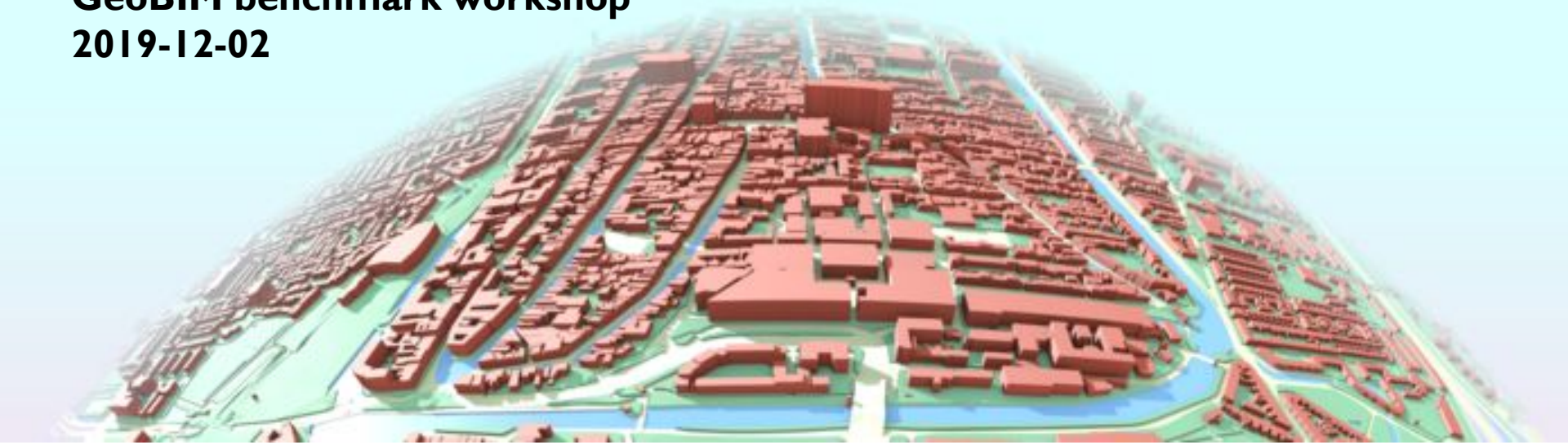
3D City Models, an overview

Tom Commandeur, Balázs Dukai

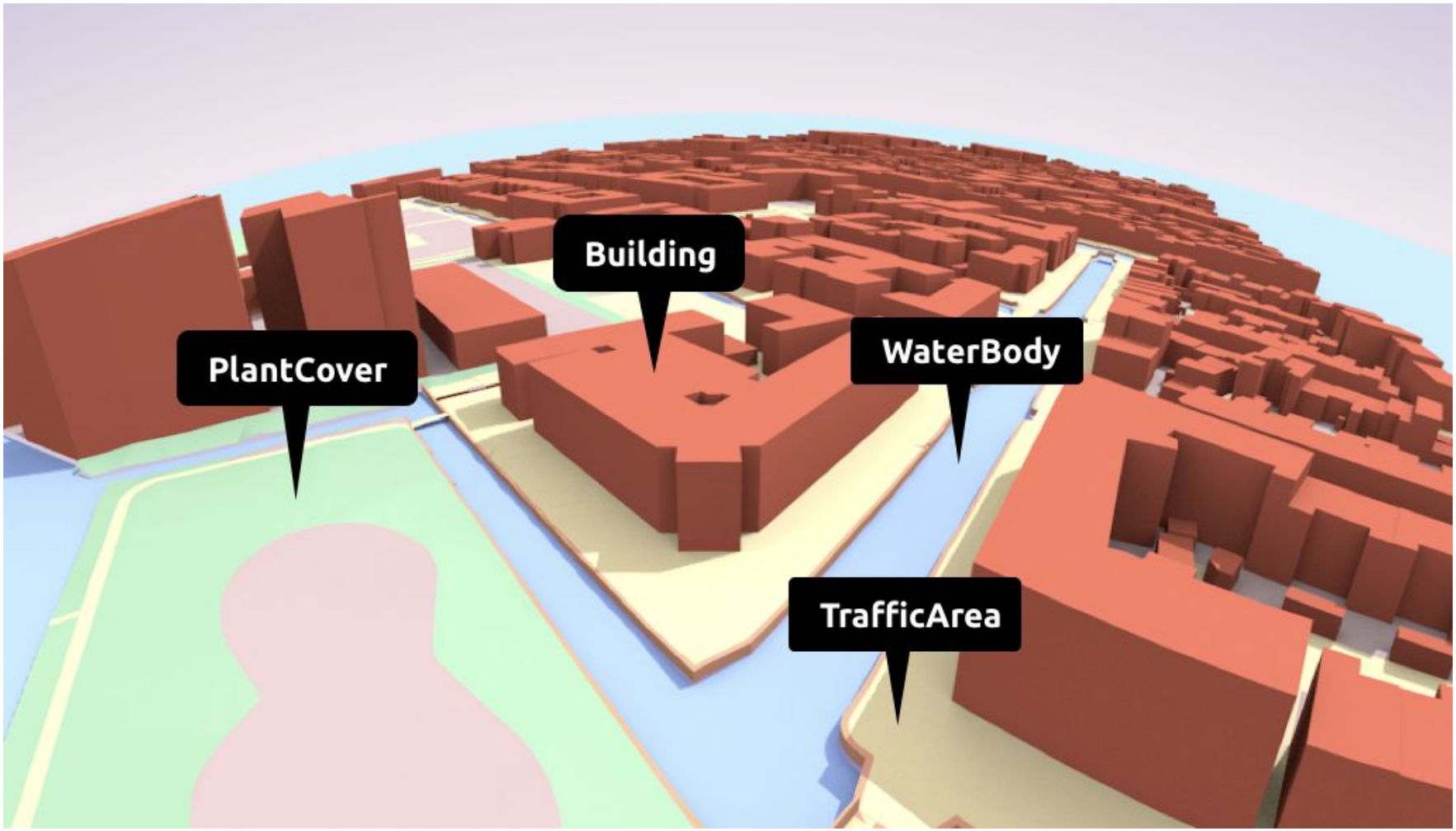
GeoBIM benchmark workshop
2019-12-02



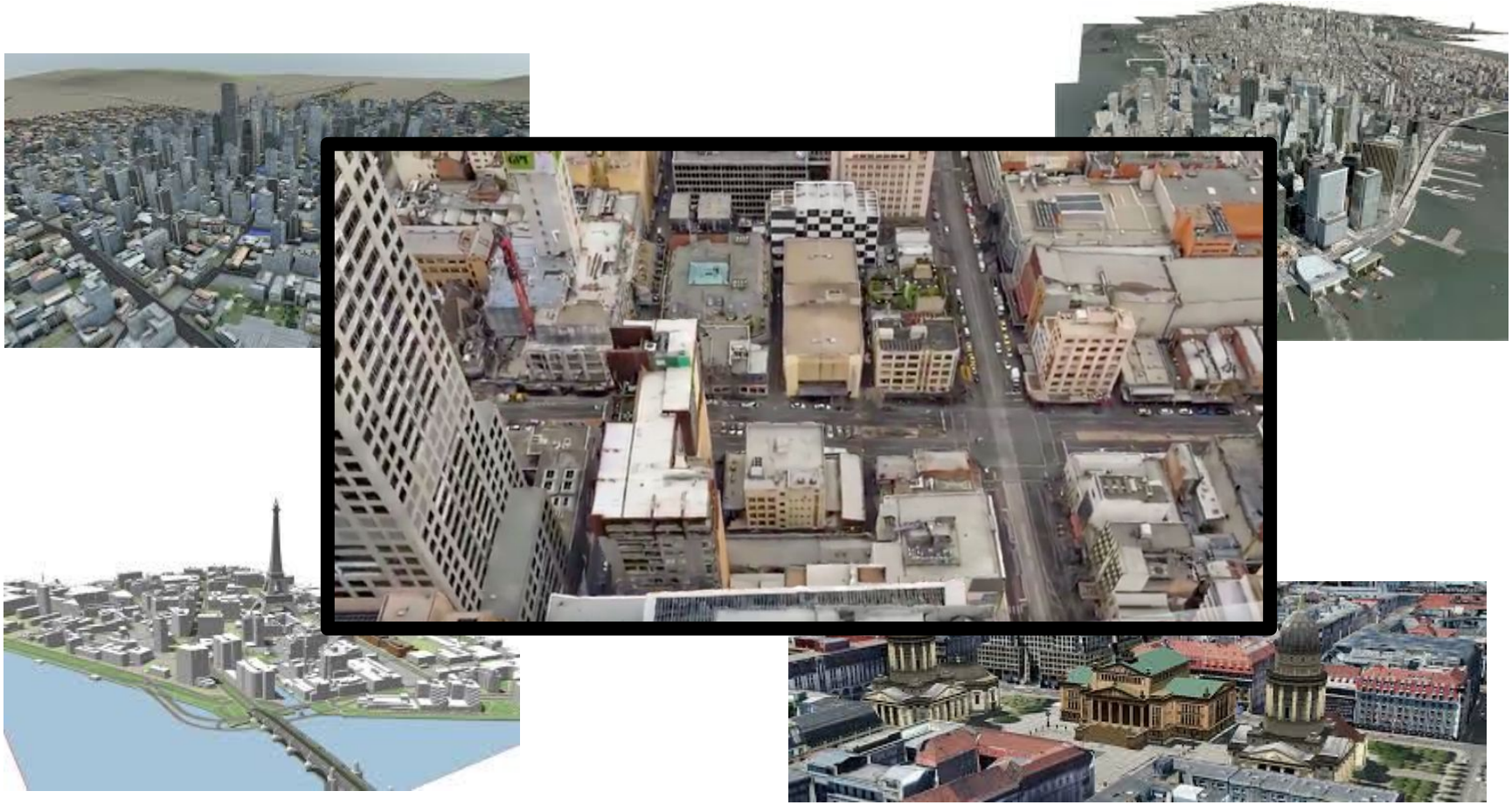
Mesh models


Source: earth.google.com
Google
2D
Camera 142 m 52°22'28"N 4°54'12"E 0 m 100%

Semantic models

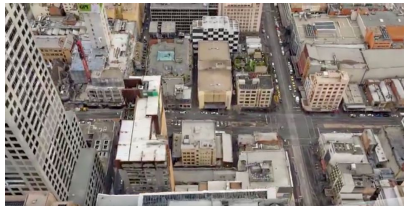


Relatively easy to reconstruct 3D city models



“The more realistic it looks, the better”, but……

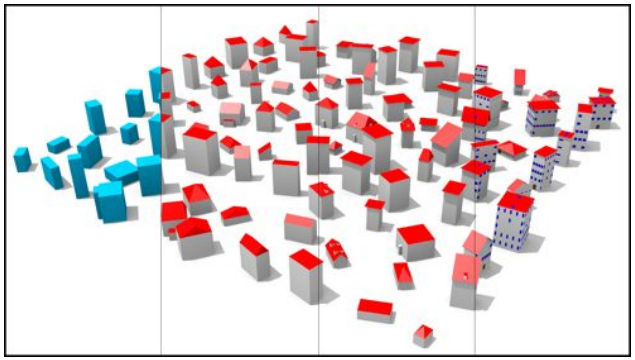


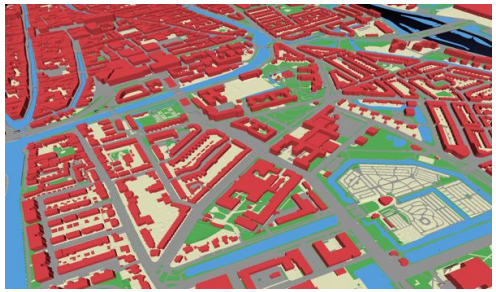
Value of 3D CM is in the data that can serve applications (and that can be updated)

3D city models differ a lot (even for same areas), due to difference in

- acquisition methods
- applications
- formats, data structures, etc

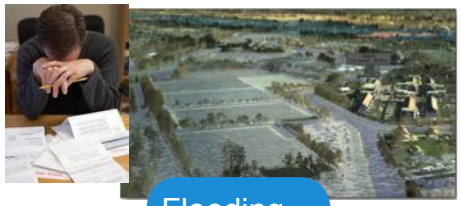


Problems of current 3D city models



- Existing 3D city models are non-consistent
- Once collected 3D data for an application can hardly be reused in another
- 3D city models often require (interactive) processing to use the data:
  - repairing the data, adding attributes, simplifying

