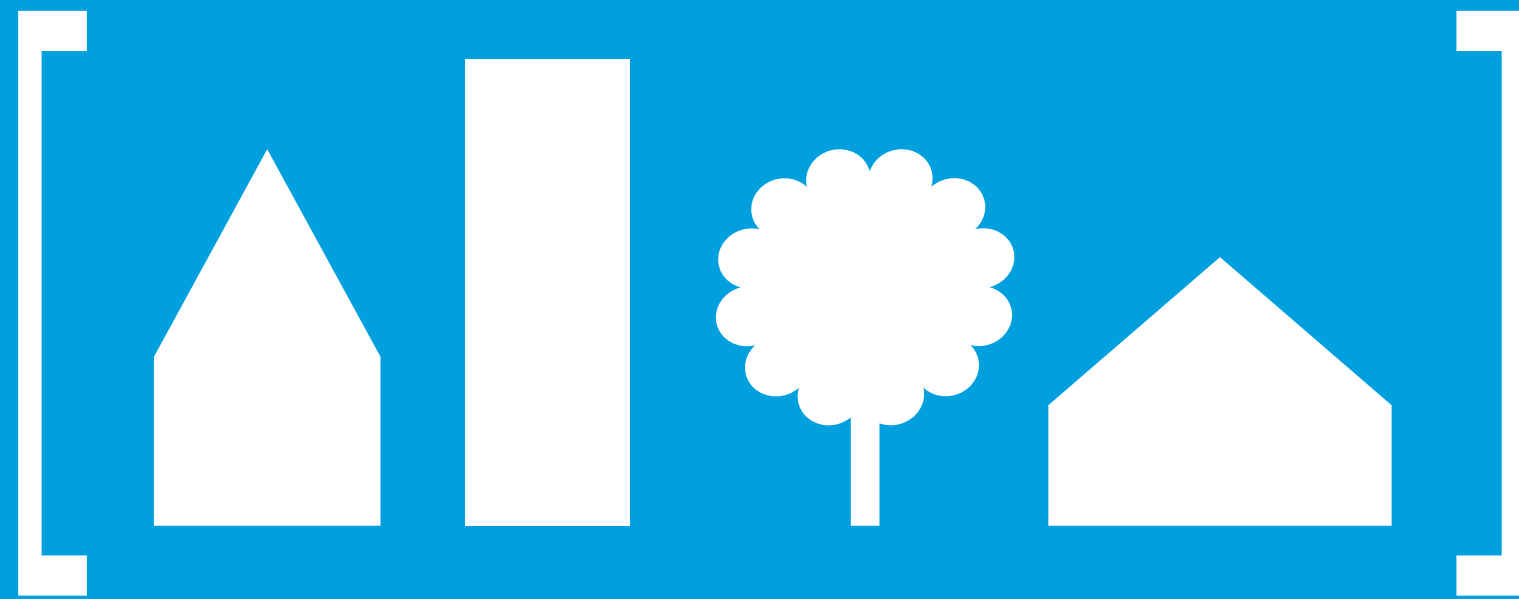




CityJSON

Hugo Ledoux

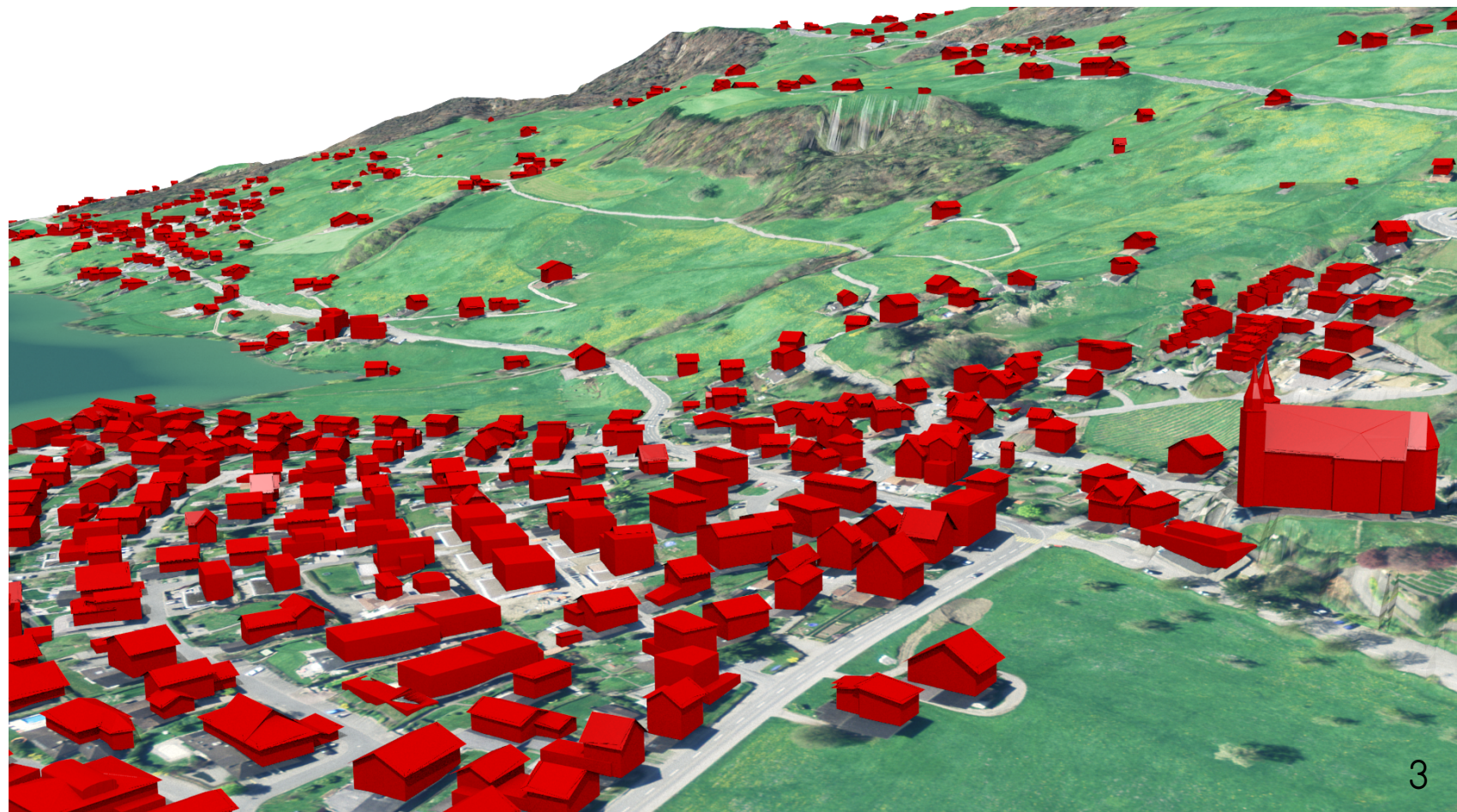
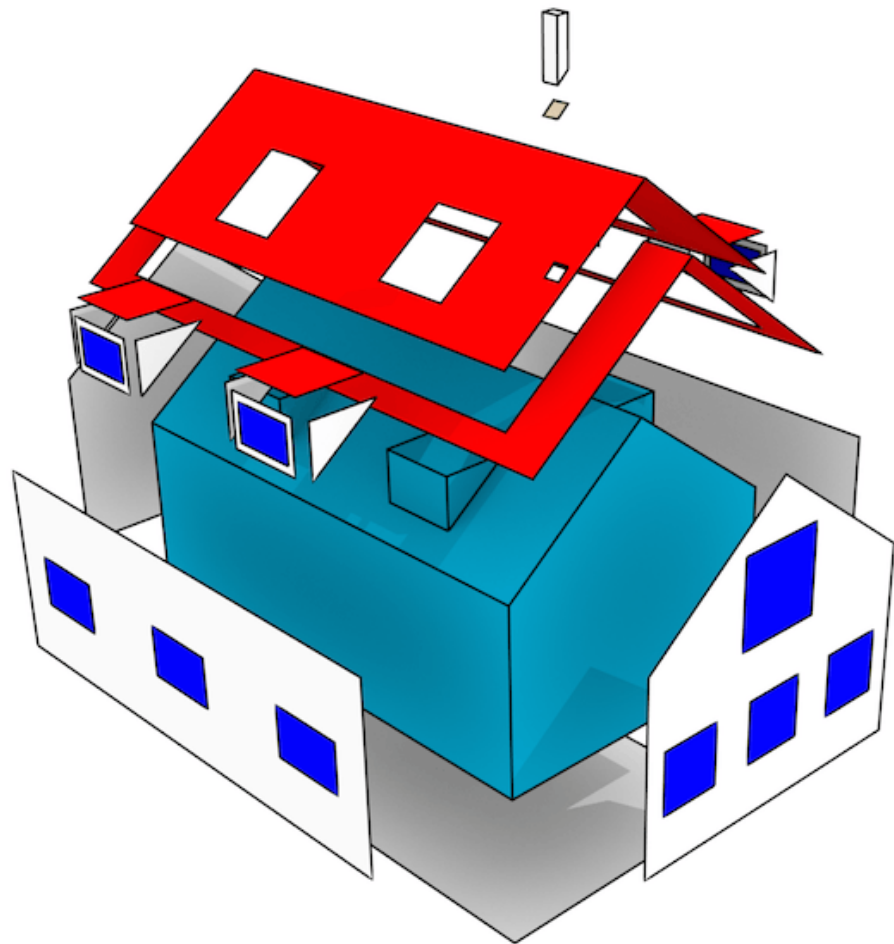
2018-07-04



CityGML

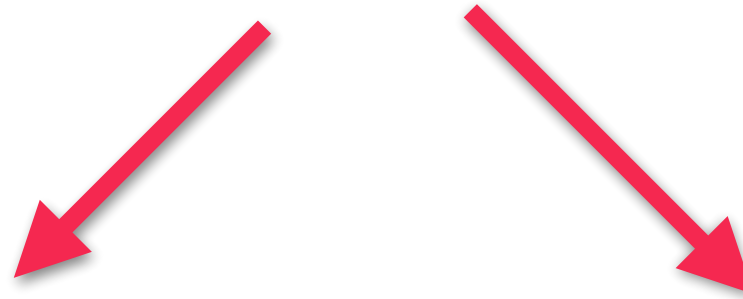


int'l standard (from OGC) for representing and storing 3D city models





CityGML



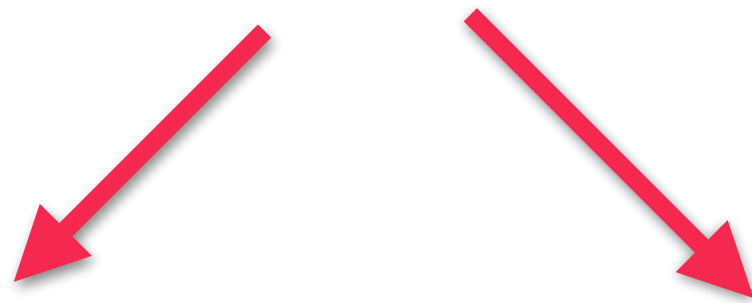
Data model

a GML encoding

both are called CityGML 🙄



CityGML



Data model

a GML encoding

**which technically doesn't exist...
but can be 'reverse engineered'
from the specs**

both are called CityGML 🙄

Two implementation of CityGML *data model*

1. CityGML (GML encoding, surprise surprise!)
2. 3DCityDB: PostgreSQL and Oracle Spatial



CityJSON can be seen as the 3rd implementation



CityJSON

JSON-based implementation of CityGML data model, easy-to-use and compact

[Watch](#) **23**

Navigation

- [CityJSON specifications](#)
- [Schema validation](#)
- [Software](#)
- [CityGML support](#)
- [Example datasets](#)
- [Changelog](#)
- [Contact](#)

