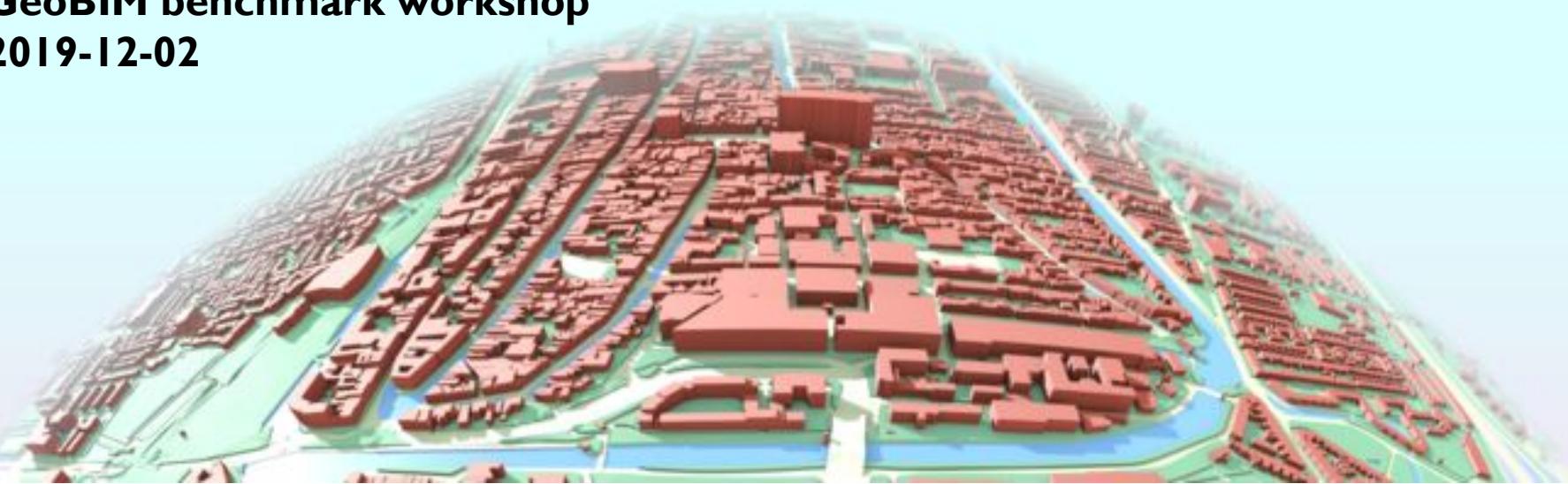


3D City Models, an overview

Tom Commandeur, Balázs Dukai

GeoBIM benchmark workshop

2019-12-02

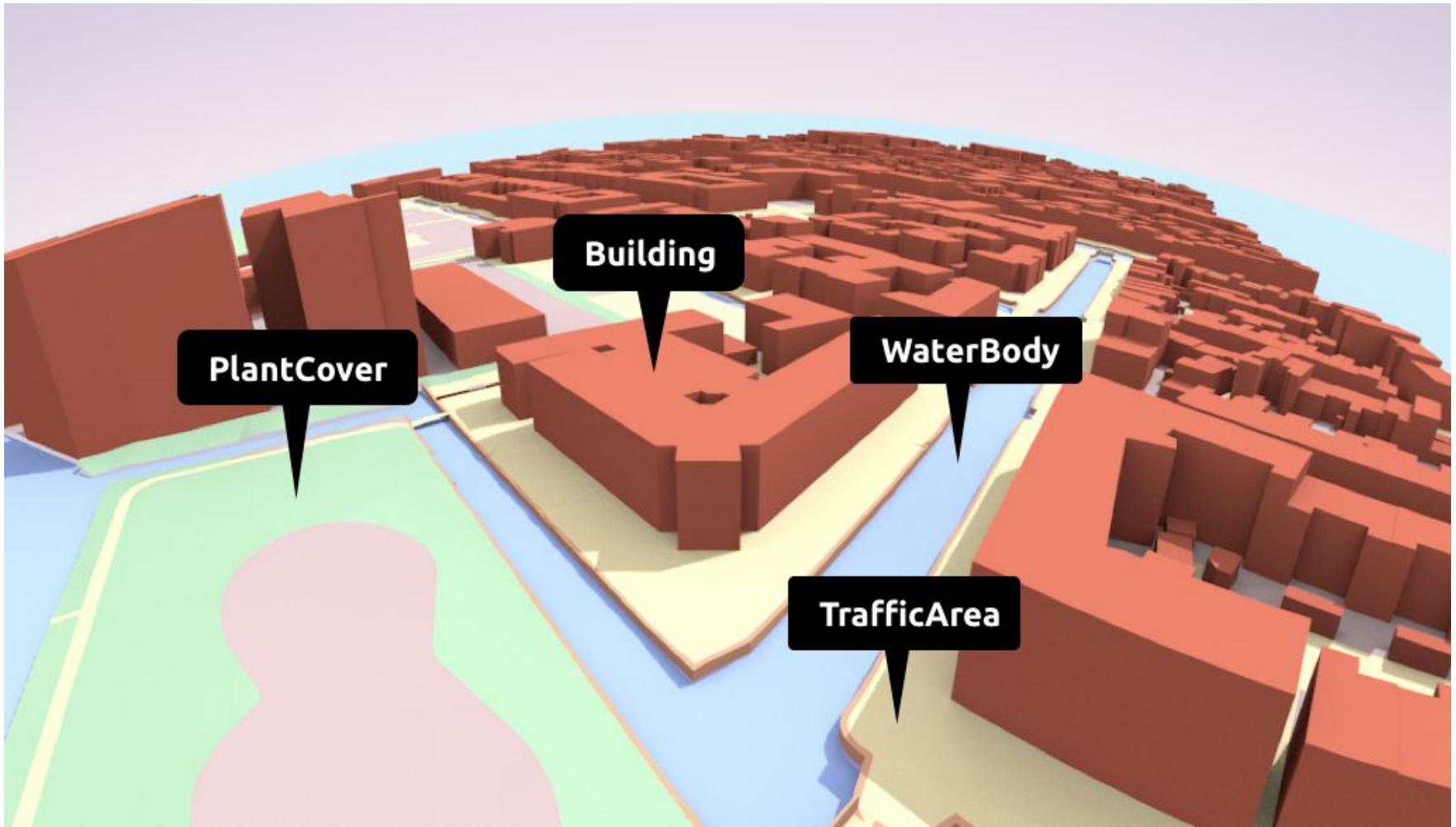


Mesh models

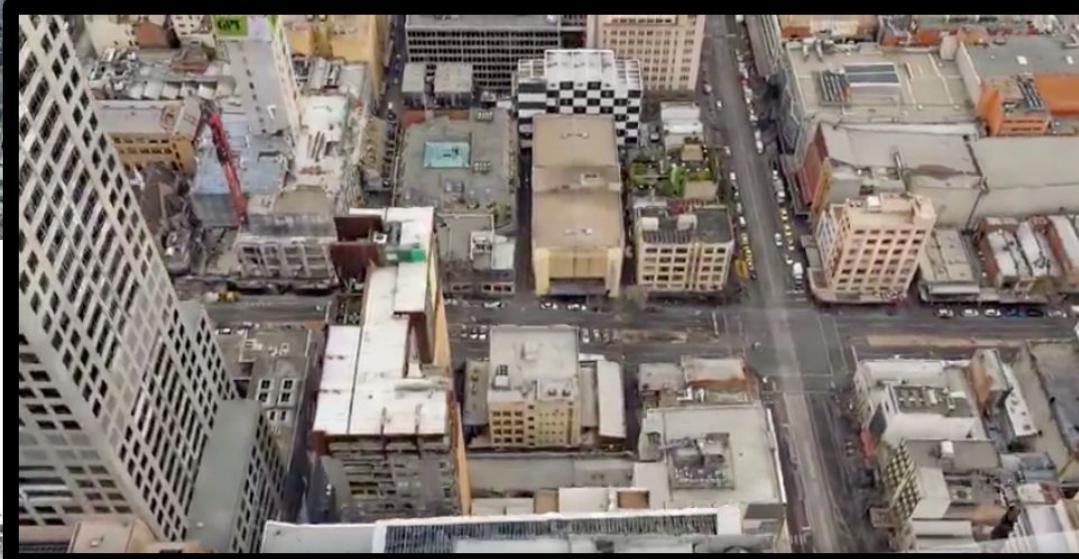


Source: earth.google.com

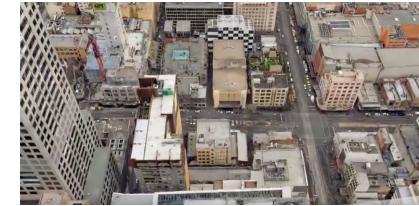
Semantic models



Relatively easy to reconstruct 3D city models



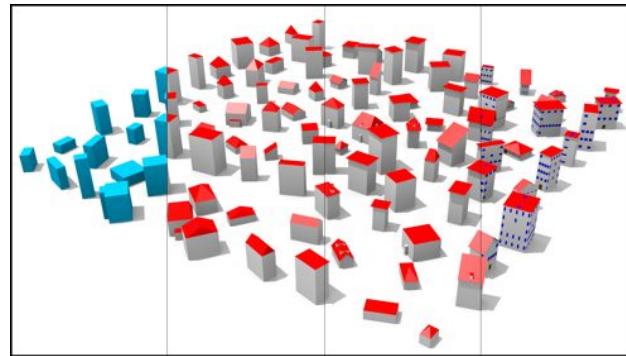
**“The more realistic it looks, the better”,
but.....**