Domain experts spend 70% of their time on 3D data processing

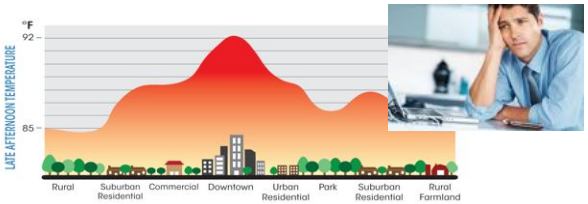


Flooding

Energy

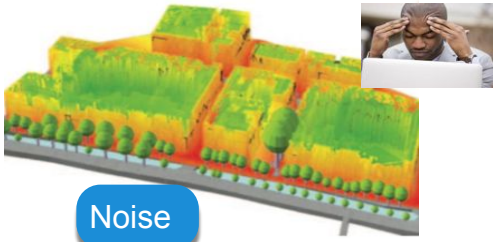
Where do I find useful data

Shadow



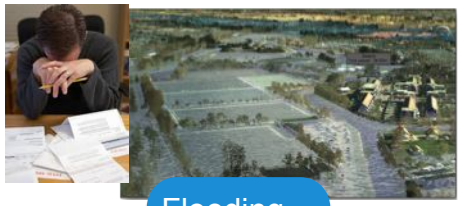
urban heat island

Solar potential



Noise

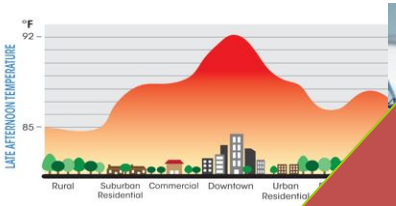
3D city models for Smart Cities



Flooding

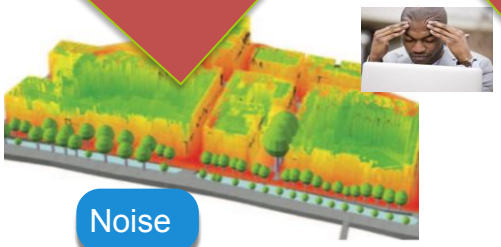
Energy

Shadow



urban heat island

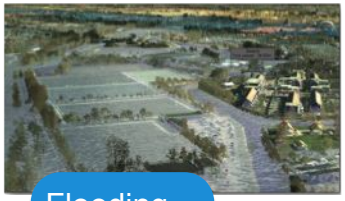
Solar potential



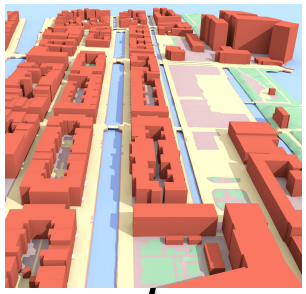
Noise

3D city models for Smart Cities

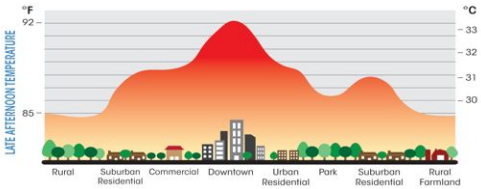
One authoritative 3D basis, in an international standard


Flooding


Energy

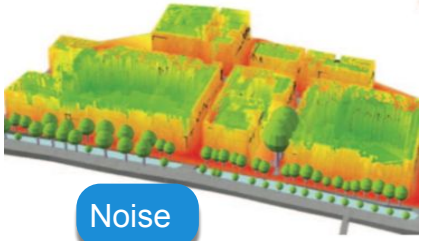



Shadow

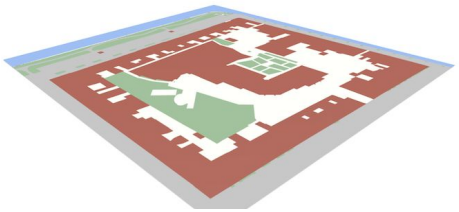


urban heat island

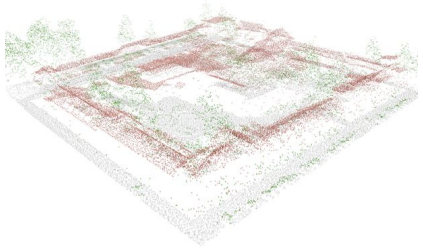

Solar potential


Noise

Open software to reconstruct 3D models

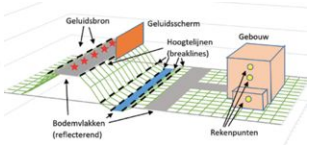


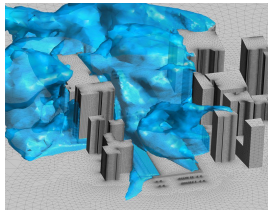
2D topography



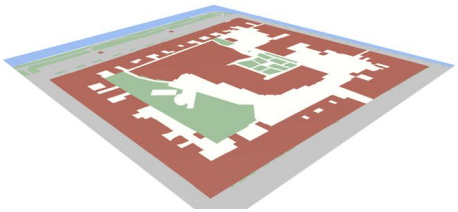
Point clouds, from
LiDAR and dense matching

3dfier

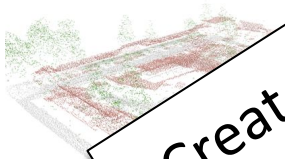




Open software to reconstruct 3D models

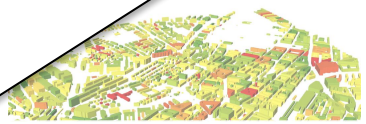


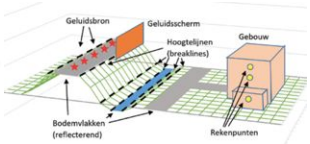
2D topography

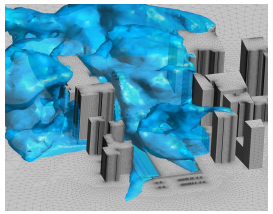


Point clou
LiDAR and atching

Creates up-to-date, valid and application specific 3D data *(for whole NL)* output: OBJ, CityGML, CityJSON etc





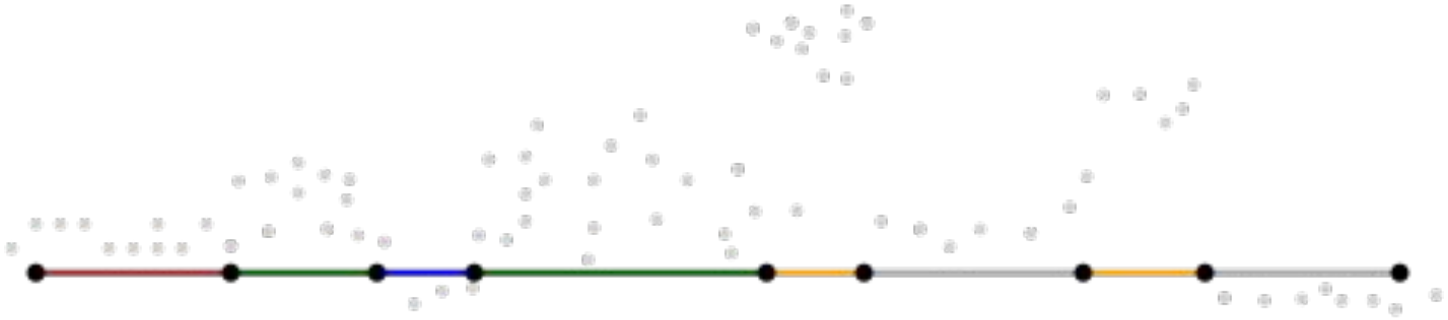


Input: any 2D datasets

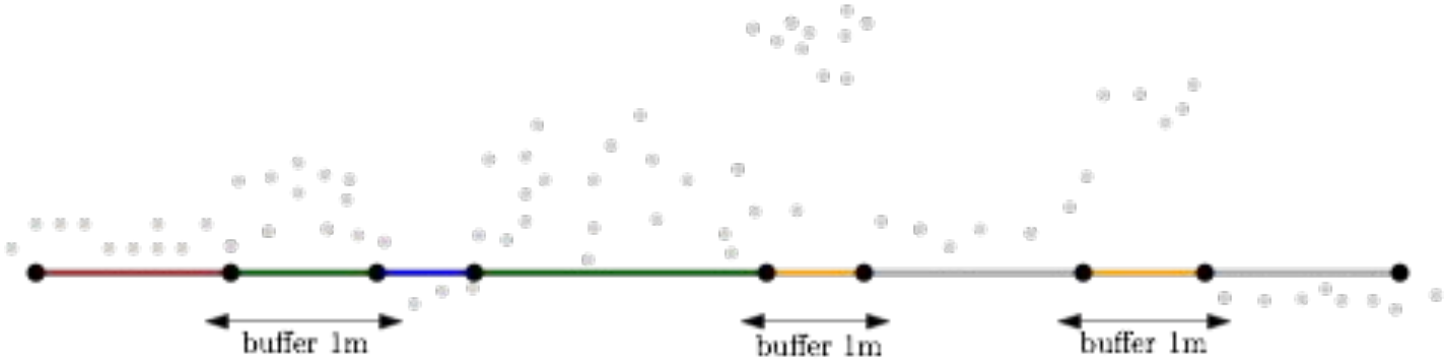
We assign each polygon to a class:

1. Building
2. Water
3. Road
4. Vegetation
5. Terrain
6. Separation
7. Bridge

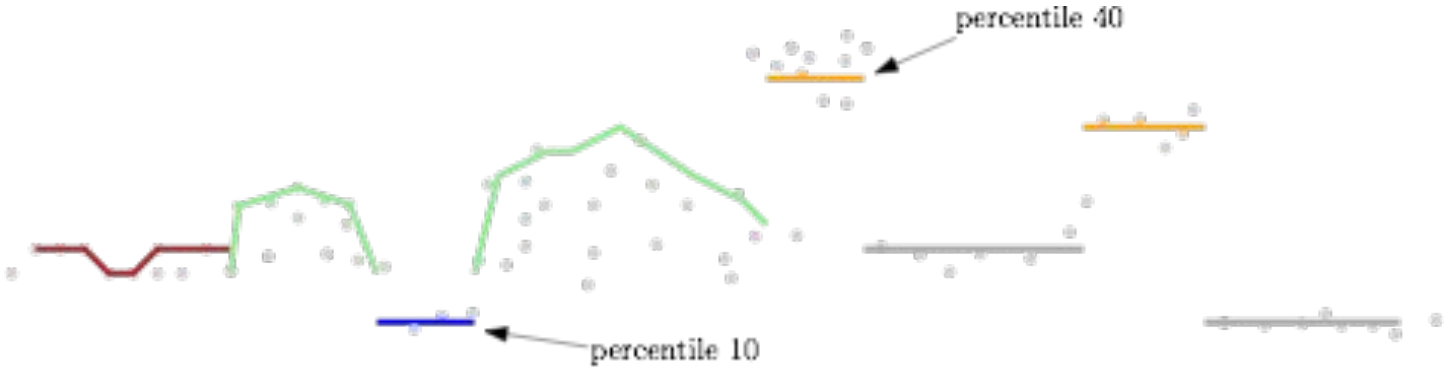
Assign each LiDAR point to polygons



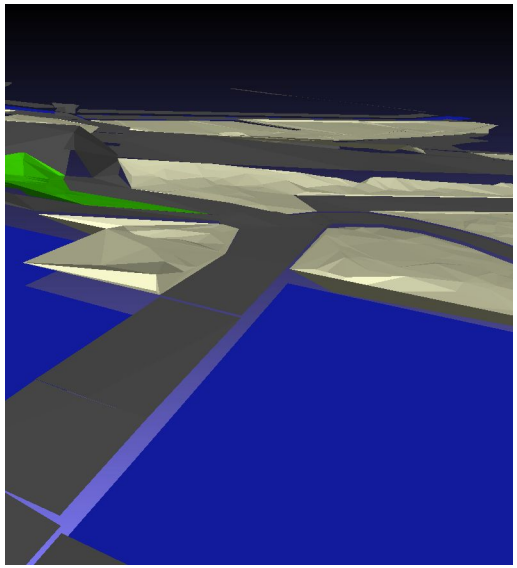
Assign each LiDAR point to polygons



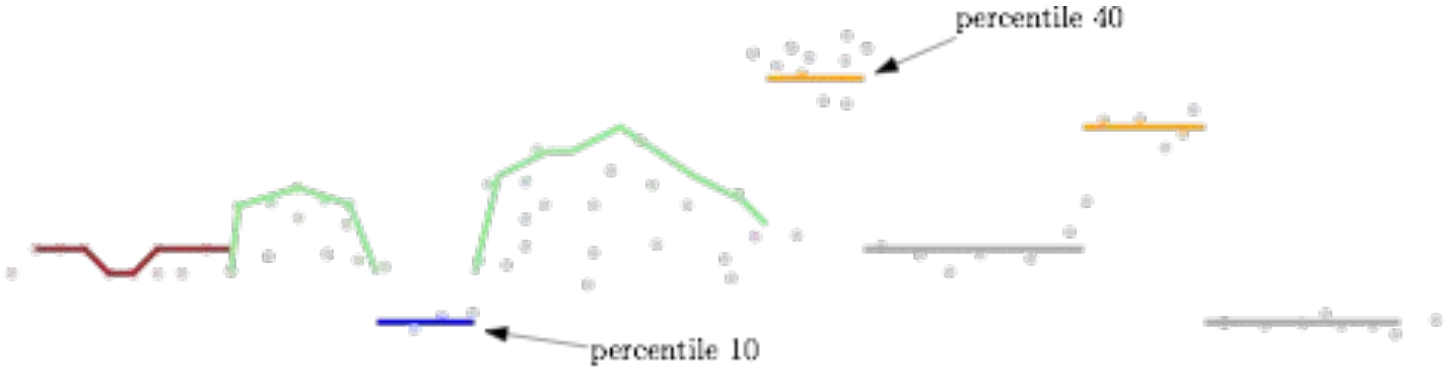
Lifting each polygon based on simple rules (eg avg)