- [CityGML homepage](#)
- [JSON homepage](#)
- [GitHub repository](#)

Quick search

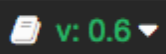
CityJSON

(version 0.6)

```
{
  "type": "CityJSON",
  "version": "0.6",
  "metadata": {
    "crs": { "epsg": 7415 }
  },
  "CityObjects": {
    "id-1": {
      "type": "Building",
      "attributes": { "roofType": "gable" },
      "geometry": [{
        "type": "Solid",
        "lod": 2,
        "boundaries": [...]
      }]
    }
  },
  "vertices": [
    [23.1, 2321.2, 11.0],
    ...
  ],
  "appearance": {...},
  "geometry-templates": {}
}
```

CityJSON is a format for encoding a subset of the [CityGML](#) data model using JavaScript Object Notation ([JSON](#)). A CityJSON file represents both the geometry and the semantics of the city features of a given area, eg buildings, roads, rivers, the vegetation, and the city furniture.

The aim of CityJSON is to offer an alternative to the GML encoding of CityGML, which can be verbose and complex (and thus rather frustrating to work with). CityJSON aims at being easy-to-use, both for reading datasets, and for creating them. It was designed with programmers in mind, so that tools and APIs supporting it can be quickly built. It



Why?

- CityGML files are complex and large
- No parsers in JavaScript
- Few software packages use CityGML
- Many diff ways to do one thing
- I personally get 😞 each time I get a new file

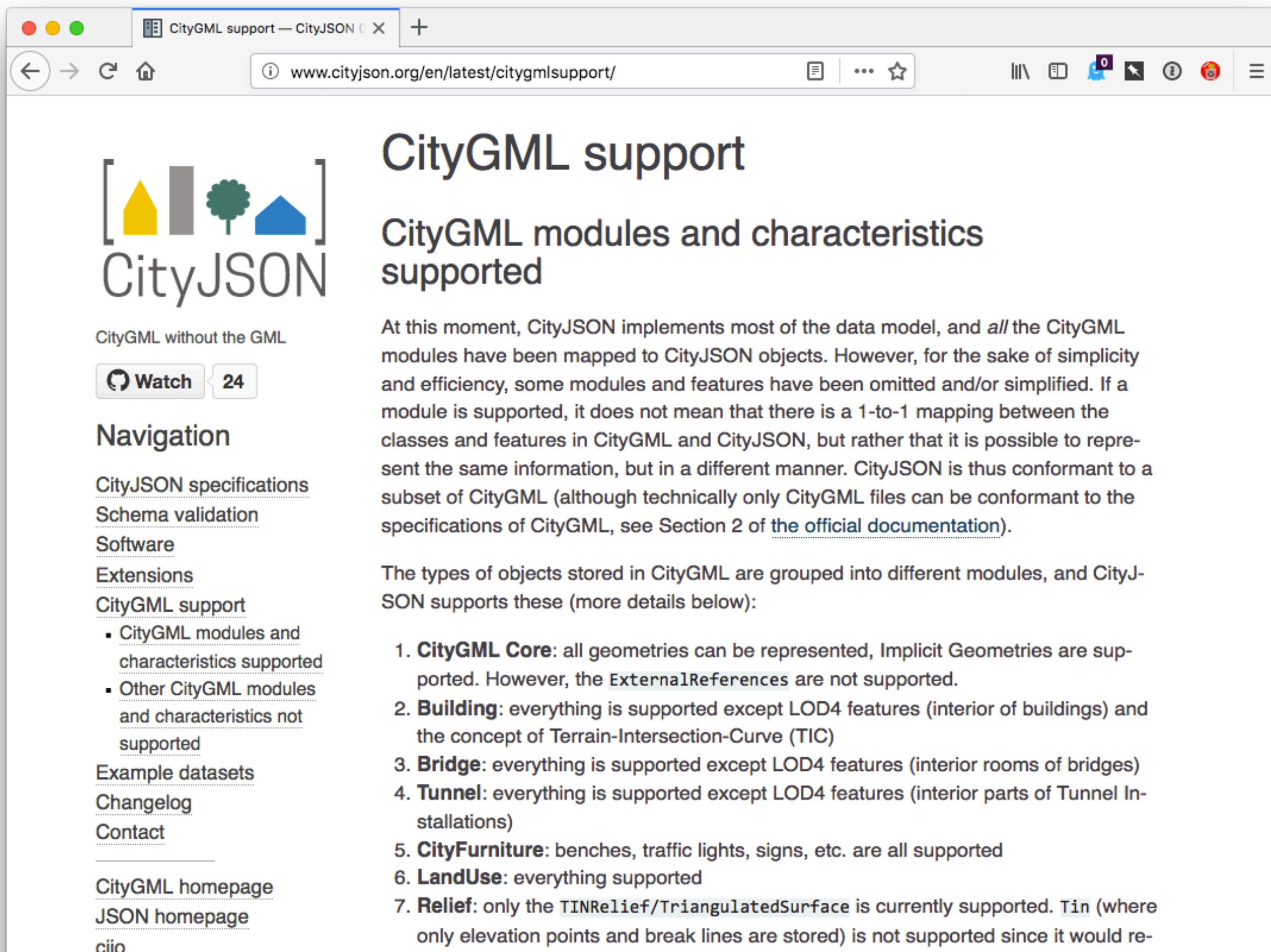


v0.6 released, v0.7 almost ready

- all modules mapped 
- software for full conversion
CityGML <-> CityJSON
- we tested it with C++, Python,
and Java

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"

“Is it CityGML compliant?” → yeah, for a subset



The screenshot shows a web browser window with the address bar displaying www.cityjson.org/en/latest/citygmalsupport/. The page title is "CityGML support". On the left, there is a sidebar with the CityJSON logo (a yellow house, a grey bar, a green tree, and a blue house) and the text "CityJSON". Below the logo, it says "CityGML without the GML" and "Watch 24". The sidebar also contains a "Navigation" section with links to "CityJSON specifications", "Schema validation", "Software", "Extensions", "CityGML support", "Example datasets", "Changelog", and "Contact". Under "CityGML support", there are two sub-links: "CityGML modules and characteristics supported" and "Other CityGML modules and characteristics not supported".