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

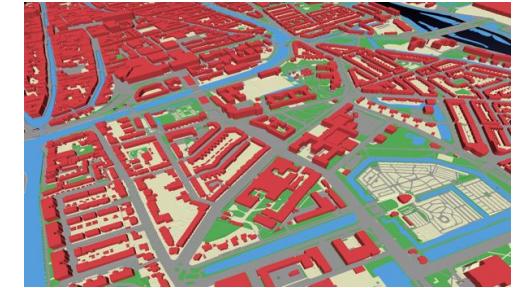
3D city models differ a lot (even for same areas), due to difference in

- 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



Energy



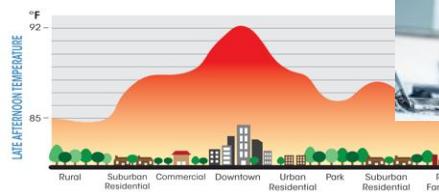
Where do I find
useful data



Flooding



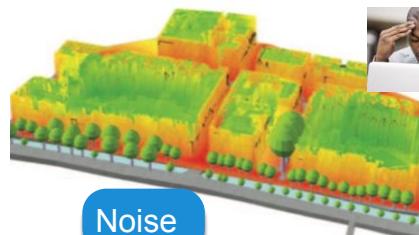
Shadow



urban heat island



Solar potential



Noise



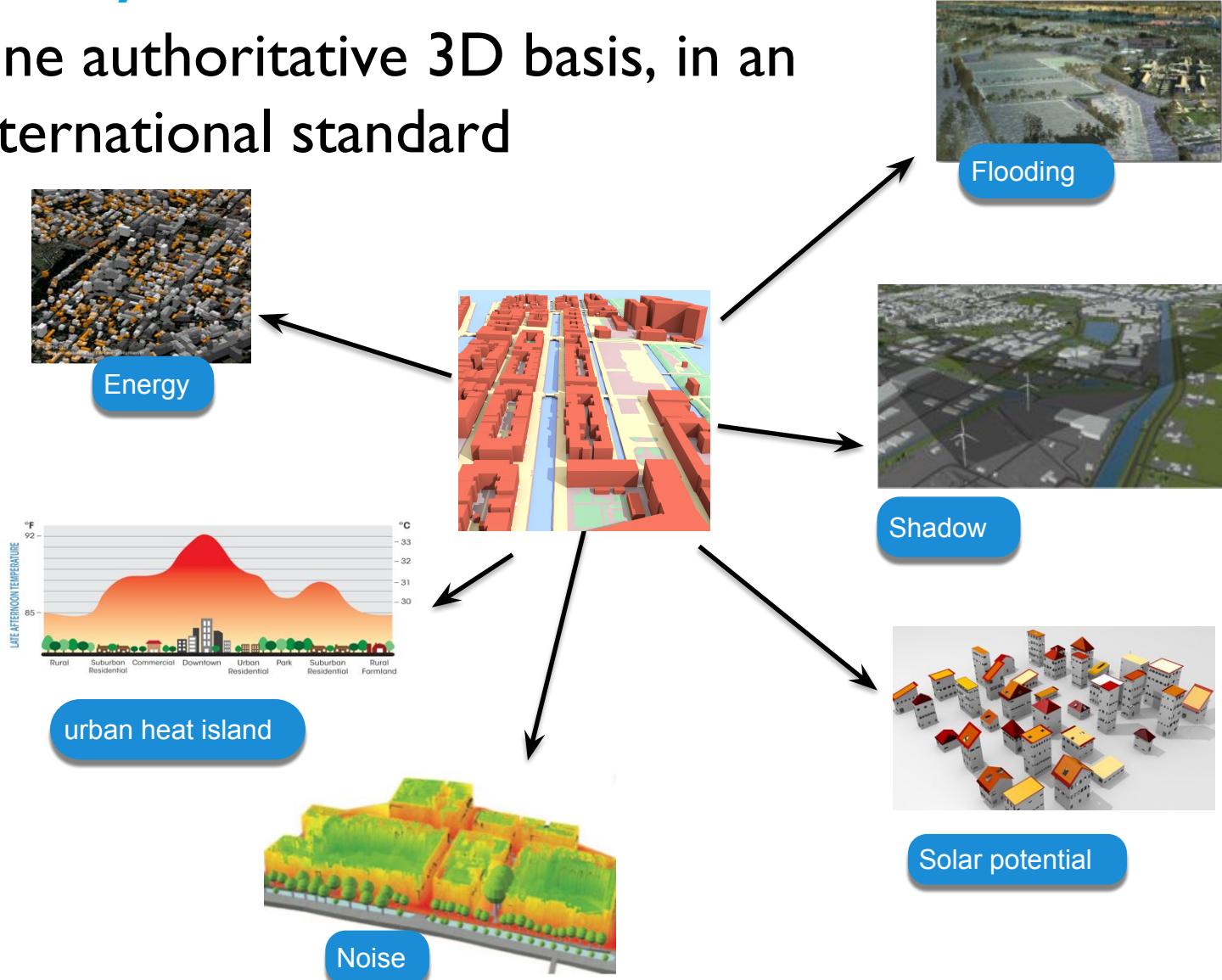
3D geoinformation

3D city models for Smart Cities