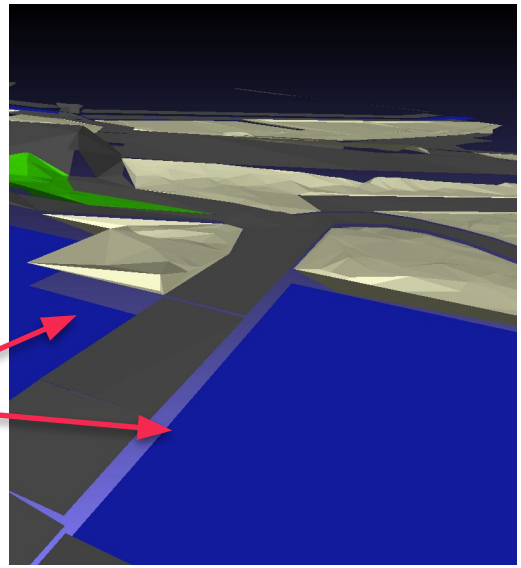
Terrain and vegetation classes have LiDAR points added within the boundary
Other classes only vertices are lifted



Lifting each polygon based on simple rules (eg avg)

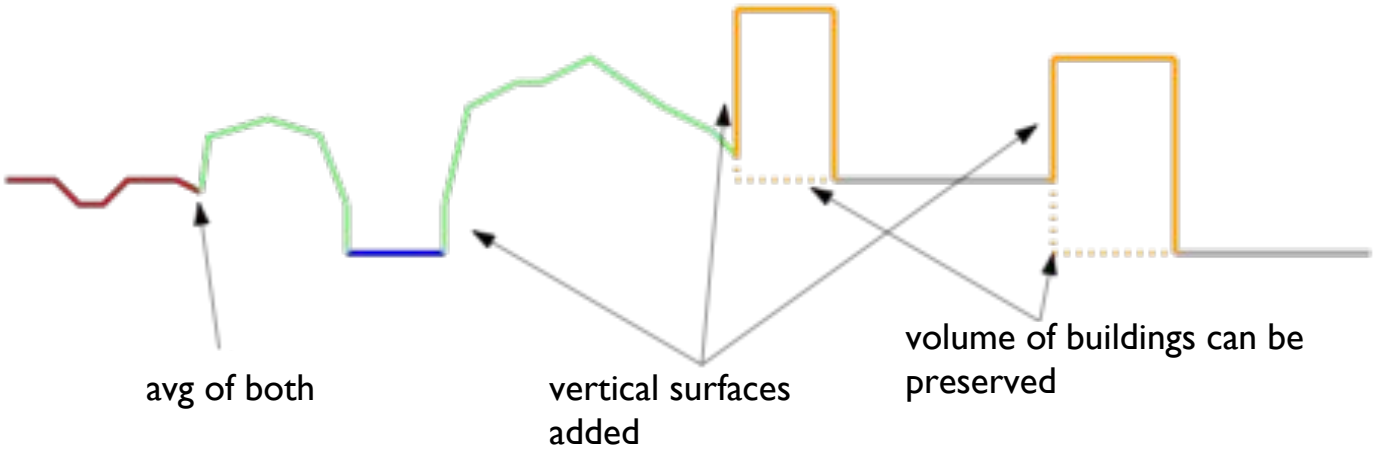


Terrain and vegetation classes have LiDAR points added within the boundary
Other classes only vertices are lifted

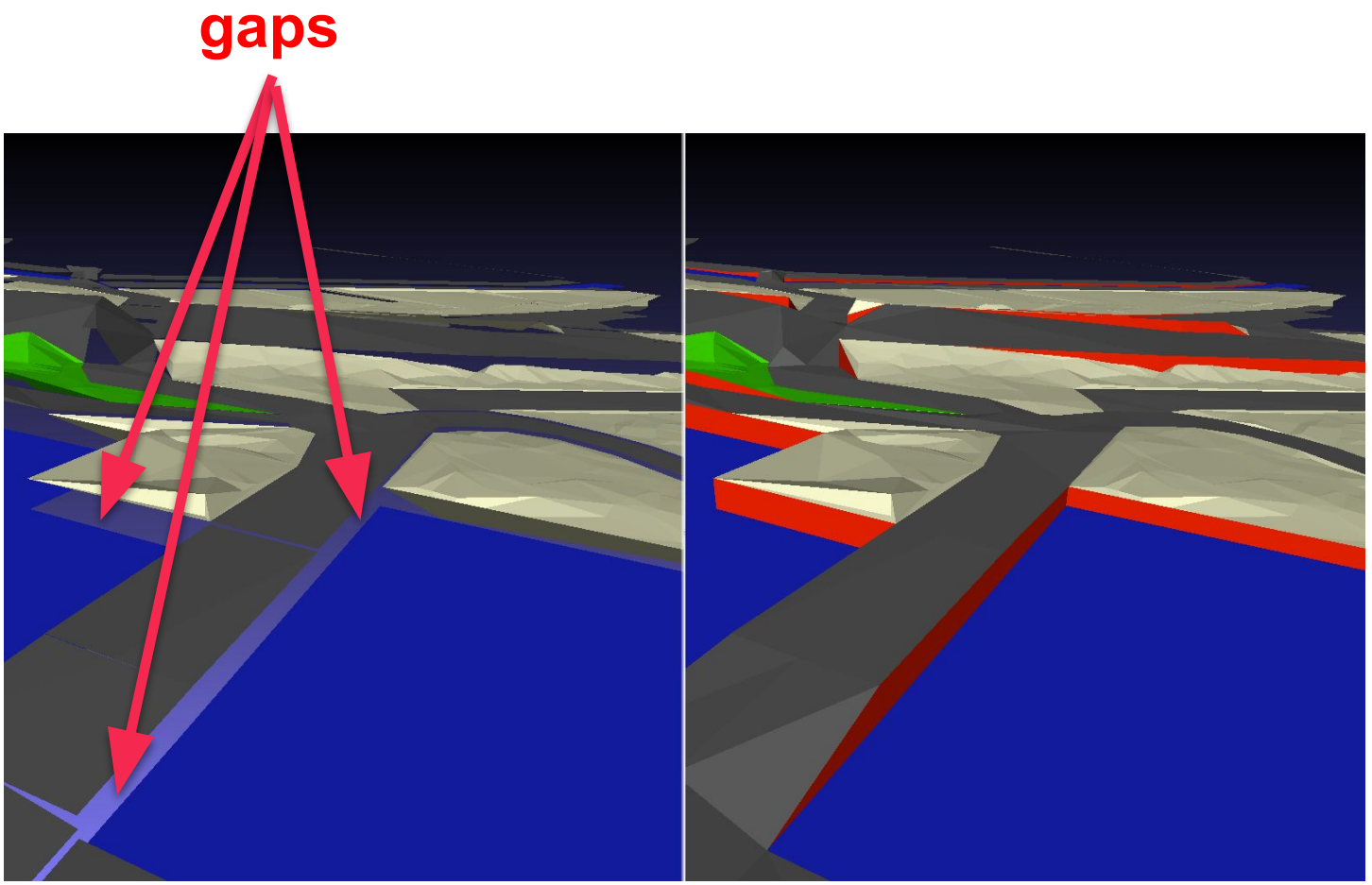


Stitching adjacent polygons with pairwise rules

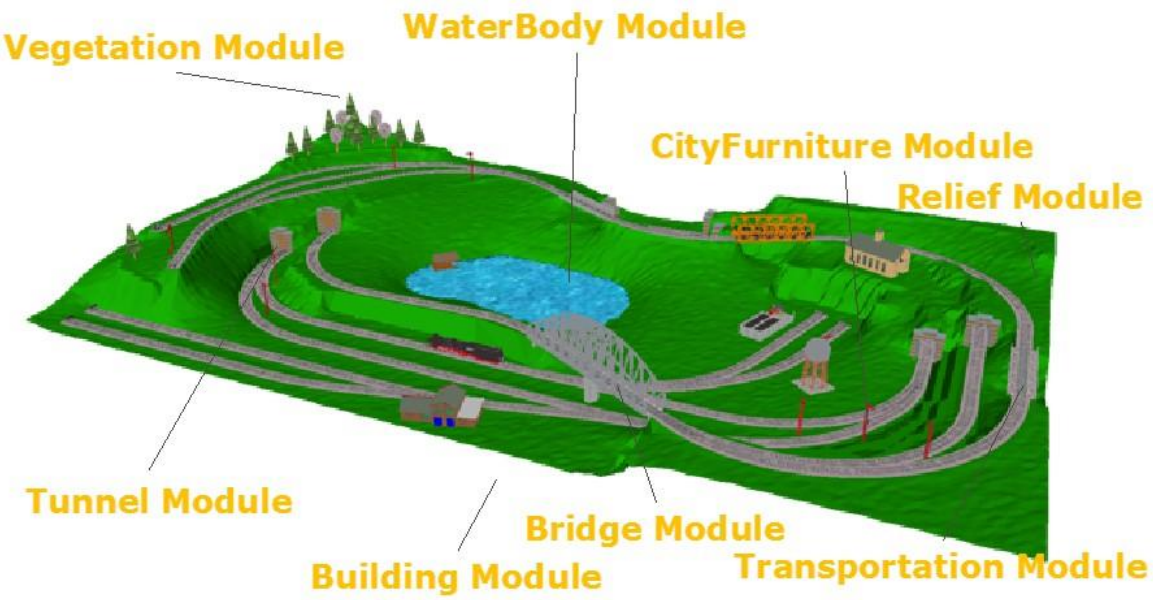
Stitching adjacent polygons with pairwise rules



Stitching adjacent polygons with pairwise rules



Level of Detail (LoD)



Even LoD1 models have different realisations

- Which height is used for extrusion?
  - Gutter? Maximum height? 2/3, 1/2 of roof height?
  - Application dependent
- How calculated? e.g. max height:
  - Highest point that falls in polygon? Median? Using buffer?
- Often users are not aware of possible differences

More awareness is needed

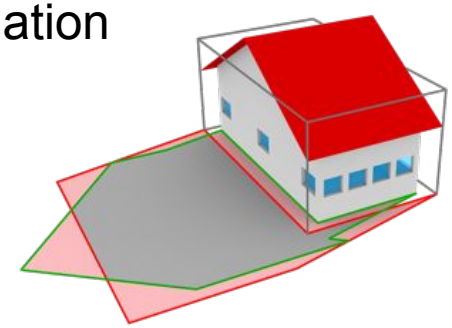
& standardisation

Biljecki, 2016

"LoD2 is more accurate than LoD1"

Effect of acquisition error and level of detail on the accuracy of spatial analyses
Filip Biljecki, G Heuvelink, H Ledoux, J Stoter, Cartography and Geographic Information Science, 45(2): 156-176, 2018.

- Accuracy of acquisition method has more impact on quality of spatial analysis than LoD
- Higher LoDs do not always bring significant improvements
  - E.g. LoD1 versus LoD2 or LoD3 for shadow estimation



3D CMs can be too detailed!