CityGML support

CityGML modules and characteristics supported

At this moment, CityJSON implements most of the data model, and *all* the CityGML modules have been mapped to CityJSON objects. However, for the sake of simplicity and efficiency, some modules and features have been omitted and/or simplified. If a module is supported, it does not mean that there is a 1-to-1 mapping between the classes and features in CityGML and CityJSON, but rather that it is possible to represent the same information, but in a different manner. CityJSON is thus conformant to a subset of CityGML (although technically only CityGML files can be conformant to the specifications of CityGML, see Section 2 of [the official documentation](#)).

The types of objects stored in CityGML are grouped into different modules, and CityJSON supports these (more details below):

1. **CityGML Core:** all geometries can be represented, Implicit Geometries are supported. However, the `ExternalReferences` are not supported.
2. **Building:** everything is supported except LOD4 features (interior of buildings) and the concept of Terrain-Intersection-Curve (TIC)
3. **Bridge:** everything is supported except LOD4 features (interior rooms of bridges)
4. **Tunnel:** everything is supported except LOD4 features (interior parts of Tunnel Installations)
5. **CityFurniture:** benches, traffic lights, signs, etc. are all supported
6. **LandUse:** everything supported
7. **Relief:** only the `TINRelief/TriangulatedSurface` is currently supported. `Tin` (where only elevation points and break lines are stored) is not supported since it would re-

“Is it CityGML compliant?” → yeah, for a subset



The screenshot shows a web browser window with the address bar displaying www.cityjson.org/en/latest/citygmlsupport/. The page title is "CityGML support". A large red overlay box with white text is centered on the page, stating: "to be clear: this is for CityGML v2.0" and "v3.0 is a draft and is way more complex". The background page content includes a navigation menu on the left with links like "CityGML without...", "Watch", "Navigation", "CityJSON spec", "Schema validation", "Software", "Extensions", "CityGML support", "Example datasets", "Changelog", and "Contact". The main content area lists supported CityGML modules: 1. CityGML modules and characteristics supported, 2. Building, 3. Bridge, 4. Tunnel, 5. CityFurniture, 6. LandUse, and 7. Relief. It also mentions that ExternalReferences are not supported.

CityGML support

to be clear: this is for CityGML v2.0

v3.0 is a draft and is way more complex

CityGML without...

Watch

Navigation

CityJSON spec

Schema validation

Software

Extensions

CityGML support

- CityGML modules and characteristics supported
- Other CityGML modules and characteristics not supported

Example datasets

Changelog

Contact

CityGML homepage

JSON homepage

CityGML support

1. **CityGML modules and characteristics supported:**

- 1. **CityGML modules and characteristics supported:**
- 2. **Building:** everything is supported except LOD4 features (interior of buildings) and the concept of Terrain-Intersection-Curve (TIC)
- 3. **Bridge:** everything is supported except LOD4 features (interior rooms of bridges)
- 4. **Tunnel:** everything is supported except LOD4 features (interior parts of Tunnel Installations)
- 5. **CityFurniture:** benches, traffic lights, signs, etc. are all supported
- 6. **LandUse:** everything supported
- 7. **Relief:** only the `TINRelief/TriangulatedSurface` is currently supported. `Tin` (where only elevation points and break lines are stored) is not supported since it would re-

ported. However, the `ExternalReferences` are not supported.

A CityJSON file

```
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {
    "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": []
  }
}
```

A CityJSON file

```
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {
    "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": []
  }
}
```

version CityJSON

A CityJSON file

```
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {
    "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": []
  }
}
```

metadata, ISO19115 “compliant”

CityGML has no mechanism in v2.0,
but I felt this is needed

ALL geometries have the same CRS,
unlike (City)GML

A CityJSON file

```
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {
    "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": []
  }
}
```

All City Objects listed here, *indexed* by their ID

Each have geometries + attributes

A CityJSON file

```
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {
    "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": []
      }
    }
  }
}
```



Geometry is ID of the vertex, global list

compression + more “topology”

A CityJSON file

```
{
  "type": "CityJSON",
  "version": "0.7",
  "metadata": {
    "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": []
  }
}
```