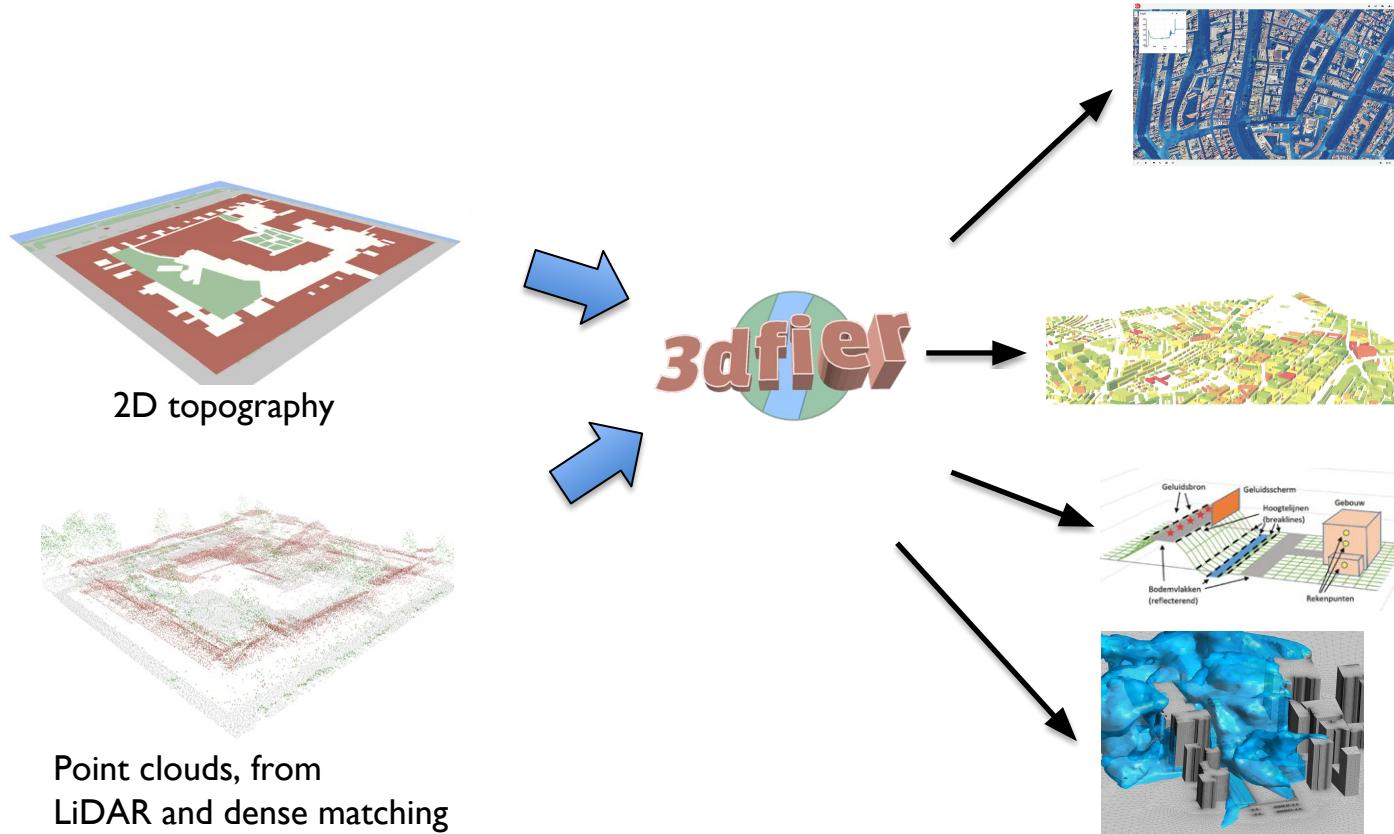


3D city models for Smart Cities

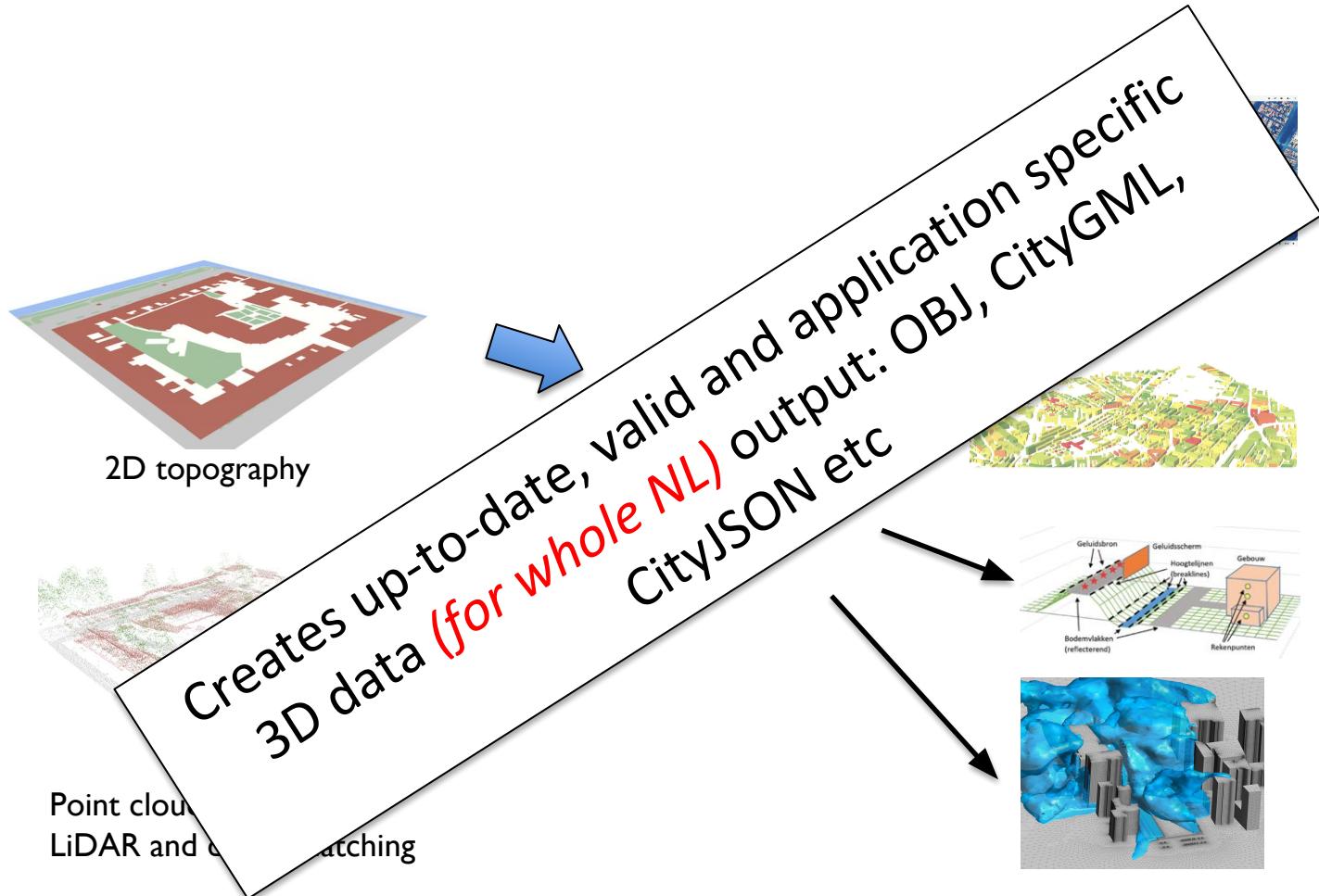
One authoritative 3D basis, in an international standard



Open software to reconstruct 3D models



Open software to reconstruct 3D models



Input: any 2D datasets

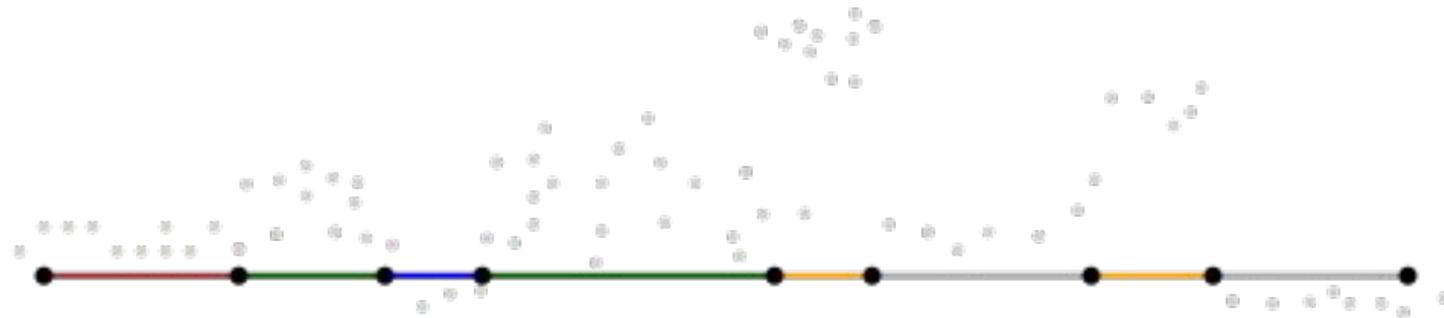


We assign each polygon to a class:

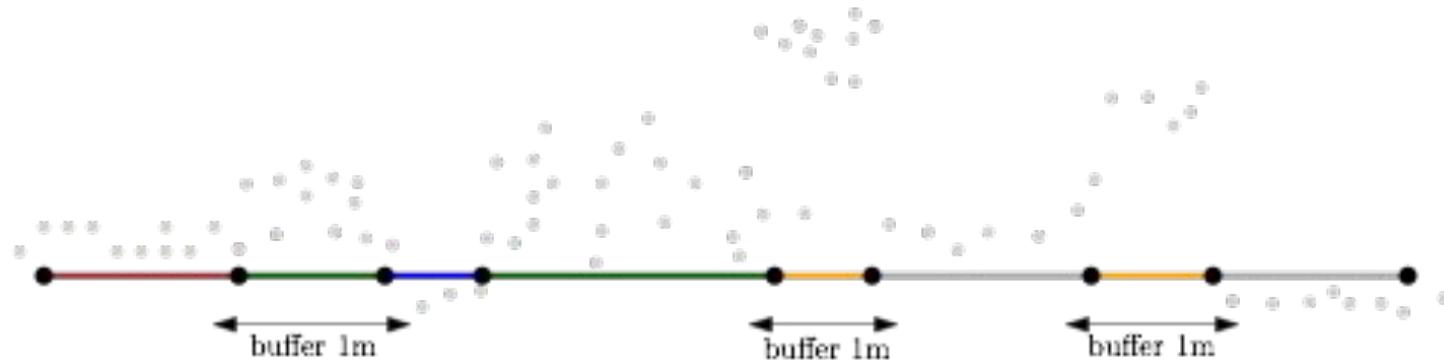
1. Building
2. Water
3. Road
4. Vegetation
5. Terrain
6. Separation
7. Bridge