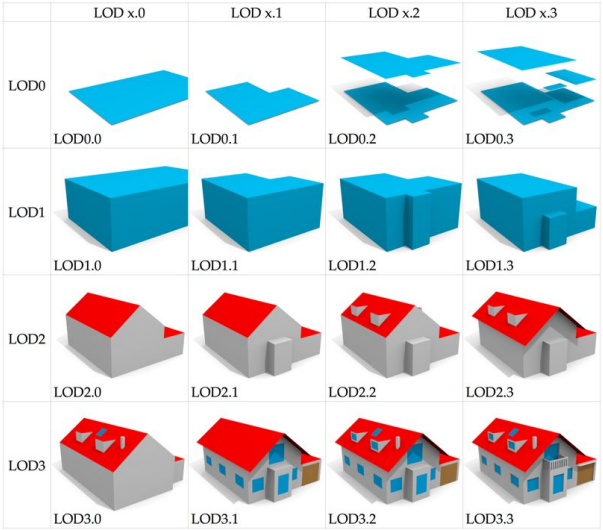
Not always strive for highest LoD, relate it to application

LoD

Level of Detail in CityGML

!=

Level of Development in IFC



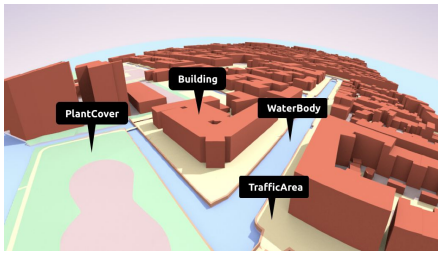
Current 3D city models

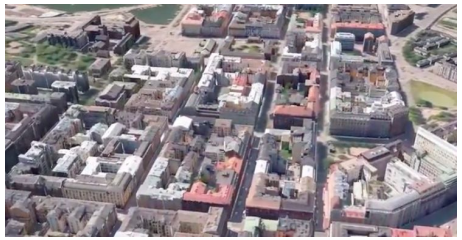
- 3D CMs are rarely generated having needs in mind
- Preparing 3D data for specific applications takes lot of time
- 3D visualisation overvalued; 3D data undervalued
- Errors in 3D CMs; cause errors in outputs

As a consequence, many 3D CMs are available but potentials are underused

3D city models for urban applications

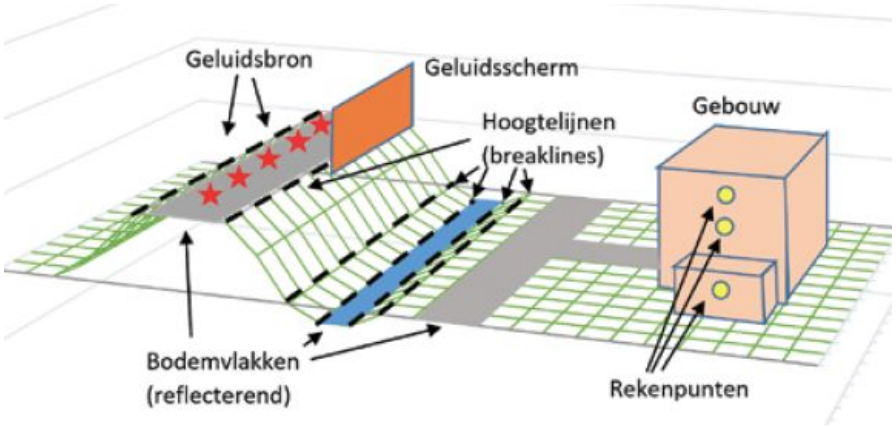
- Semantic 3D CM, not (only) a nice looking textured mesh





- Application specific Level of Detail
- Up-to-date
  - Not only acquisition: also maintenance
- Without errors
- 3D data integration is one of the biggest challenges
  - GIS-BIM

3D CMs for noise simulations



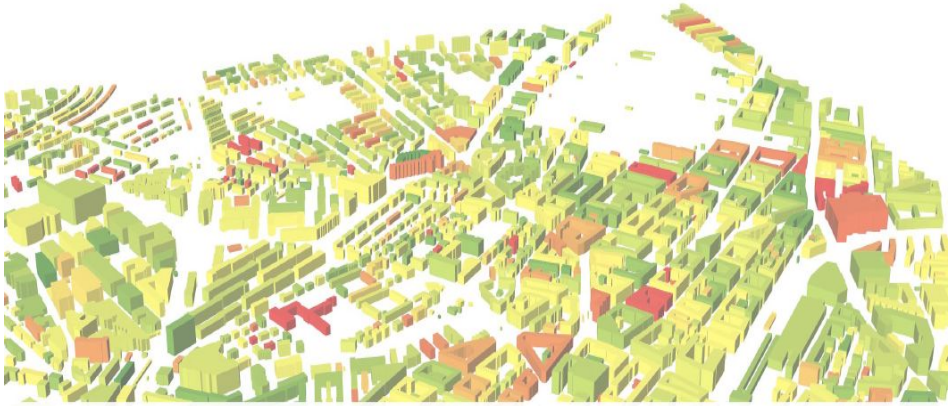
Kluijver & van Tilburg, 2018

Requires:

- LoD1.3 buildings (varying height for one footprint)
- lines that capture height of terrain with as few line as possible

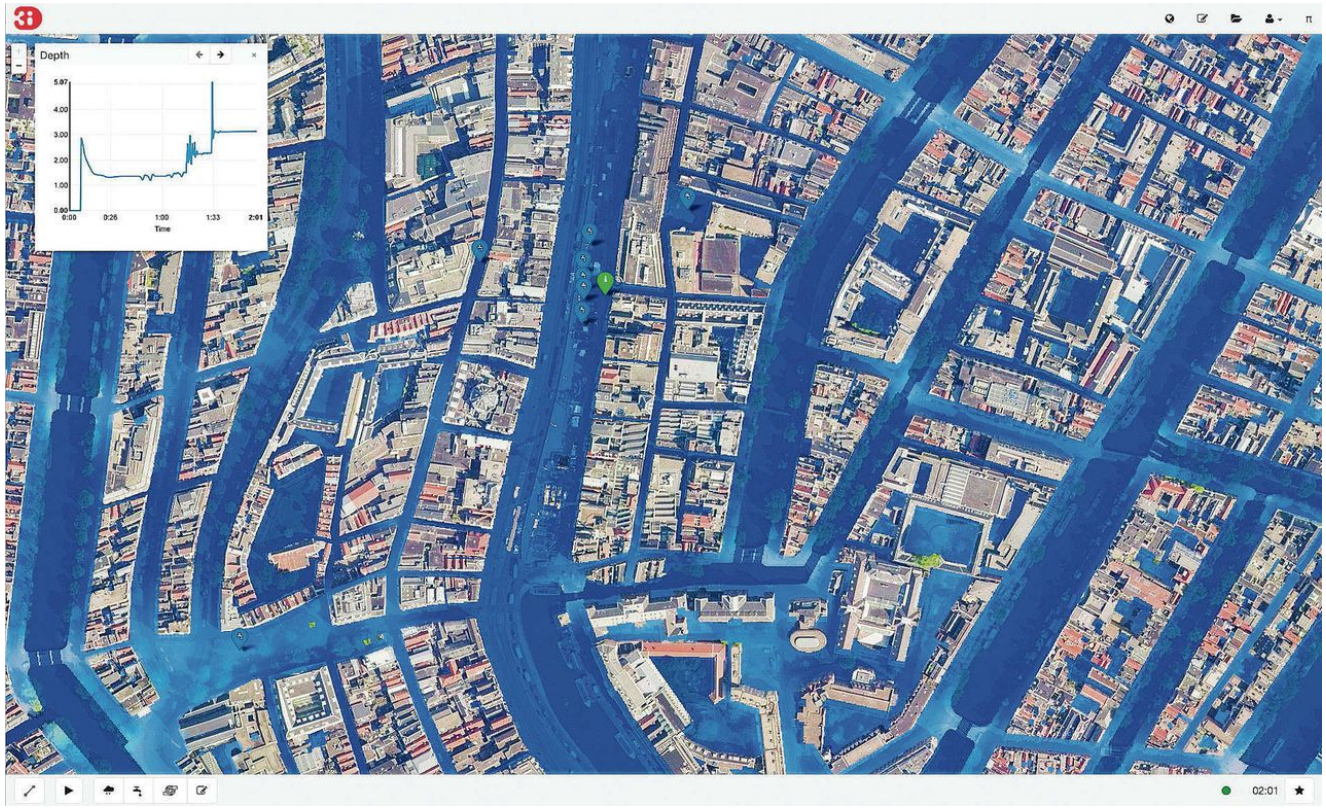
3D CMs for energy consumption simulation

- Roof models do not significantly improve results
- Ceiling, wall, floor
- Inner and outer walls
- Materials of facades
- Solar irradiation for each building



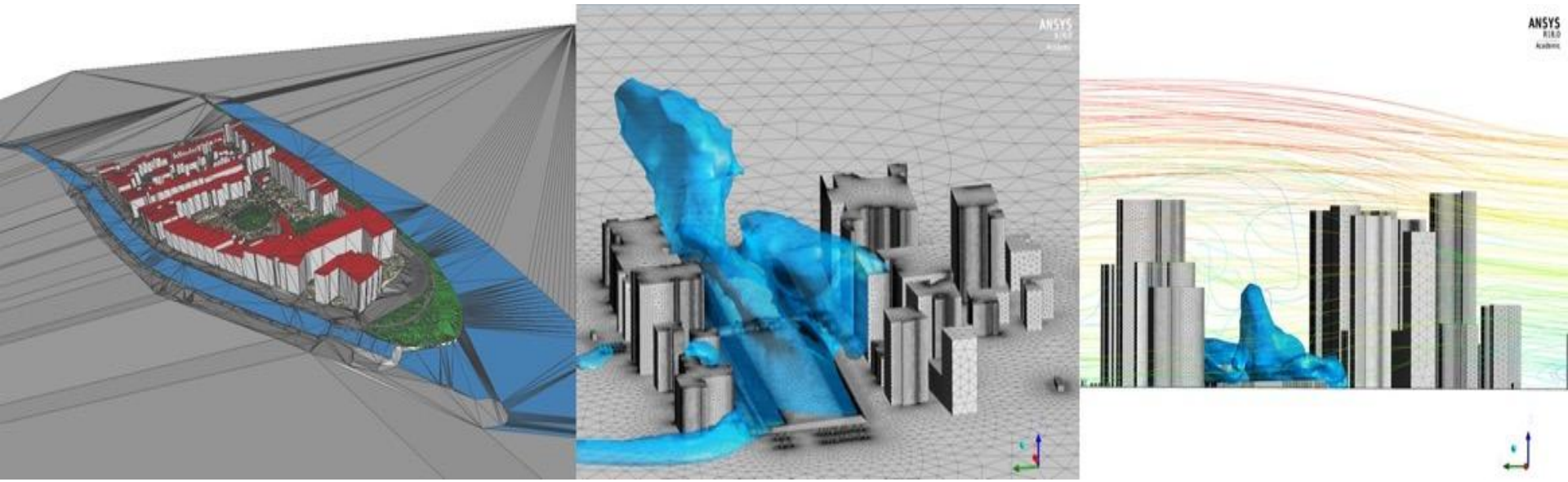
3D CMs for flooding simulations

- highly detailed terrain, with semantics for infiltration and permeability



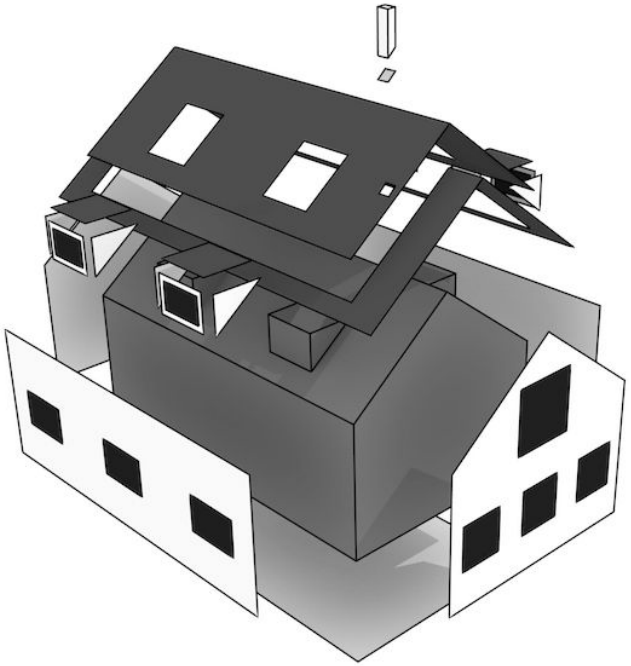
3D CMs for Simulation - CFD

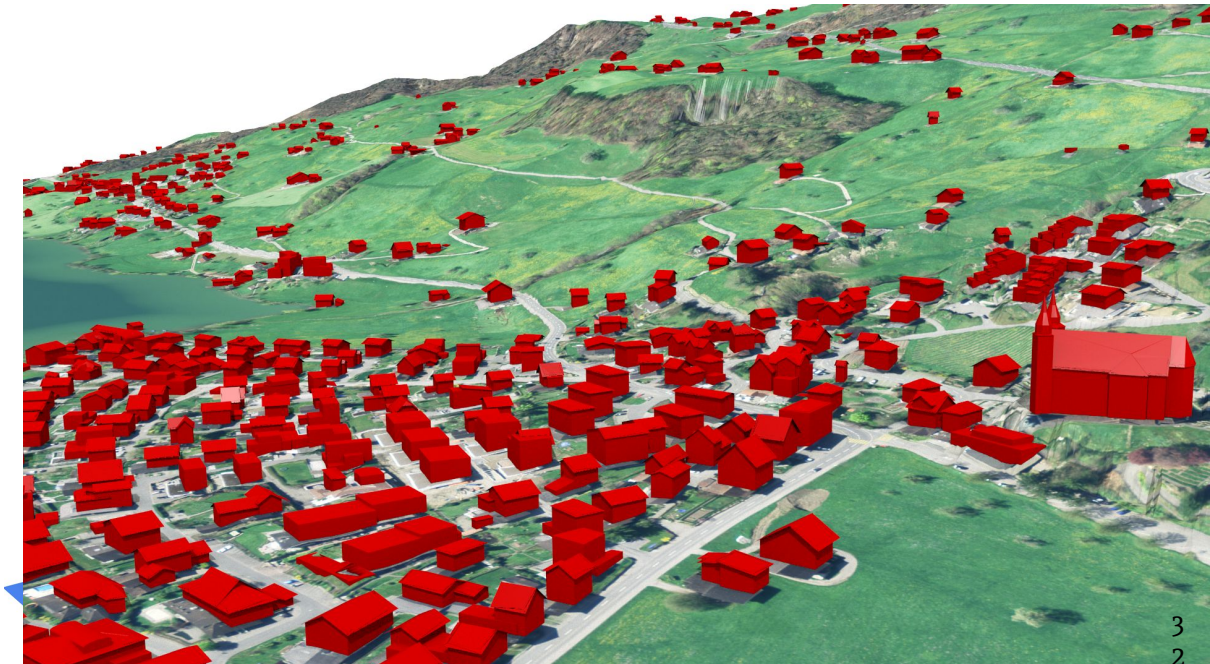
- Computer fluid dynamics modelling (wind, air quality, temperature)
- Requirement: should be 100% closed