material + texture possible

Semantics surfaces, with attributes

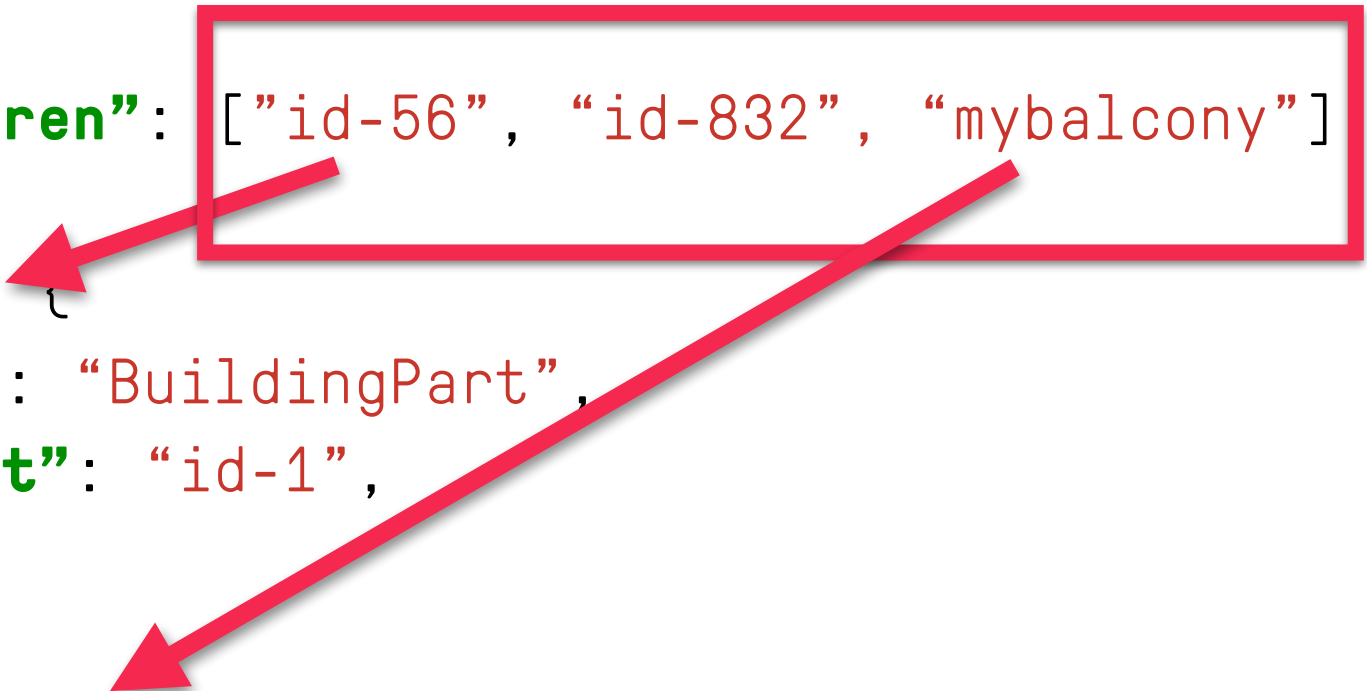
```
[
  "geometry": [
    {
      "type": "Solid",
      "lod": 2.2,
      "boundaries": [
        [ [0, 3, 2, 1]], [[4, 5, 6, 7]], [[0, 1, 5, 4]], [[1, 2, 6, 5]], [[2, 3, 7, 6]], [[3, 0, 4, 7]] ]
      ],
      "semantics": {
        "surfaces" : [
          {
            "type": "RoofSurface",
            "slope": 33.4
          },
          {
            "type": "RoofSurface",
            "slope": 66.6
          },
          {
            "type": "WallSurface",
            "paint": "blue"
          }
        ],
        "values": [
          [0, 1, null, 2, 2, 2]
        ]
      }
    }
  ]
]
```


BuildingParts: links between City Objects

```
"CityObjects": {  
  "id-1": {  
    "type": "Building",  
    "attributes": {  
      "roofType": "gable"  
    },  
    "children": ["id-56", "id-832", "mybalcony"]  
  },  
  "id-56": {  
    "type": "BuildingPart",  
    "parent": "id-1",  
    ...  
  },  
  "mybalcony": {  
    "type": "BuildingInstallation",  
    "parent": "id-1",  
    ...  
  }  
}
```

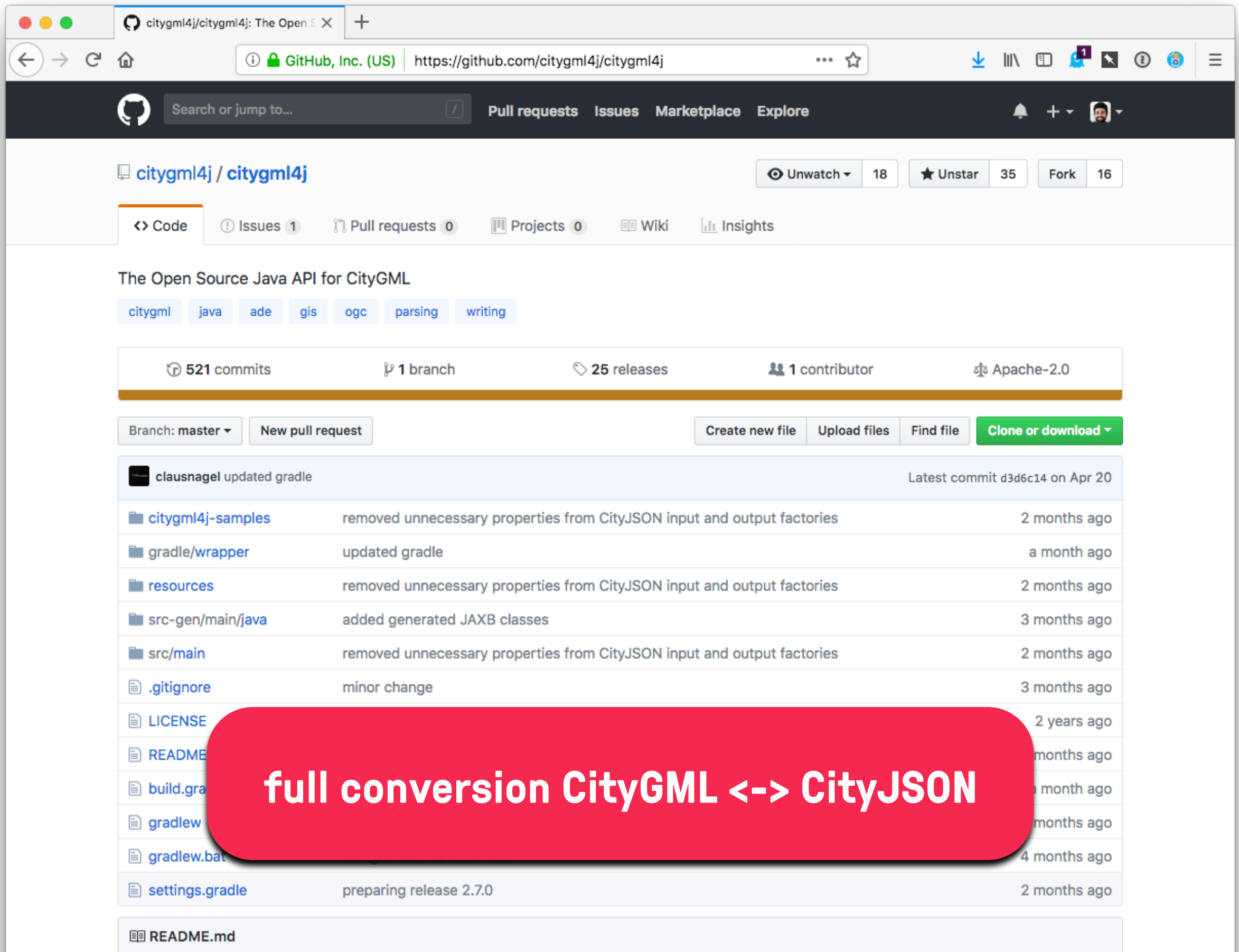
BuildingParts: links between City Objects

```
"CityObjects": {  
  "id-1": {  
    "type": "Building",  
    "attributes": {  
      "roofType": "gable"  
    },  
    "children": ["id-56", "id-832", "mybalcony"]  
  },  
  "id-56": {  
    "type": "BuildingPart",  
    "parent": "id-1",  
    ...  
  },  
  "mybalcony": {  
    "type": "BuildingInstallation",  
    "parent": "id-1",  
    ...  
  }  
}
```