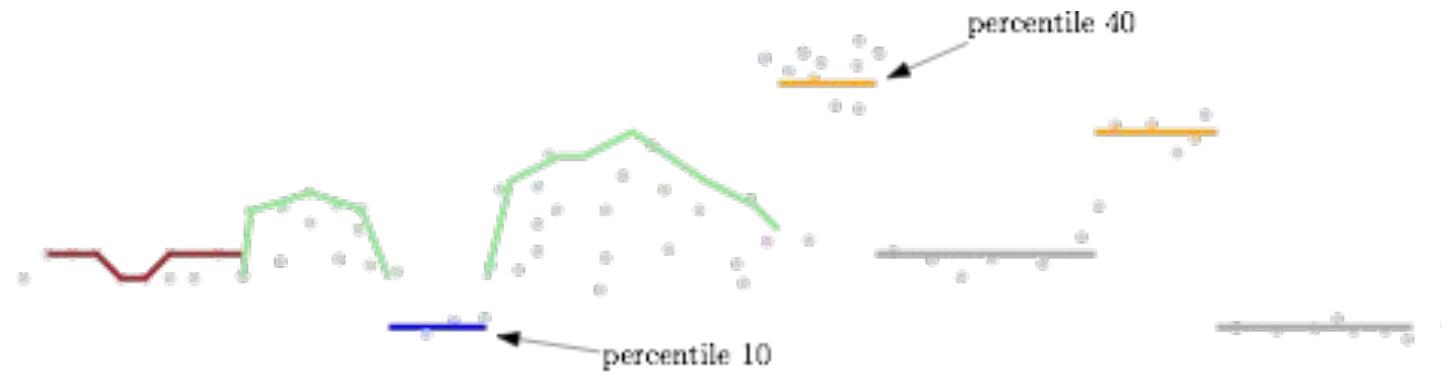
Assign each LiDAR point to polygons



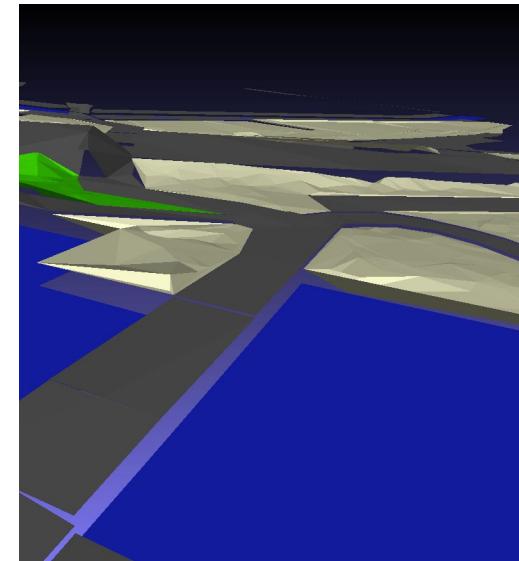
Assign each LiDAR point to polygons



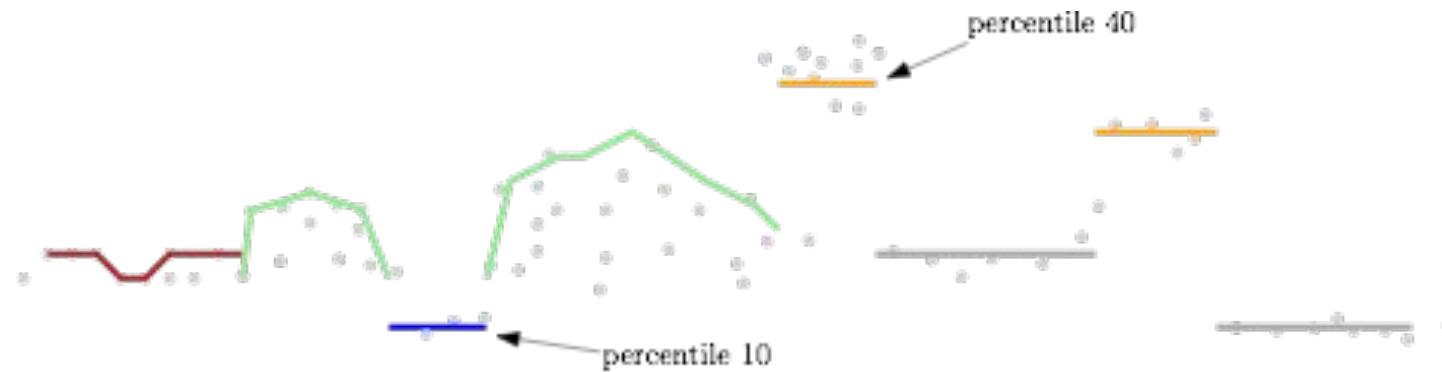
Lifting each polygon based on simple rules (eg avg)



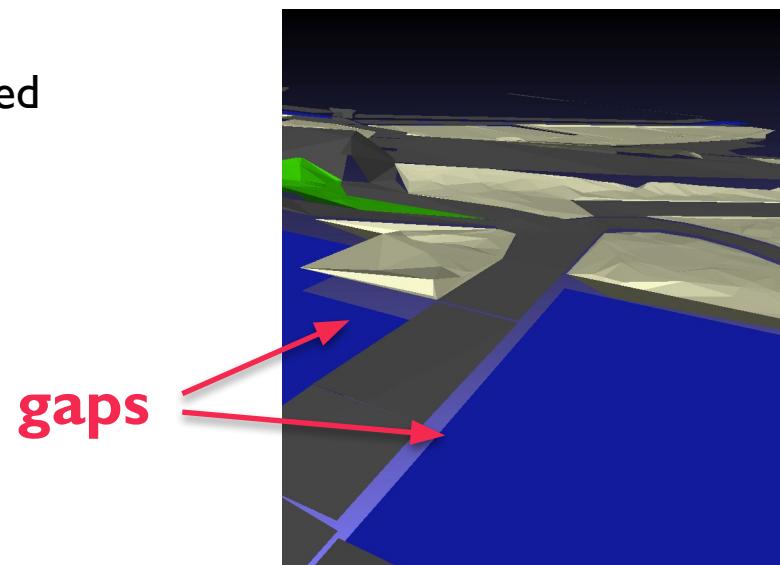
Terrain and vegetation classes have LiDAR points added within the boundary
Other classes only vertices are lifted



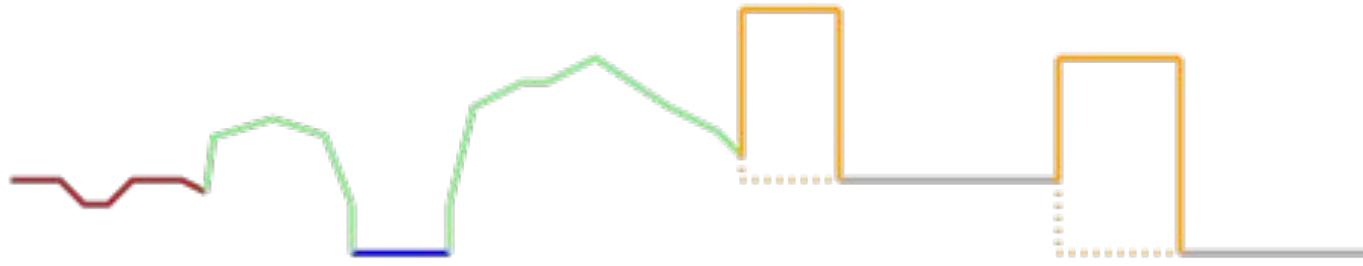
Lifting each polygon based on simple rules (eg avg)



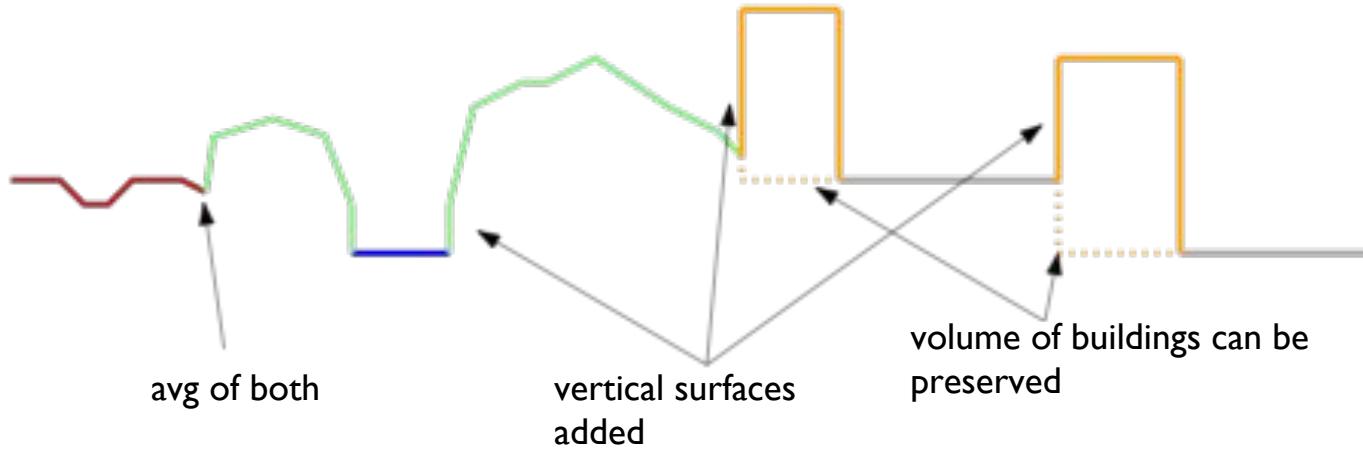
Terrain and vegetation classes have LiDAR points added within the boundary
Other classes only vertices are lifted