CityGML

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

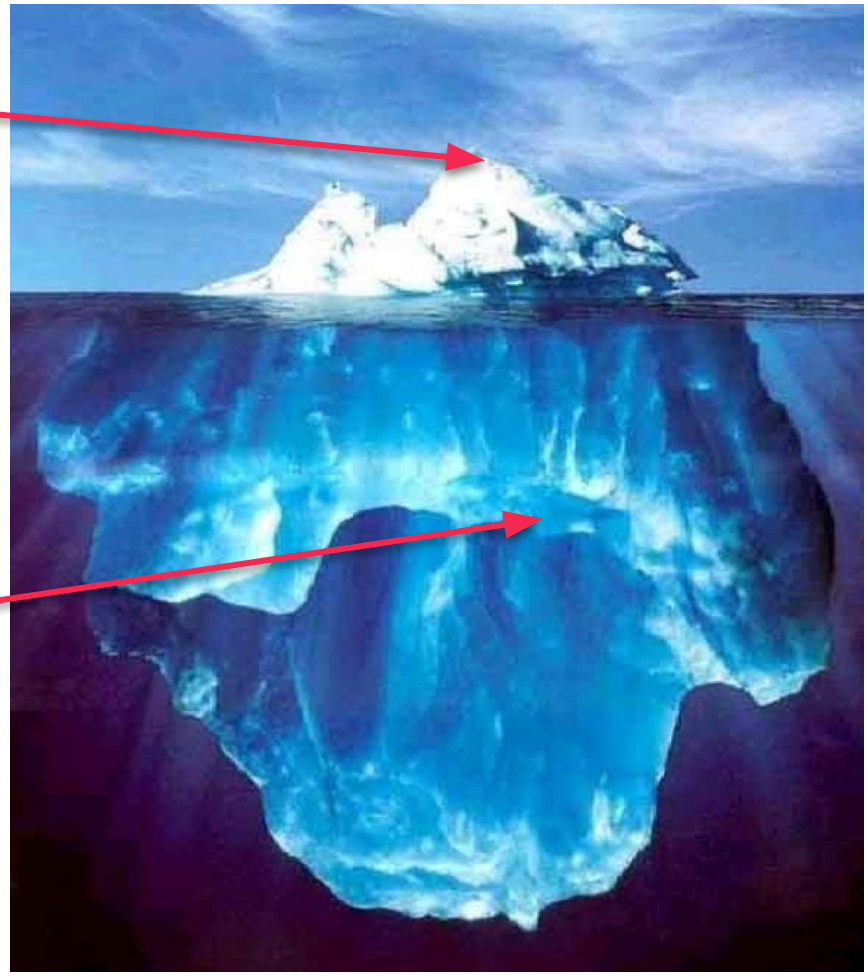




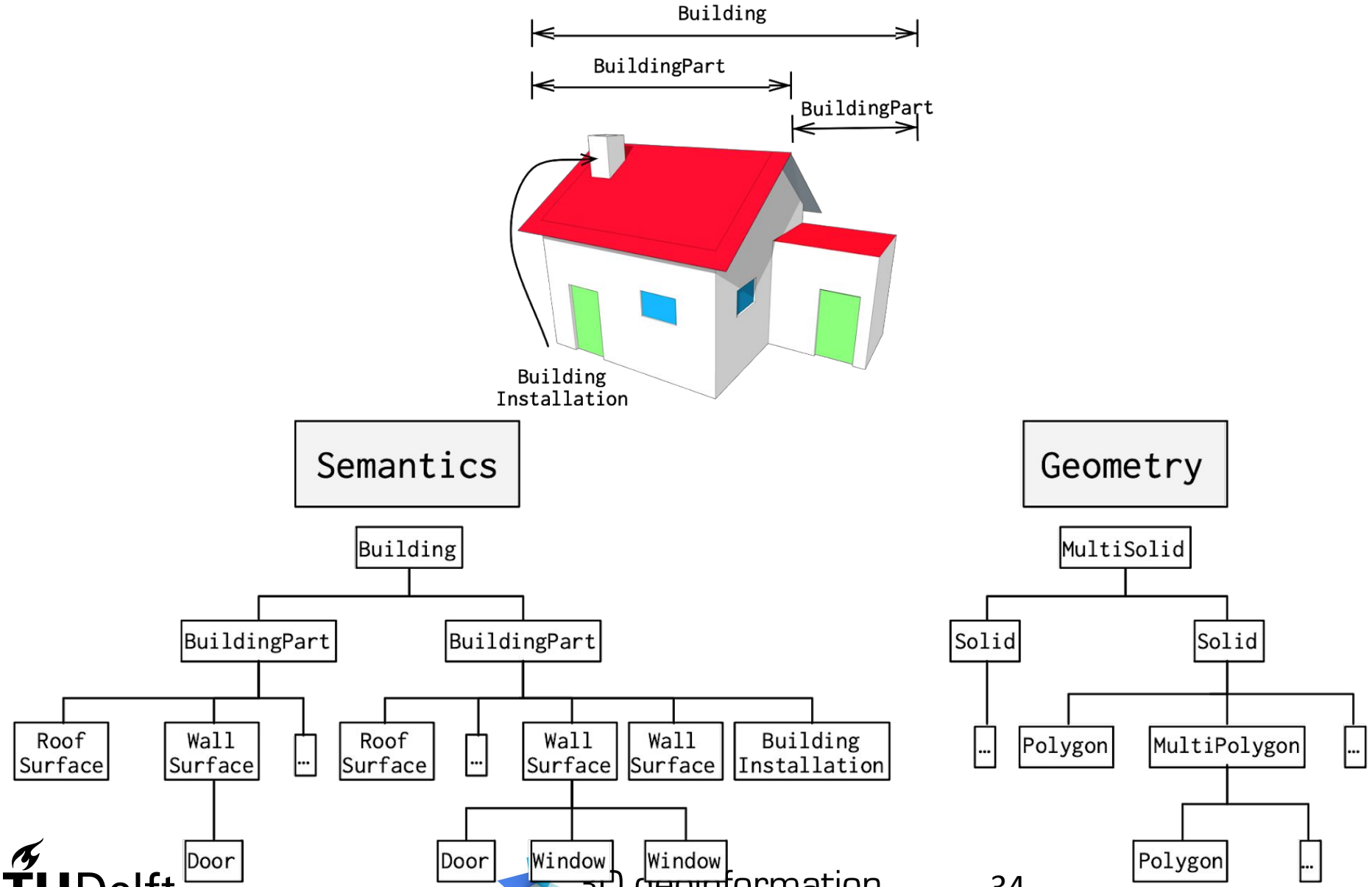
3D city modelling and CityGML == semantics