goal == a flat schema

software



citygml4j / citygml4j

Unwatch 18 Unstar 35 Fork 16

Code Issues 1 Pull requests 0 Projects 0 Wiki Insights

The Open Source Java API for CityGML

citygml java ade gis ogc parsing writing

521 commits 1 branch 25 releases 1 contributor Apache-2.0

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

clausnagel updated gradle Latest commit d3d6c14 on Apr 20

citygml4j-samples	removed unnecessary properties from CityJSON input and output factories	2 months ago
gradle/wrapper	updated gradle	a month ago
resources	removed unnecessary properties from CityJSON input and output factories	2 months ago
src-gen/main/java	added generated JAXB classes	3 months ago
src/main	removed unnecessary properties from CityJSON input and output factories	2 months ago
.gitignore	minor change	3 months ago
LICENSE		2 years ago
README		months ago
build.gradle		a month ago
gradlew		months ago
gradlew.bat		4 months ago
settings.gradle	preparing release 2.7.0	2 months ago

README.md

full conversion CityGML <-> CityJSON



Azul 4+

azul is a 3D viewer for 3D city models in (City)GML, CityJSON, OBJ, OFF and POLY. It supports loading multiple files, selecting objects by clicking them or selecting them in the sidebar, and browsing their attributes.

What's New in Version 0.8.1
Support for CityJSON 0.5

Install

Featured

Top Charts

Categories

Purchased

Updates

Azul Support

Information

Category: Utilities

Updated: 19 December 2017

Version: 0.8.1

Price: Free

Size: 9.6 MB

Family Sharing: Yes

Language: English

Developer: Ken Arroyo Otori

© 2016-2017 Ken Arroyo Otori

Rated 4+

Compatibility:
macOS 10.13 or later, 64-bit
processor

Stadt-Ettenheim-LoD3_edited_v1.0.0.gml



cjio (CityJSON/io)

2. bash

```
Hugos-MacBook-Pro:rotterdam hugo$ cjio
```

```
Usage: cjio [OPTIONS] INPUT COMMAND1 [ARGS]... [COMMAND2 [ARGS]...]...
```

Process and manipulate a CityJSON file, and allow different outputs. The different operators can be chained to perform several processing in one step, the CityJSON model goes through the different operators.

To get help on specific command, eg for 'validate':

```
  cjio validate --help
```

Usage examples:

```
  cjio example.json validate
```

```
  cjio example.json remove_textures info
```

```
  cjio example.json subset --id house12 remove_materials save out.json
```

Options:

<code>--version</code>	Show the version and exit.
<code>--off</code>	Load an OFF file and convert it to one CityJSON GenericCityObject.
<code>--ignore_duplicate_keys</code>	Load a CityJSON file even if some City Objects have the same IDs (technically invalid file)
<code>--help</code>	Show this message and exit.

Commands:

<code>compress</code>	Compress a CityJSON file, ie stores its...
<code>decompress</code>	Decompress a CityJSON file, ie remove the...
<code>info</code>	Output info in simple JSON.
<code>merge</code>	Merge the current CityJSON with others.
<code>remove_duplicate_vertices</code>	Remove duplicate vertices a CityJSON file.
<code>remove_materials</code>	Remove all materials from a CityJSON file.
<code>remove_orphan_vertices</code>	Remove orphan vertices a CityJSON file.
<code>remove_textures</code>	Remove all textures from a CityJSON file.
<code>save</code>	Save the CityJSON to a file.
<code>subset</code>	Create a subset of a CityJSON file.
<code>update_bbox</code>	Update the bbox of a CityJSON file.
<code>update_crs</code>	Update the CRS with a new value.
<code>validate</code>	Validate the CityJSON file: (1) against its...

pip install cjio

cjio (CityJSON/io)

2. bash

```
Hugos-MacBook-Pro:rotterdam hugo$ cjio
```

```
Usage: cjio [OPTIONS] INPUT COMMAND1 [ARGS]... [COMMAND2 [ARGS]...]...
```

Process and manipulate a CityJSON file, and allow different outputs. The different operators can be chained to perform several processing in one step, the CityJSON model goes through the different operators.

To get help on specific command, eg for 'validate':

```
  cjio validate --help
```

Usage examples:

```
  cjio example.json validate
```

```
  cjio example.json remove_textures info
```

```
  cjio example.json subset --id house12 remove_materials save out.json
```

Options:

<code>--version</code>	Show the version and exit.
<code>--off</code>	Load an OFF file and convert it to one CityJSON GenericCityObject.
<code>--ignore_duplicate_keys</code>	Load a CityJSON file even if some City Objects have the same IDs (technically invalid file)
<code>--help</code>	Show this message and exit.