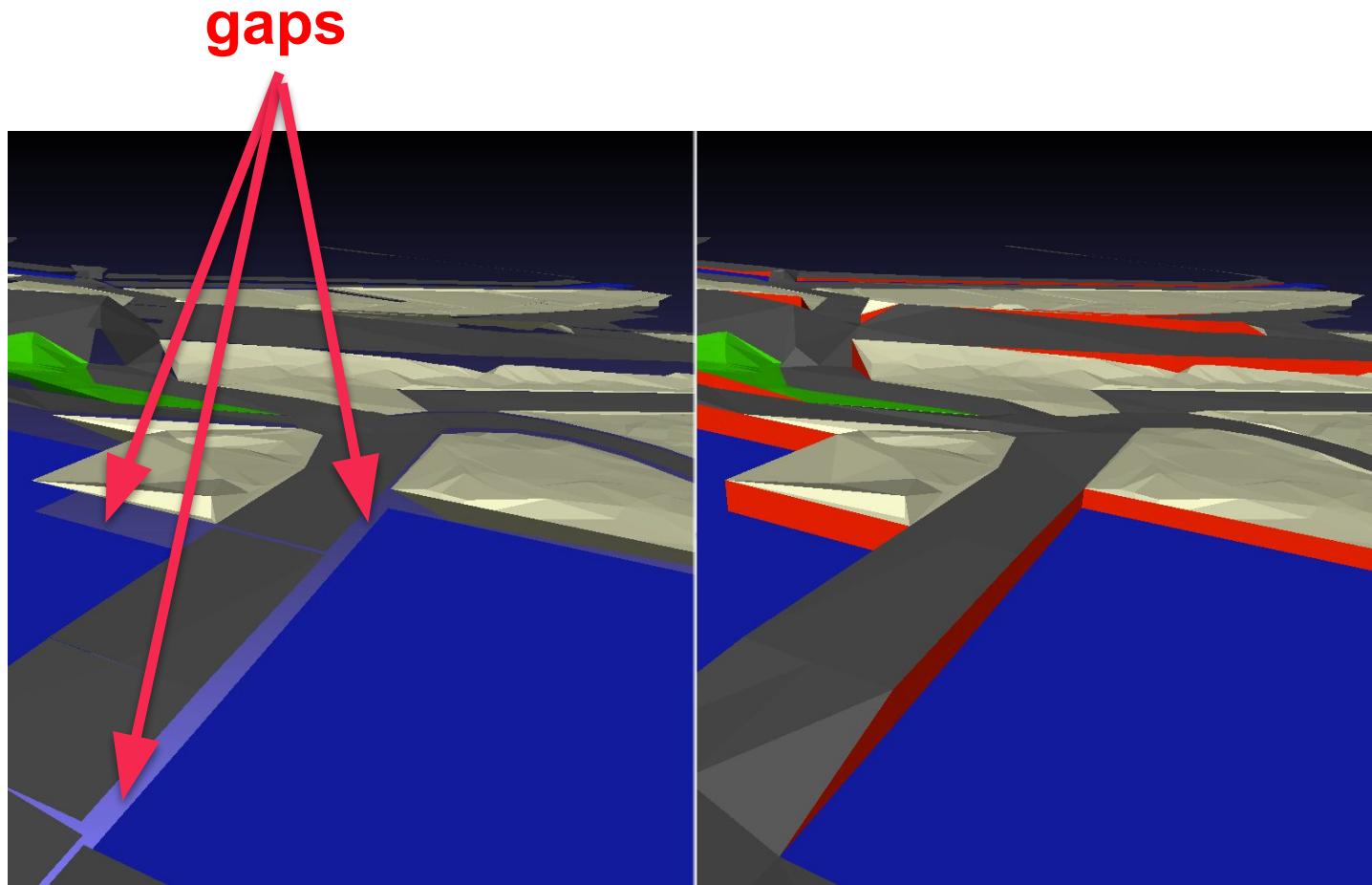
Stitching adjacent polygons with pairwise rules



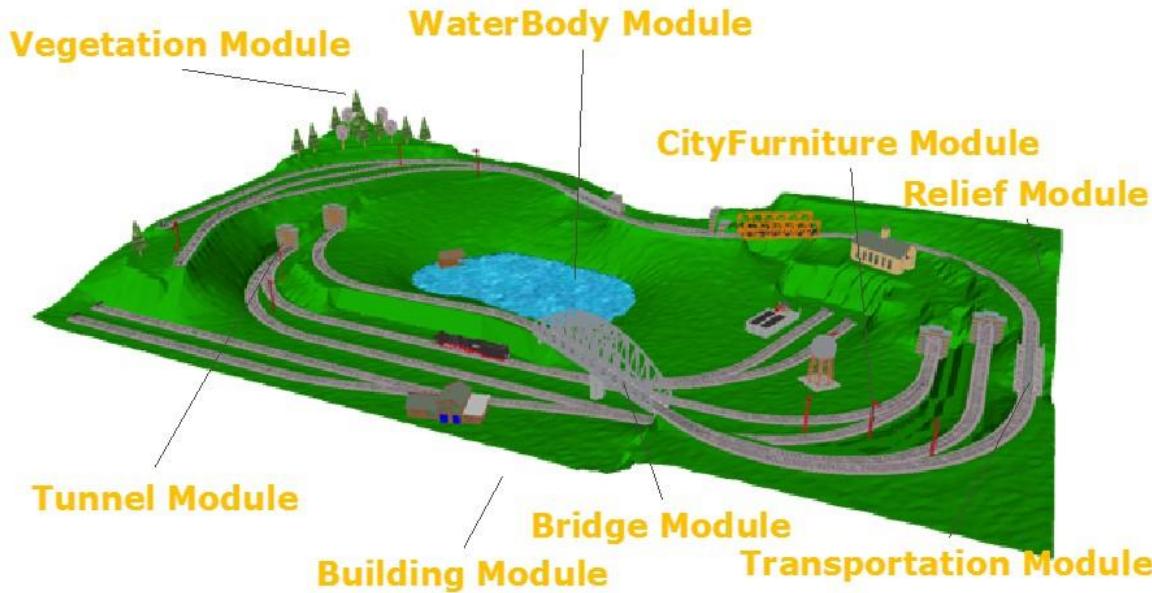
Stitching adjacent polygons with pairwise rules



Stitching adjacent polygons with pairwise rules



Level of Detail (LoD)



Even LoDI 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



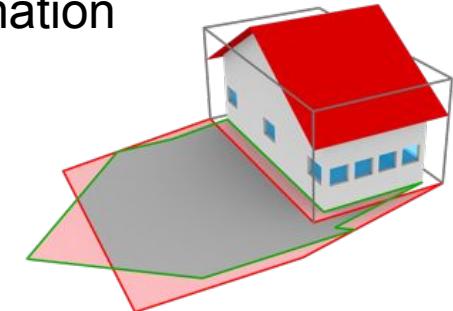
Biljecki, 2016

“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!

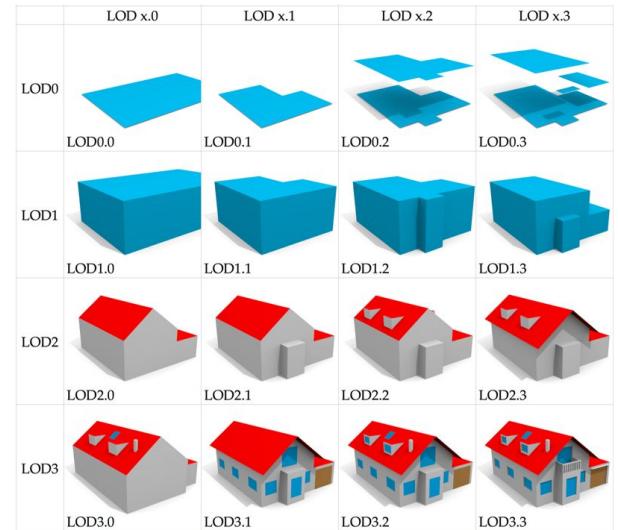
→ Not always strive for highest LoD, relate it to application

LoD

Level of Detail in CityGML

!=

Level of Development in IFC



Current 3D city models

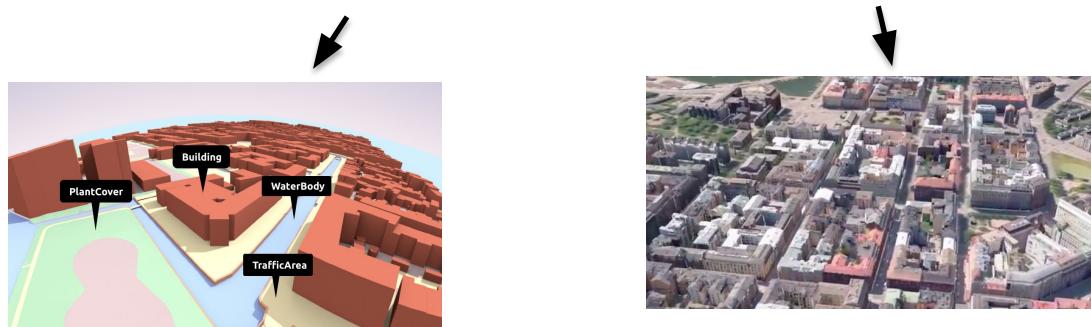
- 3D CMs are rarely generated having needs in mind
- Preparing 3D data for specific applications takes lot of time
- 3D visualisation overvalued; 3D data undervalued
- Errors in 3D CMs; cause errors in outputs