more than just visualisation

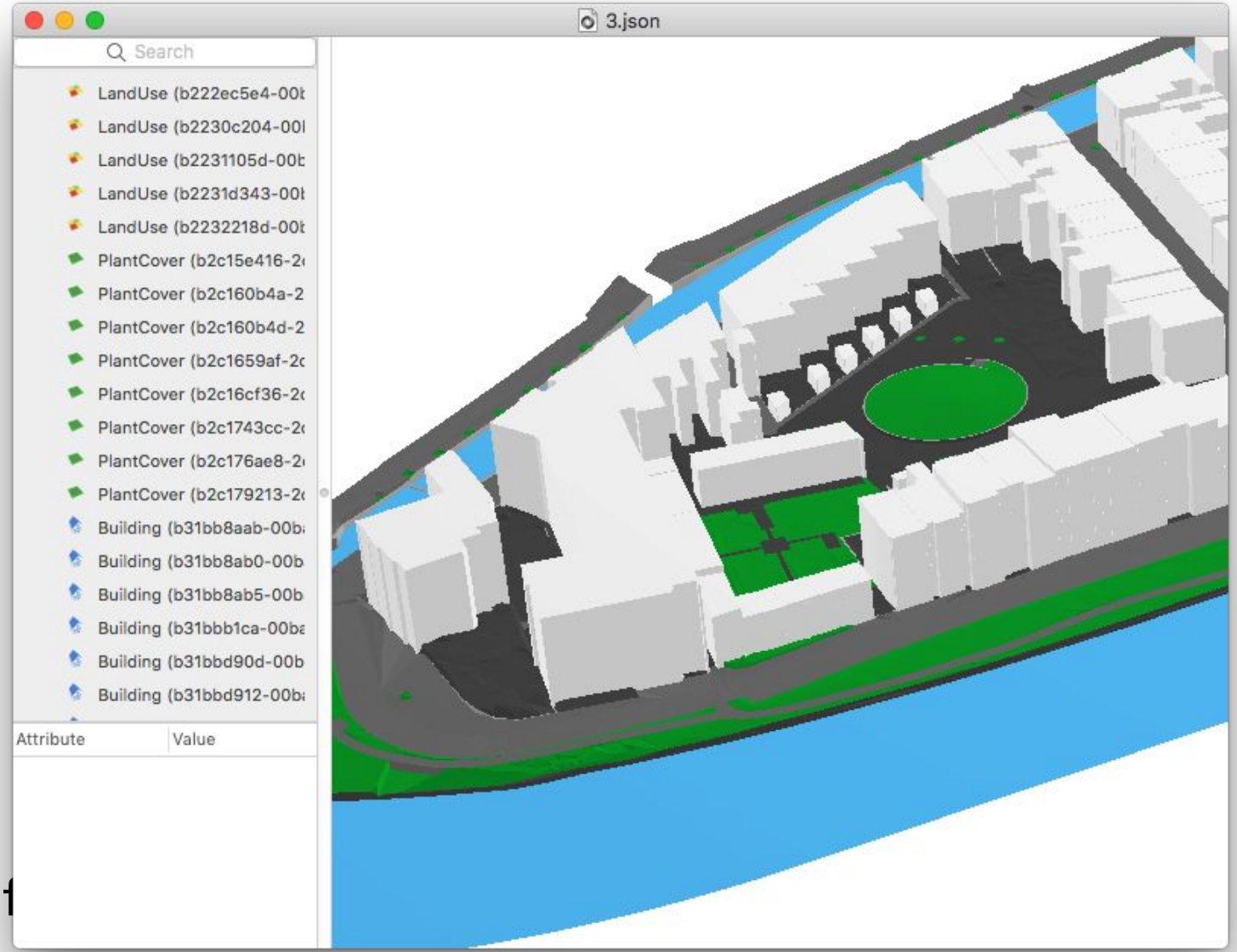
key aspect: semantic modelling



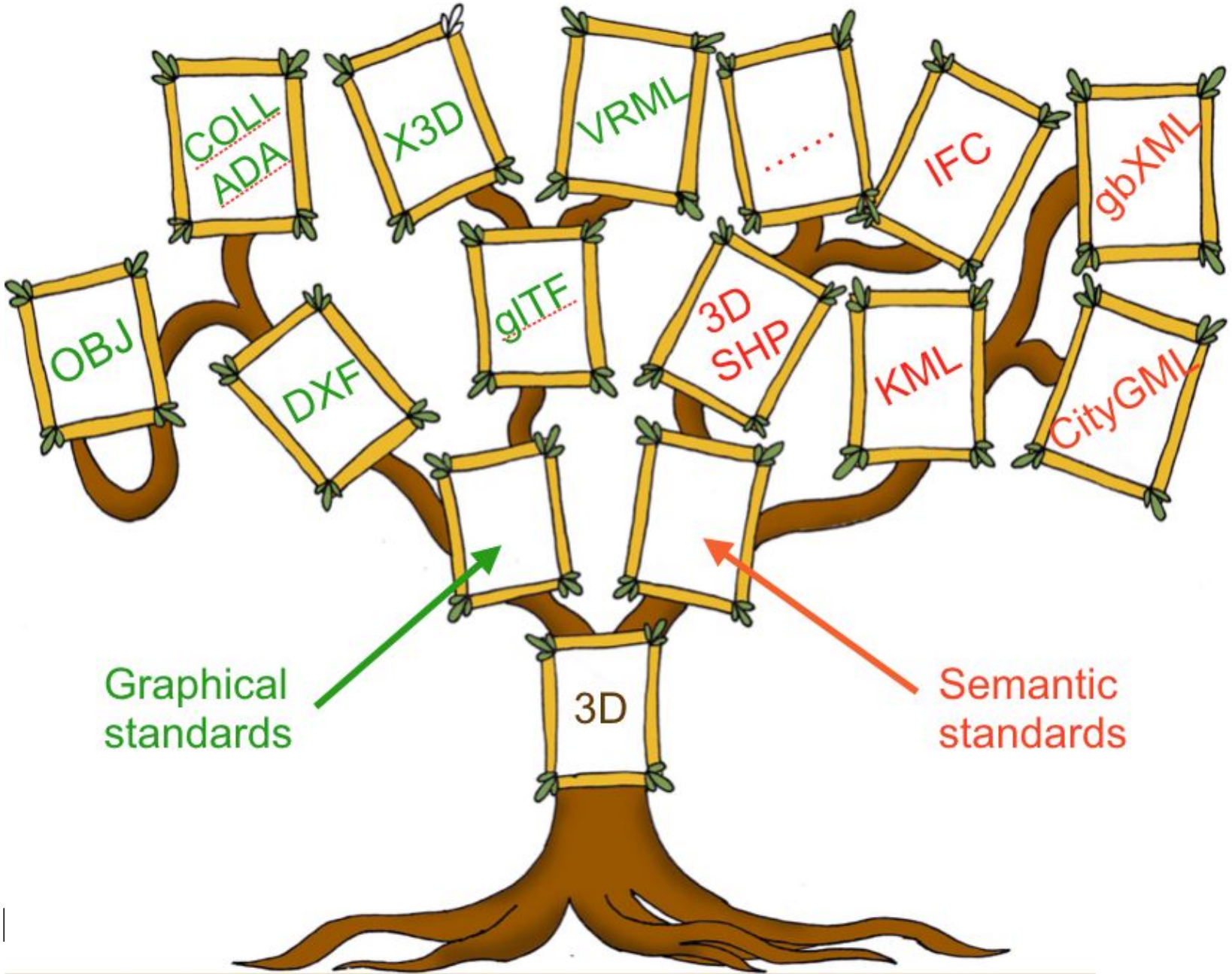
Spatio-semantic coherence



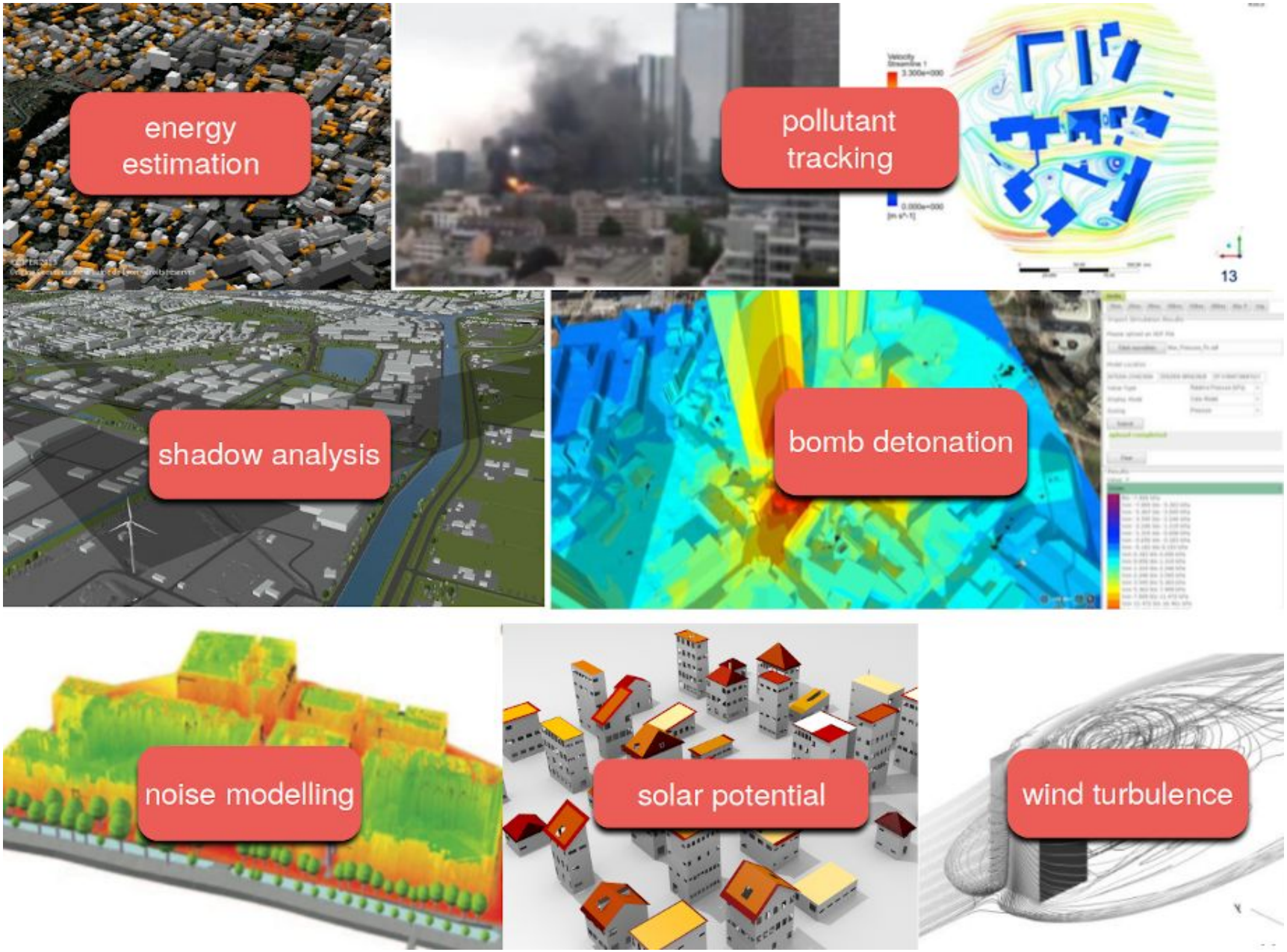
A whole area can be semantically decomposed



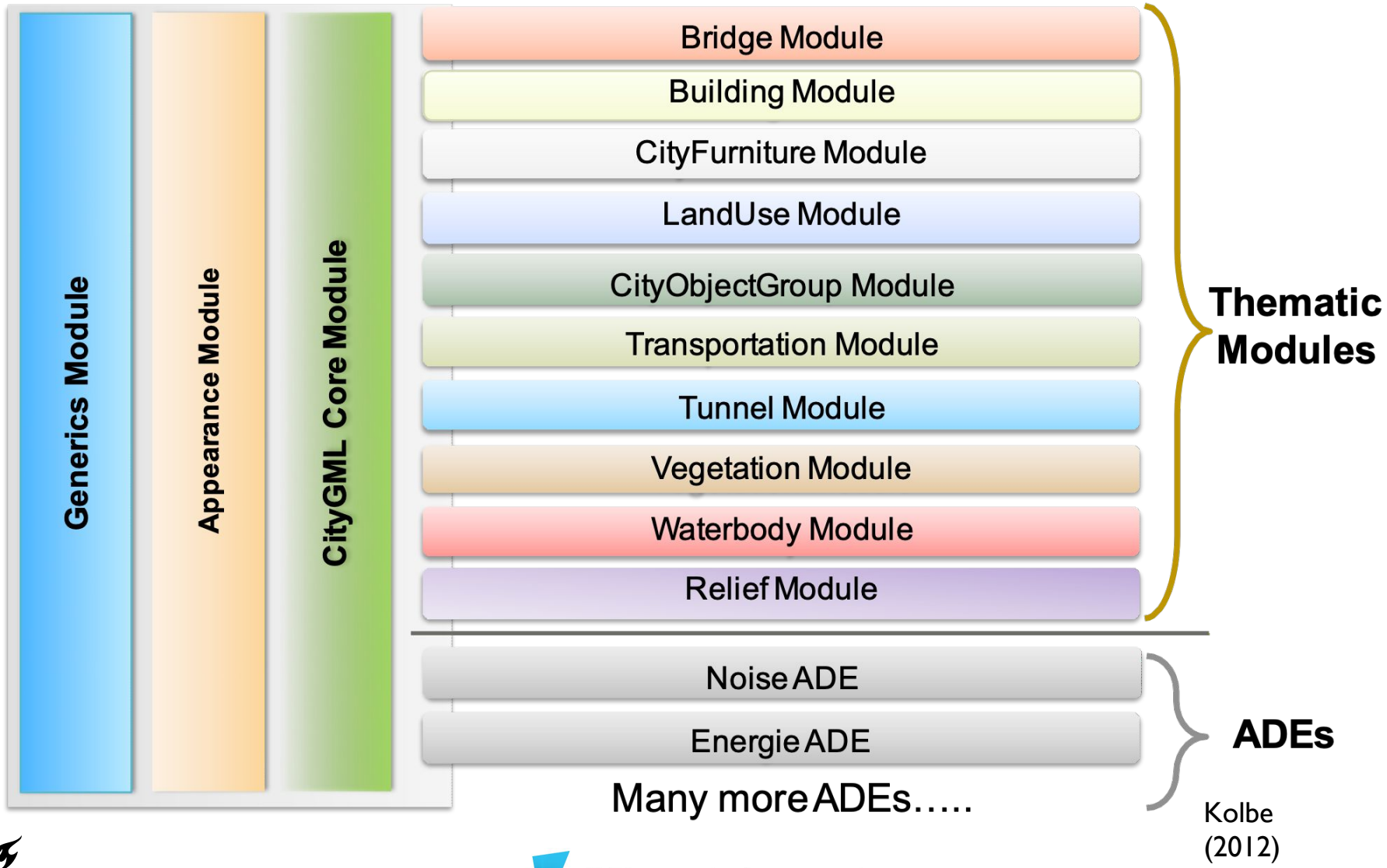
3D formats


COLLADA
X3D
VRML
.....
IFC
gbXML
OBJ
DXF
glTF
3D SHP
KML
CityGML
3D
Graphical standards
Semantic standards

Why semantics are important?

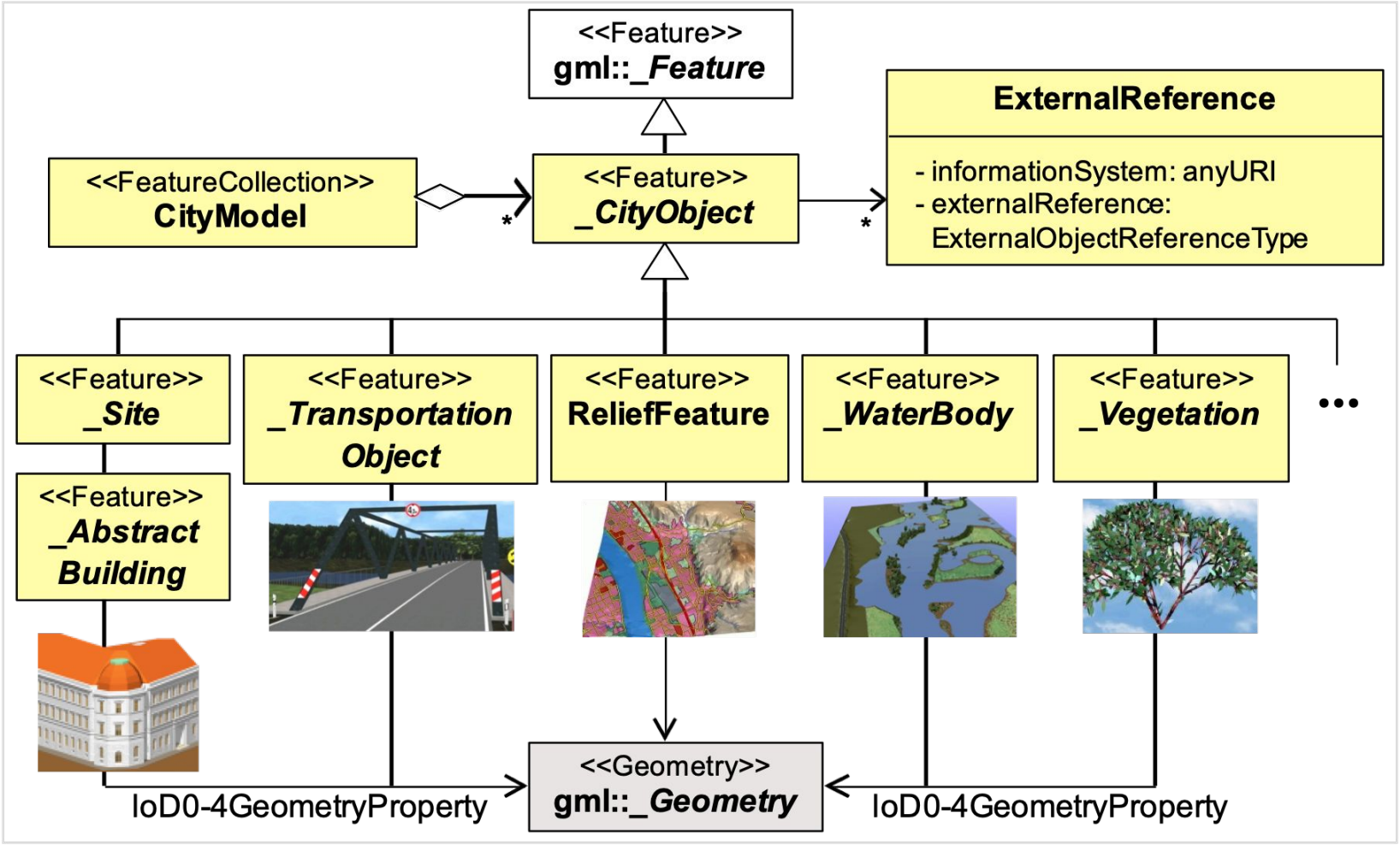

energy estimation
pollutant tracking
shadow analysis
bomb detonation
noise modelling
solar potential
wind turbulence