Commands:

<code>compress</code>	Compress a CityJSON file, ie stores its...
<code>decompress</code>	Decompress a CityJSON file, ie remove the...
<code>info</code>	Output info in simple JSON.
<code>merge</code>	Merge the current CityJSON with others.
<code>remove_duplicate_vertices</code>	Remove duplicate vertices a CityJSON file.
<code>remove_materials</code>	Remove all materials from a CityJSON file.
<code>remove_orphan_vertices</code>	Remove orphan vertices a CityJSON file.
<code>remove_textures</code>	Remove all textures from a CityJSON file.
<code>save</code>	Save the CityJSON to a file.
<code>subset</code>	Create a subset of a CityJSON file.
<code>update_bbox</code>	Update the bbox of a CityJSON file.
<code>update_crs</code>	Update the CRS with a new value.
<code>validate</code>	Validate the CityJSON file: (1) against its...

Compression factor == ~7X

file	CityGML size (original)	CityGML size (w/o spaces)	textures?	CityJSON	CityJSON compressed	compression factor
CityGML demo "GeoRes"	4.3MB	4.1MB	yes	582KB	524KB	8.0
CityGML v2 demo "Railway"	45MB	34MB	yes	4.5MB	4.3MB	8.1
Den Haag "tile 01"	23MB	18MB	no, material	3.1MB	2.9MB	6.2
Montréal VM05	56MB	42MB	yes	5.7MB	5.4MB	7.8
New York LoD2 (DA13)	590MB	574MB	no	110MB	105MB	5.5
Rotterdam Delfshaven	16MB	15MB	yes	2.8MB	2.6MB	5.4
Vienna	37MB	36MB	no	5.6MB	5.3MB	6.8

A snippet of a given CityGML file

```
<cityObjectMember>
  <veg:SolitaryVegetationObject gml:id="GMLID_S00200811_525_4986">
    <gml:name>Forest tree 1</gml:name>
    <relativeToTerrain>entirelyAboveTerrain</relativeToTerrain>
    <veg:class>1060</veg:class>
    <veg:species>1640</veg:species>
    <veg:lod3ImplicitRepresentation>
      <ImplicitGeometry>
        <transformationMatrix>1.0 0.0 0.0 1.11999777602438 0.0 1.0 0.0 6.88000452683357 0.0 0.0 1.0 8.8500288642283 0.0 0.0 0.0 1.0</transformationMatrix>
        <relativeGMLGeometry>
          <gml:MultiSurface gml:id="Mult_S0045475_3243_790864">
            <gml:surfaceMember>
              <gml:Polygon gml:id="PolyID72_147_863766_278961">
                <gml:exterior>
                  <gml:LinearRing gml:id="PolyID72_147_863766_278961_0">
                    <gml:posList>0.111775082264331 0.12072112251111 0.502222057066866 0.107420882383068 0.120661653965679 0.502222101238315 0.103420882385374 0.118940482732408 0.50222205960968 0.100384031890171 0.115819514600485
0.502222126199463 0.0987726533874718 0.111774040103382 0.502222074955247 0.0988320939989807 0.107419851779835 0.502222054525534 0.100553293837055 0.103419839857094 0.502222036213702 0.103674216513289 0.100382948053549
0.502222094509167 0.107719709663888 0.0987716382157979 0.502221997505005 0.112073909545149 0.0988310413593261 0.502222052999943 0.116073924443975 0.100552277994501 0.502221994962193 0.119110774939178 0.10367318072452 0.502222028038794
0.120722138540746 0.107718655221622 0.502222079283008 0.120662697929237 0.112072843545171 0.502222099712726 0.118941498091162 0.116072855467913 0.502222118024554 0.115820590316061 0.119109747271456 0.502222059729089 0.111775082264331
0.12072112251111 0.502222057066866</gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
          <gml:surfaceMember>
            <gml:Polygon gml:id="PolyID73_911_469224_286874">
              <gml:exterior>
                <gml:LinearRing gml:id="PolyID73_911_469224_286874_0">
                  <gml:posList>0.0161136471915356 0.170479293291229 0.130205976087941 -4.51194637207664E-17 0.130024270847741 0.130205851900683 0.0987726533874718 0.111774040103382 0.502222074955247 0.100384031890171
0.115819514600485 0.502222126199463 0.0161136471915356 0.170479293291229 0.130205976087941</gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon gml:id="PolyID74_838_123593_188440">
            <gml:exterior>
              <gml:LinearRing gml:id="PolyID74_838_123593_188440_0">
                <gml:posList>-4.51194637207664E-17 0.130024270847741 0.130205851900683 5.94394939233379E-4 0.086482219725548 0.13020582260217 0.0988320939989807 0.107419851779835 0.502222054525534 0.0987726533874718
0.111774040103382 0.502222074955247 -4.51194637207664E-17 0.130024270847741 0.130205851900683</gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </ImplicitGeometry>
    </veg:lod3ImplicitRepresentation>
  </veg:SolitaryVegetationObject>
</cityObjectMember>
```