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

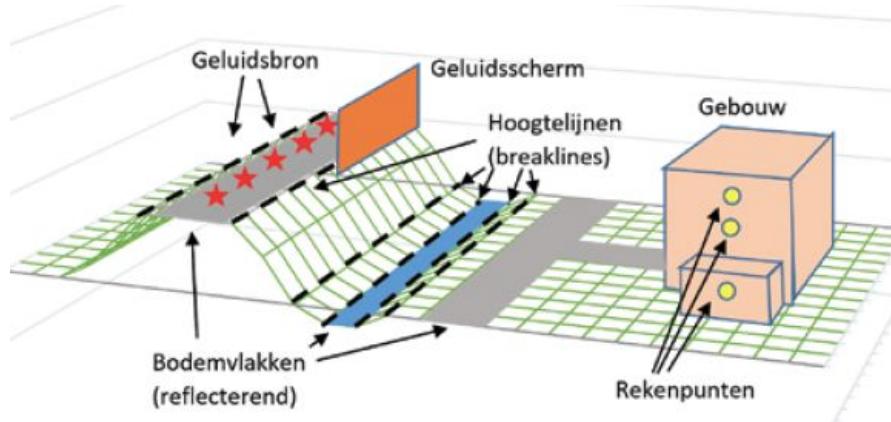
3D city models for urban applications

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

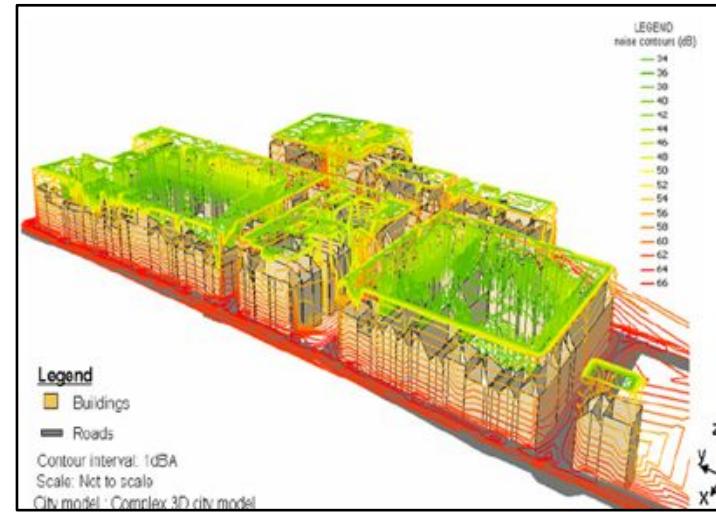


- Application specific Level of Detail
- Up-to-date
 - Not only acquisition: also **maintenance**
- Without errors
- 3D data integration is one of the biggest challenges
 - GIS-BIM

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



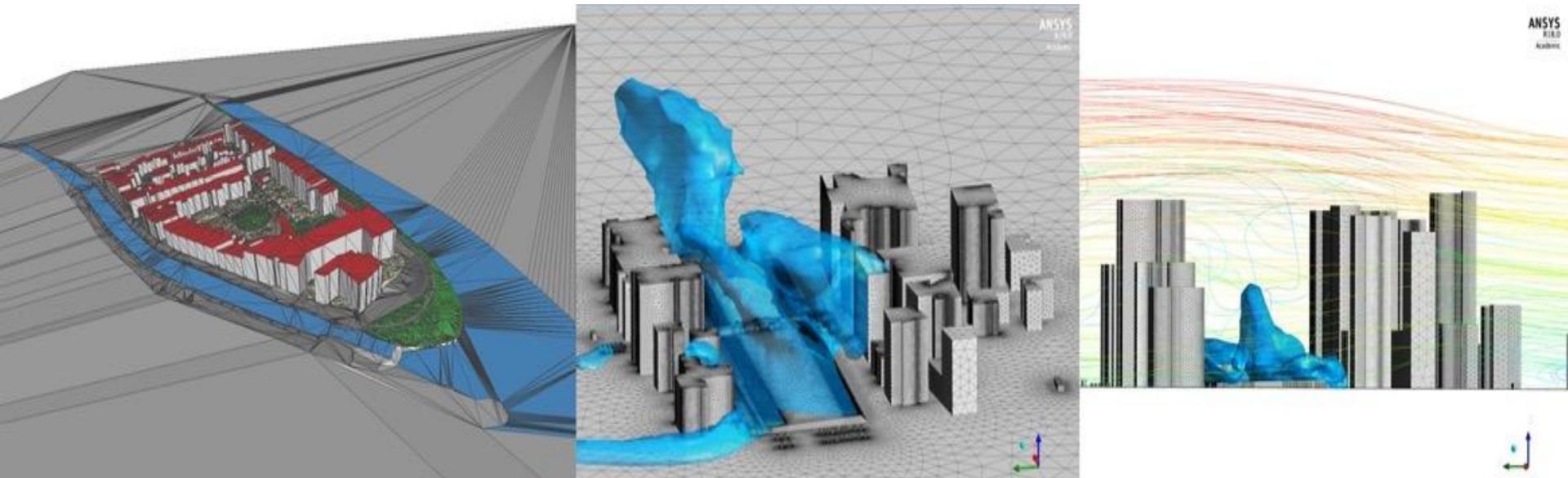
3D CMs for flooding simulations

- highly detailed terrain, with semantics for infiltration and permeability



3D CMs for Simulation - CFD

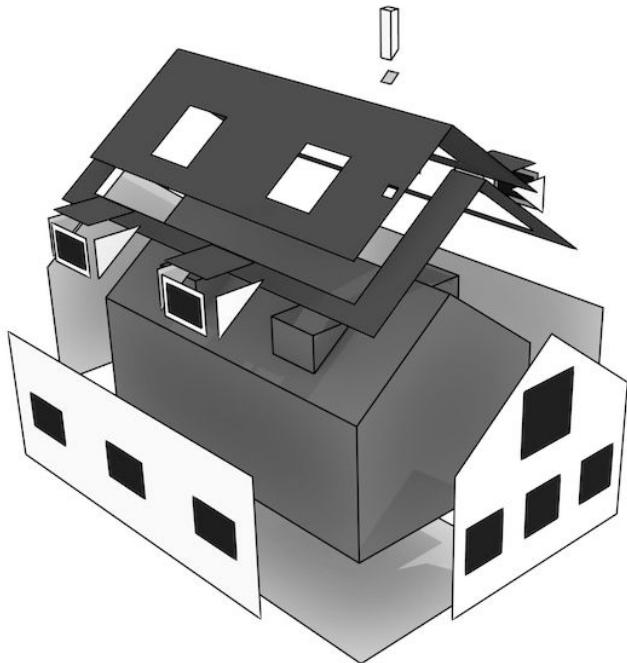
- Computer fluid dynamics modelling (wind, air quality, temperature)
- Requirement: should be 100% closed





CityGML

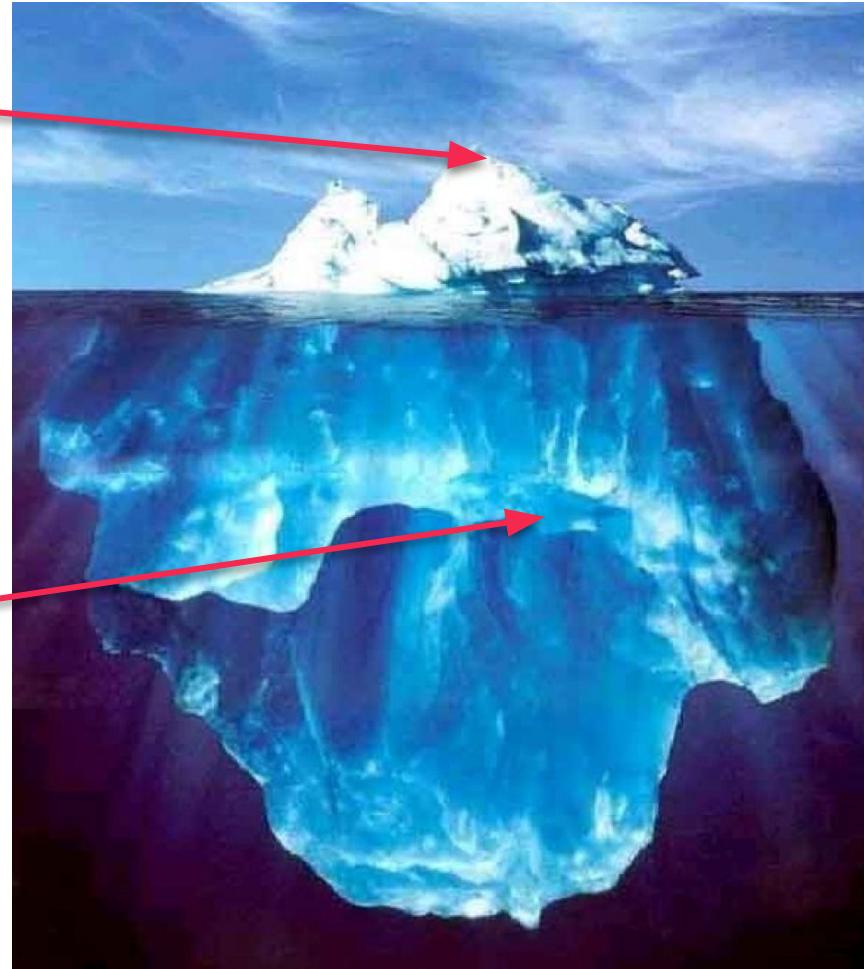
int'l standard (from OGC) for representing and storing 3D city models