CityGML = standardisation of the data model



Kolbe (2012)

CityGML = standardisation of the data model



Gröger et al. (2012)

Data model

a GML encoding

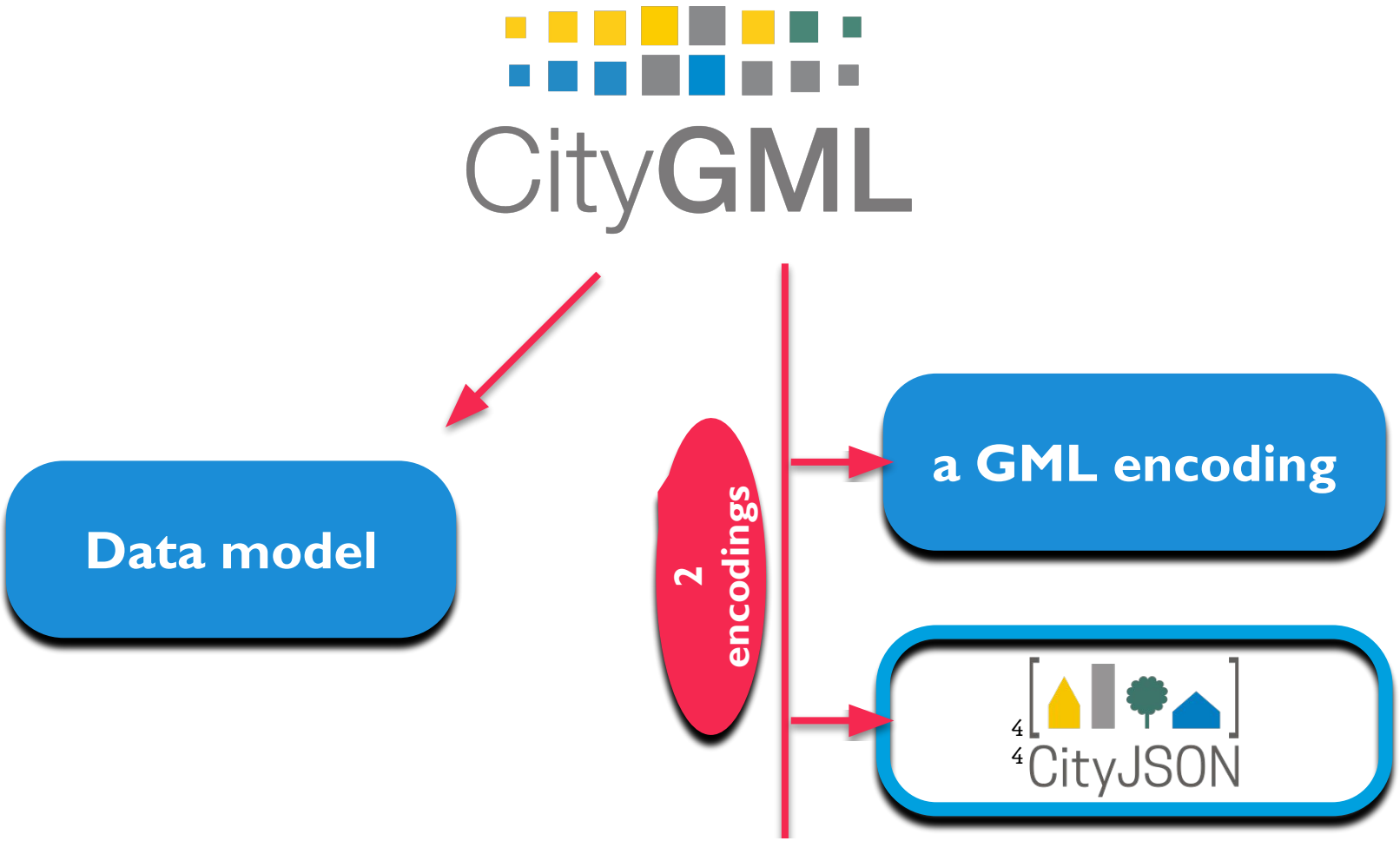
both are called CityGML

A CityGML (the encoding) file

```
<?xml version="1.0" encoding="UTF-8"?>
<CityModel xmlns="http://www.citygml.org/citygml/1/0/0“  ...further namespaces omitted>
   <gml:name>Cologne</gml:name>
   <gml:boundedBy>
      <gml:Envelope
         srsName="urn:ogc:def:crs,crs:EPSG:6.12:31466,crs:EPSG:6.12:5783">
         <gml:pos>  5659800.0   2561800.0   15.9 </gml:pos>
         <gml:pos>  5662200.0   2564200.0   95.7</gml:pos>
      </gml:Envelope>
   </gml:boundedBy>
   <!-- now come the CityObjects like Buildings, DTM, Roads etc. -->
   <cityObjectMember>
      <Building gml:id=“Building0815">
      <!-- shown on following slides -->
      </Building>
   </cityObjectMember>
   <!-- more CityObjects here -->
</CityModel >
```

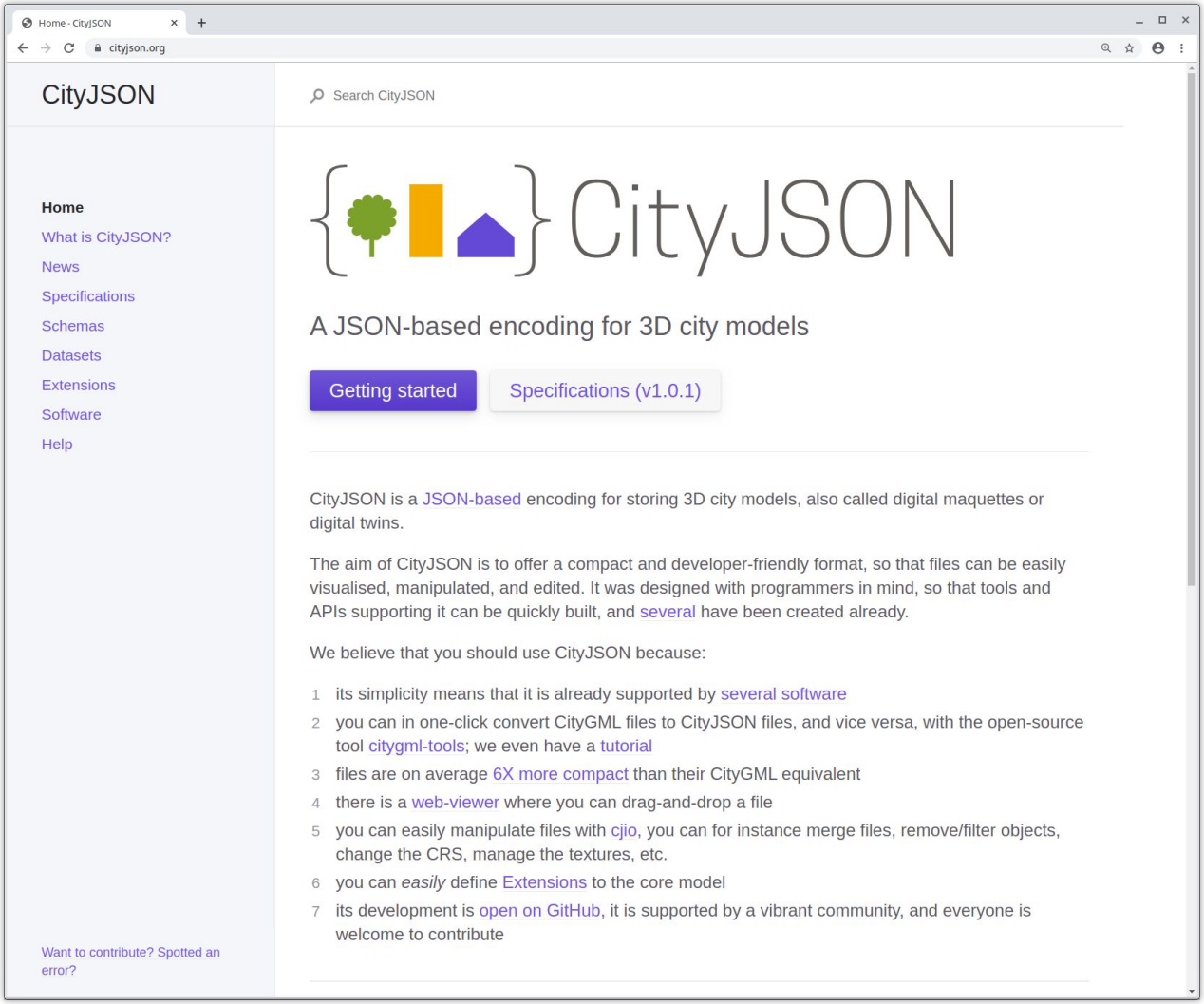
CityGML files are very complex

Complete, but verbose & complex, and therefore often difficult to work with


CityGML
Data model
2 encodings
a GML encoding
CityJSON

CityGML in JSON

cityjson.org



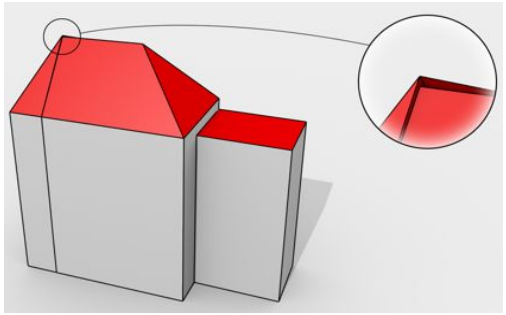
v1.0.1 released

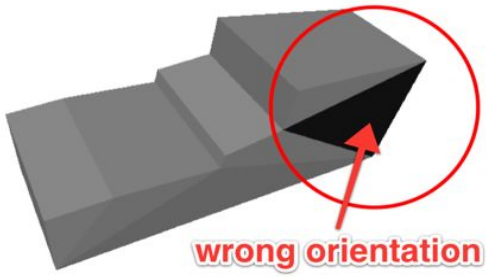
- all CityGML modules mapped 💪🏽
- software for full conversion CityGML <-> CityJSON
- Extensions (ADEs in CityGML world) are possible
- software:
  - create (3dfier)
  - visualise online (viewer.cityjson.org) and locally (Azul, QGIS, Blender)
  - manipulate (cjio, FME coming soon…)

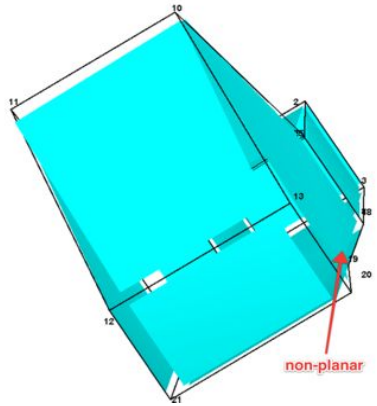
Compression factor == ~7X

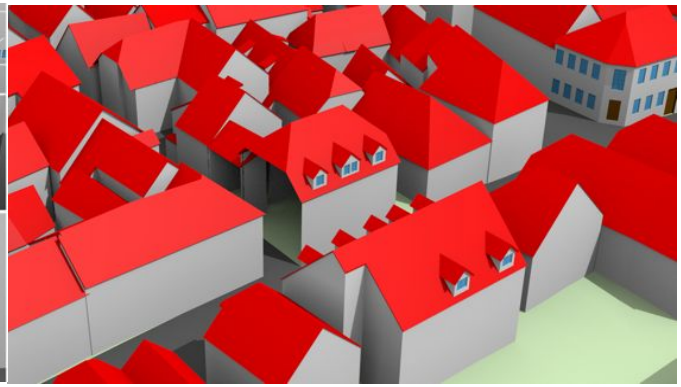
file	CityGML size (original)	CityGML size (w/o spaces)	textures?	CityJSON	CityJSON compressed	compression factor
CityGML demo "GeoRes"	4.3MB	4.1MB	yes	582KB	524KB	8.0
CityGML v2 demo "Railway"	45MB	34MB	yes	4.5MB	4.3MB	8.1
Den Haag "tile 01"	23MB	18MB	no, material	3.1MB	2.9MB	6.2
Montréal VM05	56MB	42MB	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