A snippet of a given CityGML file

```
<cityObjectMember><veg:SolitaryVegetationObject gml:id="GMLID_S00200811_525_4986"><gml:name>Forest tree 1</  
gml:name><relativeToTerrain>entirelyAboveTerrain</relativeToTerrain><veg:class>1060</veg:class><veg:species>1640</  
veg:species><veg:lod3ImplicitRepresentation><ImplicitGeometry><transformationMatrix>1.0 0.0 0.0 1.11999777602438 0.0 1.0 0.0 6.88000452683357  
0.0 0.0 1.0 8.8500288642283 0.0 0.0 0.0 1.0</transformationMatrix><relativeGMLGeometry><gml:MultiSurface  
gml:id="Mult_S0045475_3243_790864"><gml:surfaceMember><gml:Polygon gml:id="PolyID72_147_863766_278961"><gml:exterior><gml:LinearRing  
gml:id="PolyID72_147_863766_278961_0"><gml:posList>0.111775082264331 0.12072112251111 0.502222057066866 0.107420882383068 0.120661653965679  
0.502222101238315 0.103420882385374 0.118940482732408 0.50222205960968 0.100384031890171 0.115819514600485 0.502222126199463 0.0987726533874718  
0.111774040103382 0.502222074955247 0.0988320939989807 0.107419851779835 0.502222054525534 0.100553293837055 0.103419839857094  
0.502222036213702 0.103674216513289 0.100382948053549 0.502222094509167 0.107719709663888 0.0987716382157979 0.502221997505005  
0.112073909545149 0.0988310413593261 0.502222052999943 0.116073924443975 0.100552277994501 0.502221994962193 0.119110774939178 0.10367318072452  
0.502222028038794 0.120722138540746 0.107718655221622 0.502222079283008 0.120662697929237 0.112072843545171 0.502222099712726 0.118941498091162  
0.116072855467913 0.502222118024554 0.115820590316061 0.119109747271456 0.502222059729089 0.111775082264331 0.12072112251111  
0.502222057066866</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></gml:surfaceMember><gml:surfaceMember><gml:Polygon  
gml:id="PolyID73_911_469224_286874"><gml:exterior><gml:LinearRing gml:id="PolyID73_911_469224_286874_0"><gml:posList>0.0161136471915356  
0.170479293291229 0.130205976087941 -4.51194637207664E-17 0.130024270847741 0.130205851900683 0.0987726533874718 0.111774040103382  
0.502222074955247 0.100384031890171 0.115819514600485 0.502222126199463 0.0161136471915356 0.170479293291229 0.130205976087941</gml:posList></  
gml:LinearRing></gml:exterior></gml:Polygon></gml:surfaceMember><gml:surfaceMember><gml:Polygon  
gml:id="PolyID74_838_123593_188440"><gml:exterior><gml:LinearRing gml:id="PolyID74_838_123593_188440_0"><gml:posList>-4.51194637207664E-17  
0.130024270847741 0.130205851900683 5.94394939233379E-4 0.086482219725548 0.13020582260217 0.0988320939989807 0.107419851779835  
0.502222054525534 0.0987726533874718 0.111774040103382 0.502222074955247 -4.51194637207664E-17 0.130024270847741 0.130205851900683</  
gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></gml:surfaceMember>
```

I removed all the spaces and CR, which
saved 25% in some cases...

Compression factor == ~7X

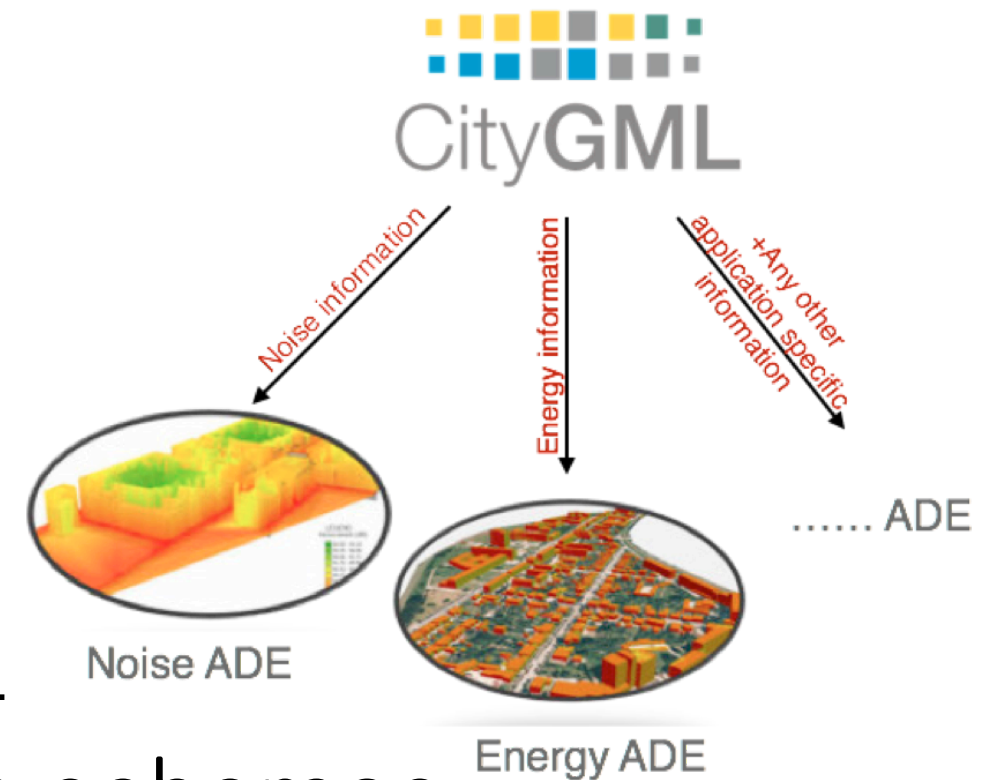
file	CityGML size (original)	CityGML size (w/o spaces)	textures?	CityJSON	CityJSON compressed	compression factor
CityGML demo "GeoRes"	4.3MB	4.1MB	yes	582KB	524KB	8.0
CityGML v2 demo "Railway"	45MB	34MB	yes	4.5MB	4.3MB	8.1
Den Haag "tile 01"	23MB	18MB	no, material	3.1MB	2.9MB	6.2
Montréal VM05	56MB	42MB	yes	5.7MB	5.4MB	7.8
New York LoD2 (DA13)	590MB	574MB	no	110MB	105MB	5.5
Rotterdam Delfshaven	16MB	15MB	yes	2.8MB	2.6MB	5.4
Vienna	37MB					6.8

👉 **CityGML v3 will be way more complex, and CityGML file size will increase**

Extensions

(aka as ADEs)

CityGML Application Domain Extensions



- <https://www.citygml.org/ade/>
- many designed in UML and with schemas
- but almost no software support...
- no viewer, no parser—> everything needs to be tailored-made for each

CityJSON Extensions

- simpler: strict rules for extensions
- less flexible than ADEs though
- in beta: <http://www.cityjson.org/en/latest/extensions/>
- one just defines a new JSON schema and that's it
- all downstream software should work (touch wood...)

What issues are left to tackle?

1. large files → tiling/splitting + master file
2. making it more “web-aware”: WFS?
3. developing software, viewers most urgent

www.cityjson.org

(all dev/discussions are open on GitHub)