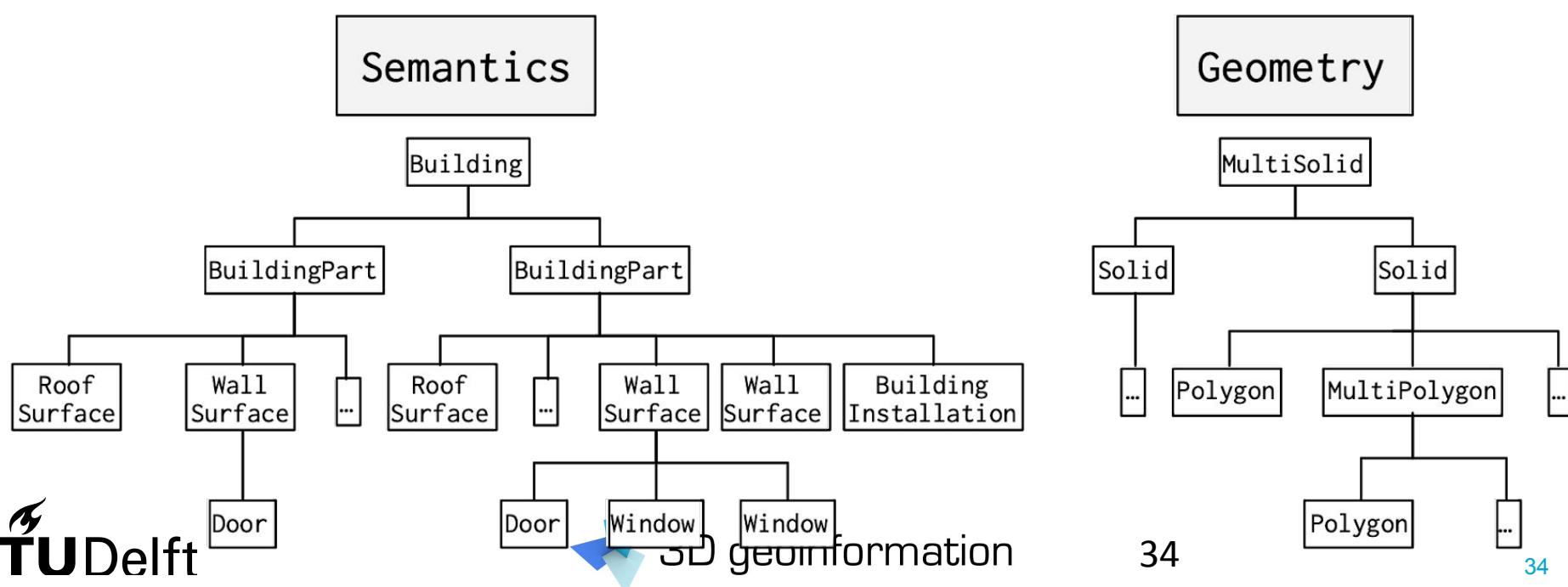
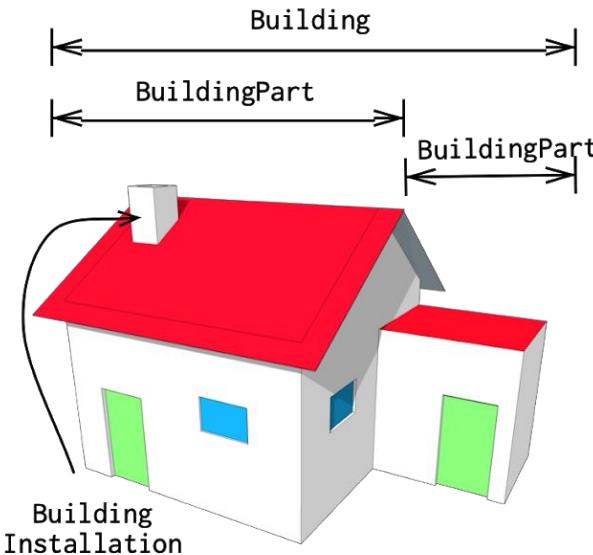
3D city modelling and CityGML == semantics