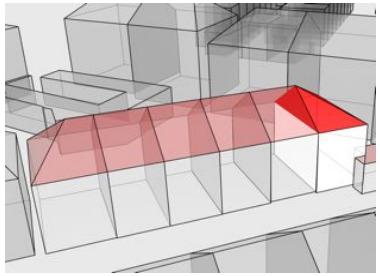
Errors = ~~common~~ always in 3D

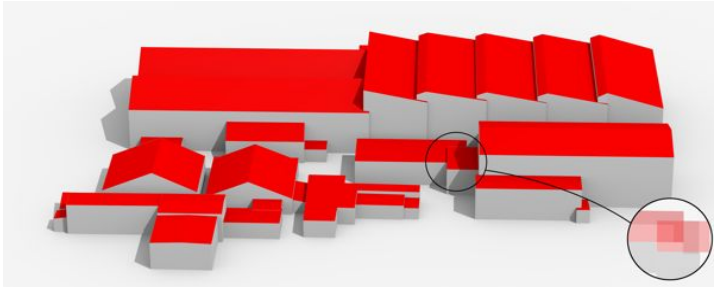






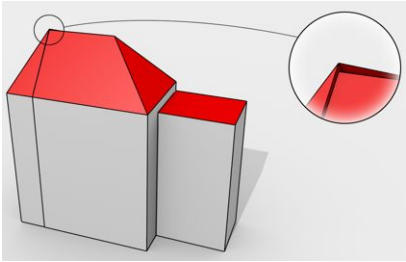






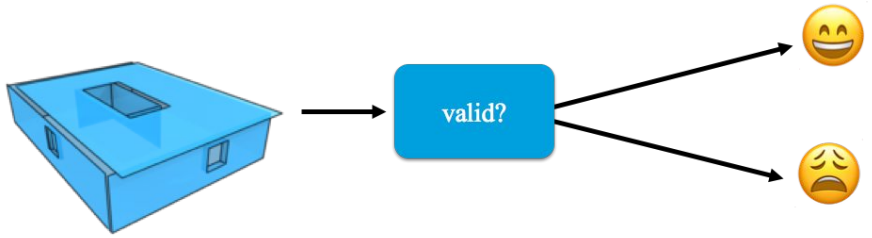
Errors in 3D models

- Not visible-> users are not aware
- Not problematic in specific software or applications
- But not possible to reuse 3D data in other software and applications



Software to validate 3D city models

geovalidation.bk.tudelft.nl



1. distinct vertex
2. closedness of the rings of every surface
3. orientation of points within a surface (with inner rings)
4. planarity of surfaces
5. non-self intersection of surfaces
6. non-overlapping inner rings on a surface
7. orientation of normal vectors
8. "watertightness" of every shell
9. "connectedness" of the interior
10. how inner/outer shells interact with each others
11. ...

ISO19107 & OGC

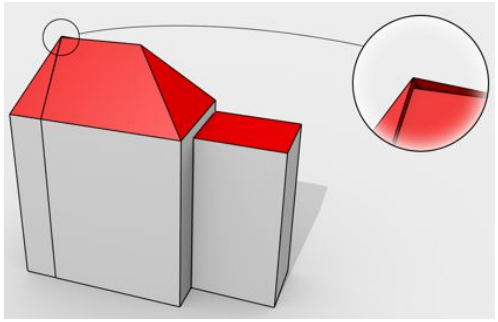
Test of quality of 3D CMs

To understand quality of existing 3D data sets

- Applied to 37 datasets in 9 countries
  - 3.6m buildings
  - 16m 3D primitives
  - 40m surfaces

The most common geometric and semantic errors in CityGML datasets
Filip Biljecki, Hugo Ledoux, Xin DU, Jantien Stoter, Kean Huat SOON, Victor KHOO
ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-2/W1: 13-22, 2016.

Results of validating existing 3D city models

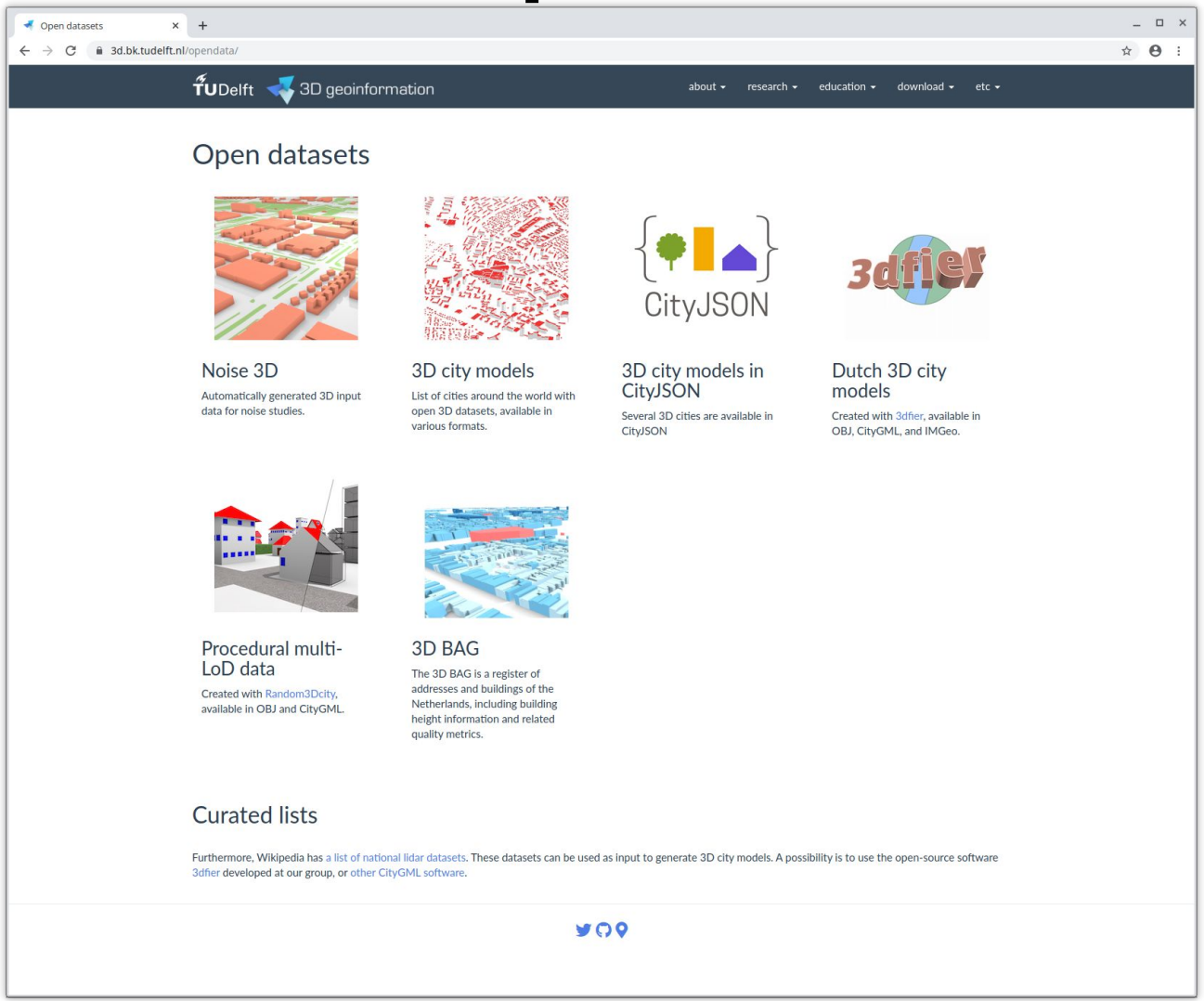


- 3D city models without errors are rare
- Most valid models are LoD1 models
- Many errors can be automatically fixed or prevented:
  - Missing faces; geometries not properly snapped; orientation of surfaces; non planar faces (often caused by deviations of few cm only)
  - Easier to prevent than correct errors afterwards

Reconstruct valid 3D models, if you want your 3D data to be (re)used

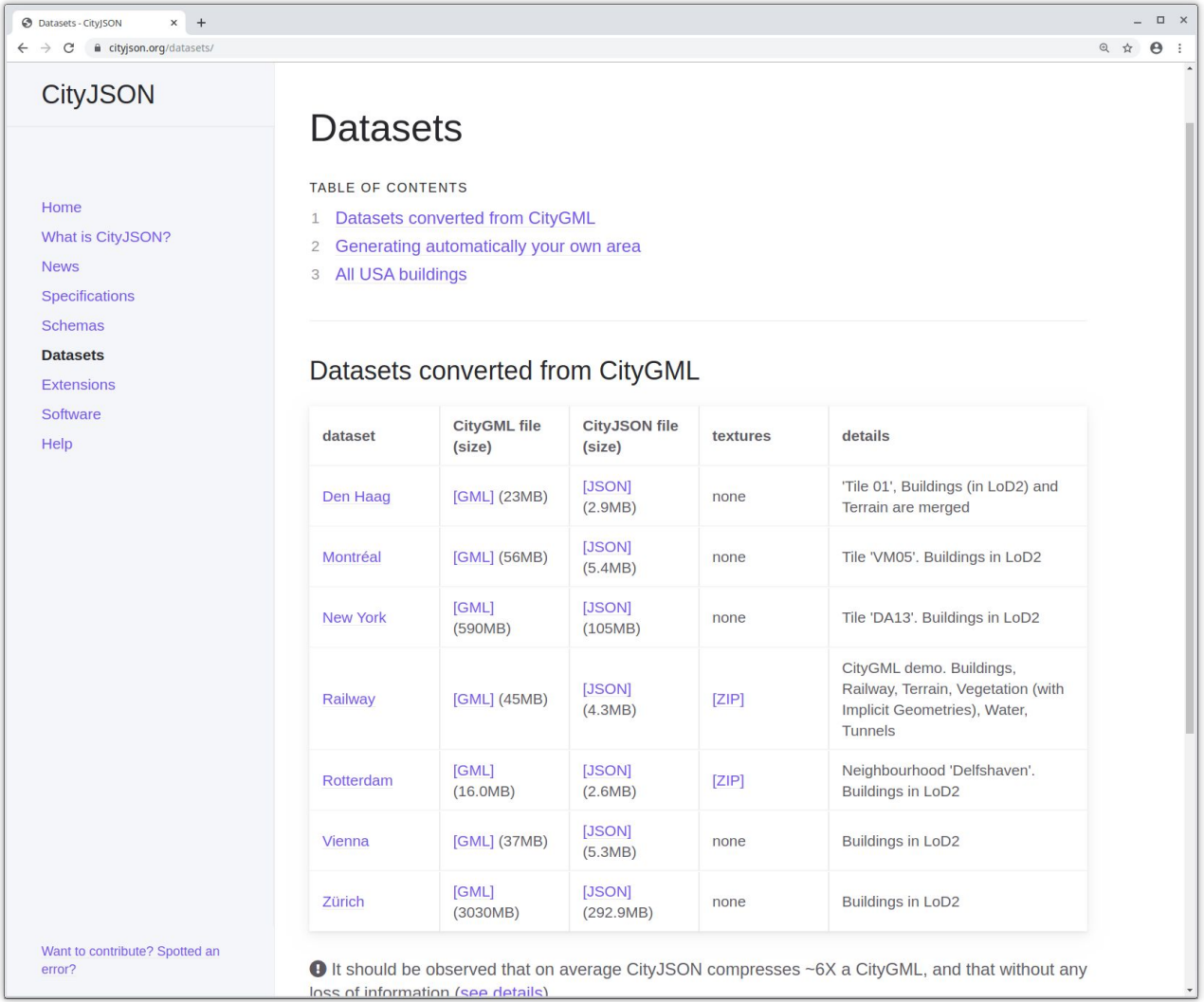
Where to find 3D City Models?

3d.bk.tudelft.nl/opendata



Where to find 3D City Models?

cityjson.org/datasets



Datasets - CityJSON

cityjson.org/datasets/

CityJSON

- Home
- What is CityJSON?
- News
- Specifications
- Schemas
- **Datasets**
- Extensions
- Software
- Help

Want to contribute? Spotted an error?

Datasets

TABLE OF CONTENTS

1 Datasets converted from CityGML
2 Generating automatically your own area
3 All USA buildings

Datasets converted from CityGML

dataset	CityGML file (size)	CityJSON file (size)	textures	details
Den Haag	[GML] (23MB)	[JSON] (2.9MB)	none	'Tile 01', Buildings (in LoD2) and Terrain are merged
Montréal	[GML] (56MB)	[JSON] (5.4MB)	none	Tile 'VM05'. Buildings in LoD2
New York	[GML] (590MB)	[JSON] (105MB)	none	Tile 'DA13'. Buildings in LoD2
Railway	[GML] (45MB)	[JSON] (4.3MB)	[ZIP]	CityGML demo. Buildings, Railway, Terrain, Vegetation (with Implicit Geometries), Water, Tunnels
Rotterdam	[GML] (16.0MB)	[JSON] (2.6MB)	[ZIP]	Neighbourhood 'Delfshaven'. Buildings in LoD2
Vienna	[GML] (37MB)	[JSON] (5.3MB)	none	Buildings in LoD2
Zürich	[GML] (3030MB)	[JSON] (292.9MB)	none	Buildings in LoD2

It should be observed that on average CityJSON compresses ~6X a CityGML, and that without any loss of information (see details)

Thank you!

Tom Commandeur

t.j.f.commandeur@tudelft.nl

Balázs Dukai

b.dukai@tudelft.nl

3D Geoinformation Group

3d.bk.tudelft.nl