more than just
visualisation

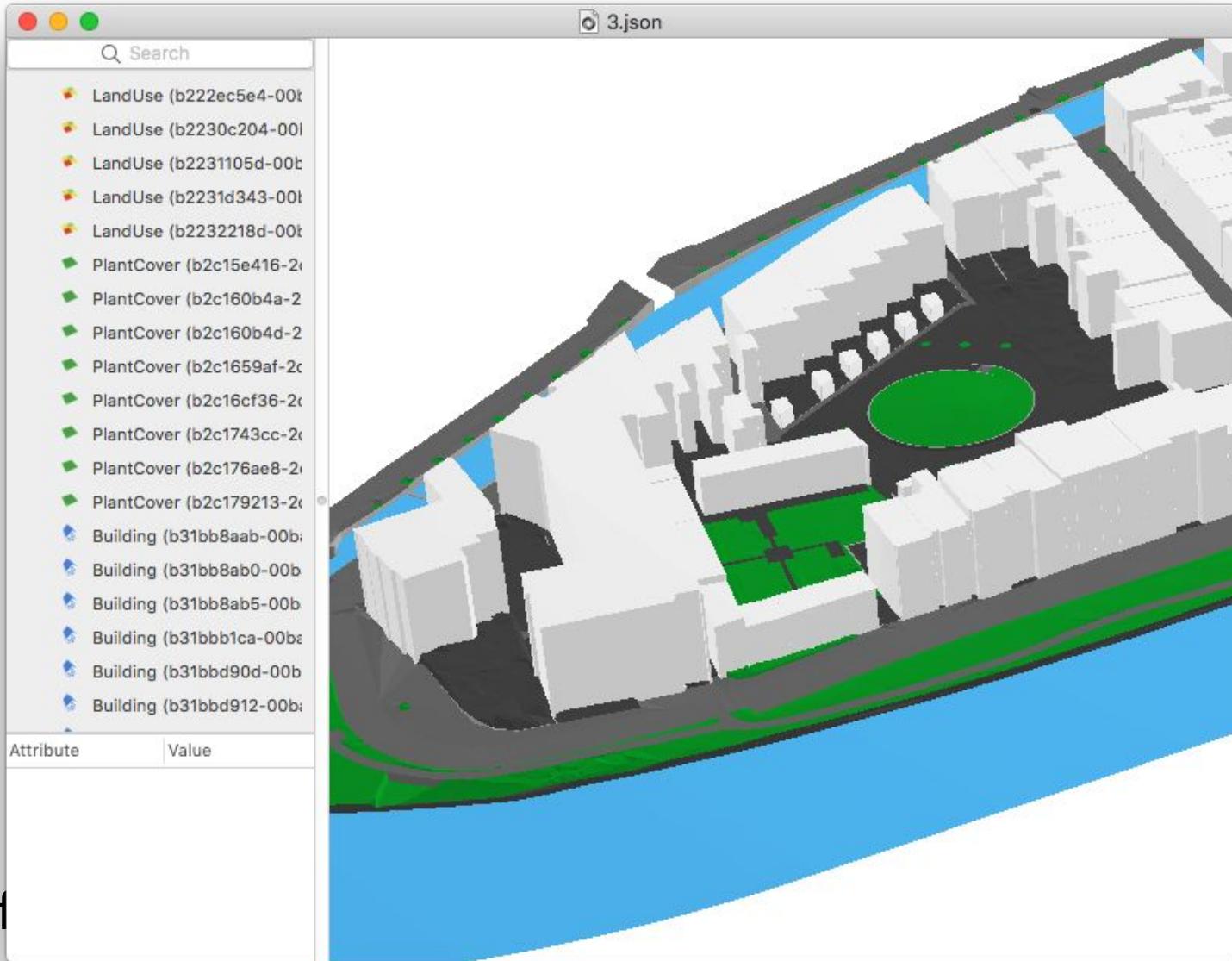
key aspect:
semantic modelling



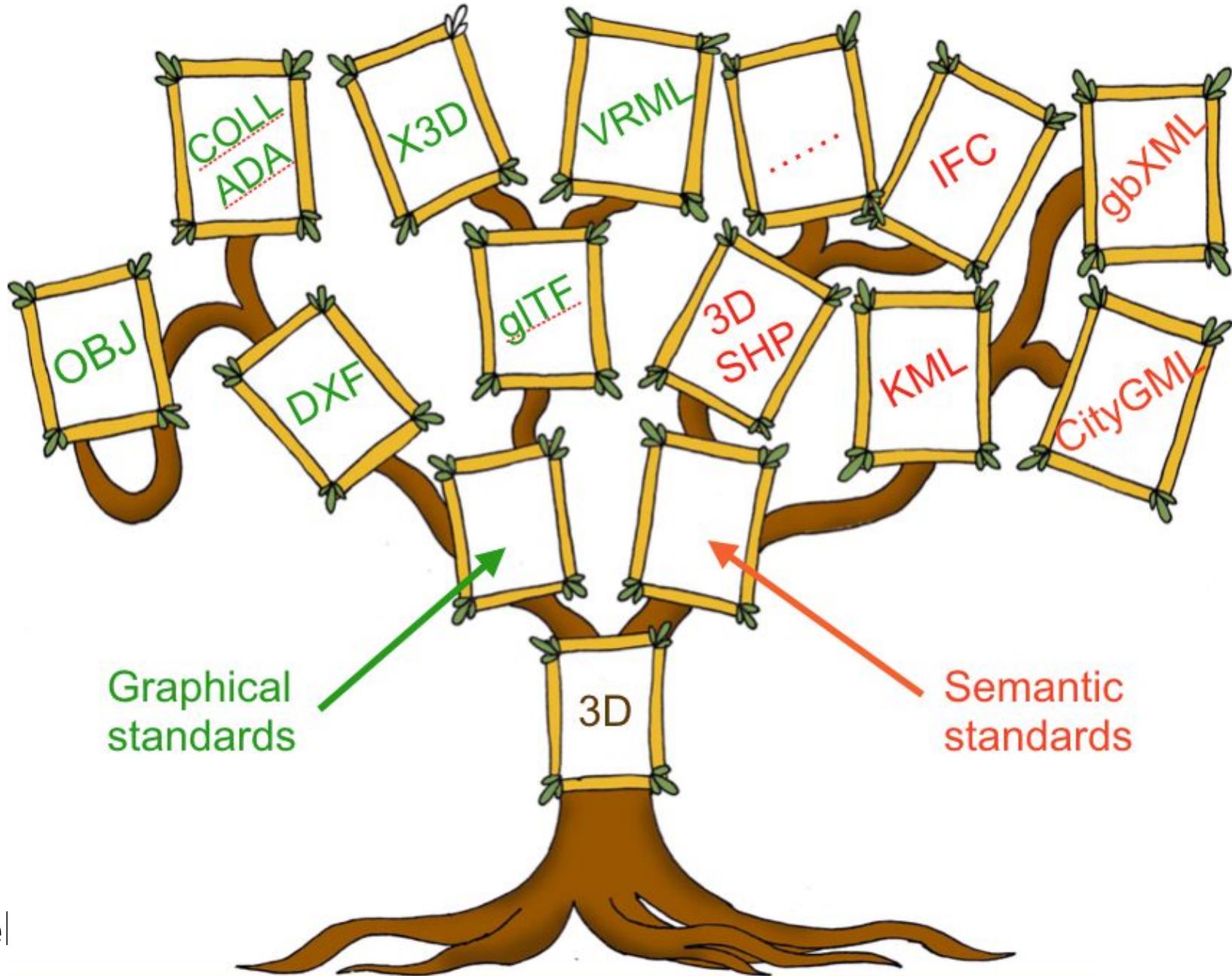
Spatio-semantic coherence



A whole area can be semantically decomposed

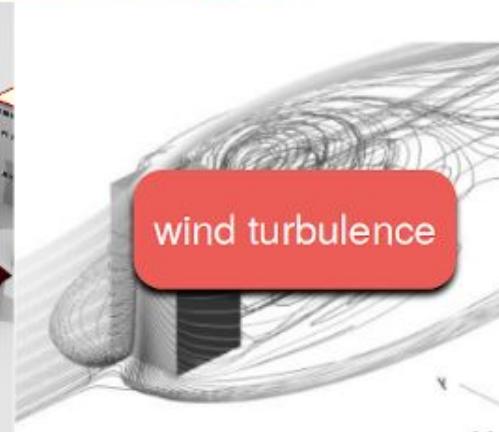
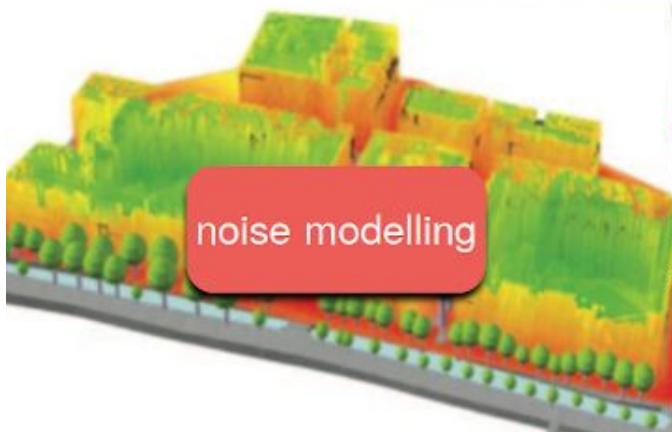
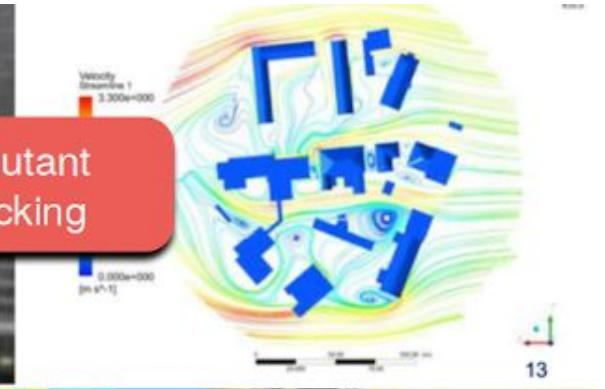
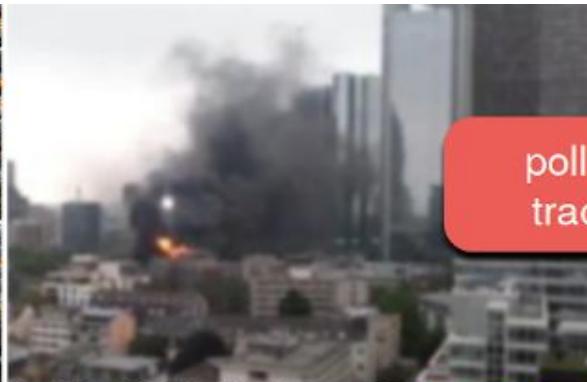


3D formats

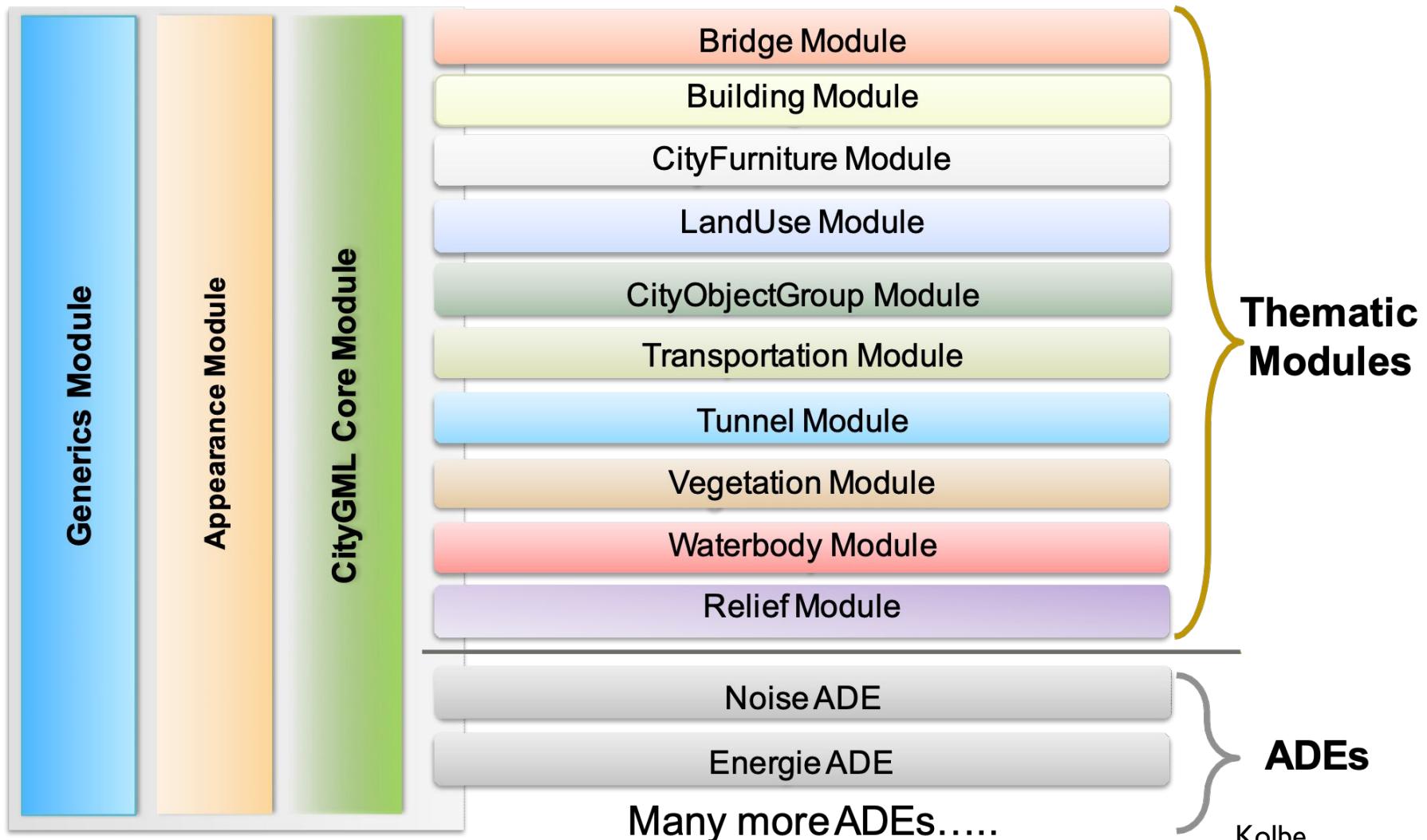


Why semantics are important?

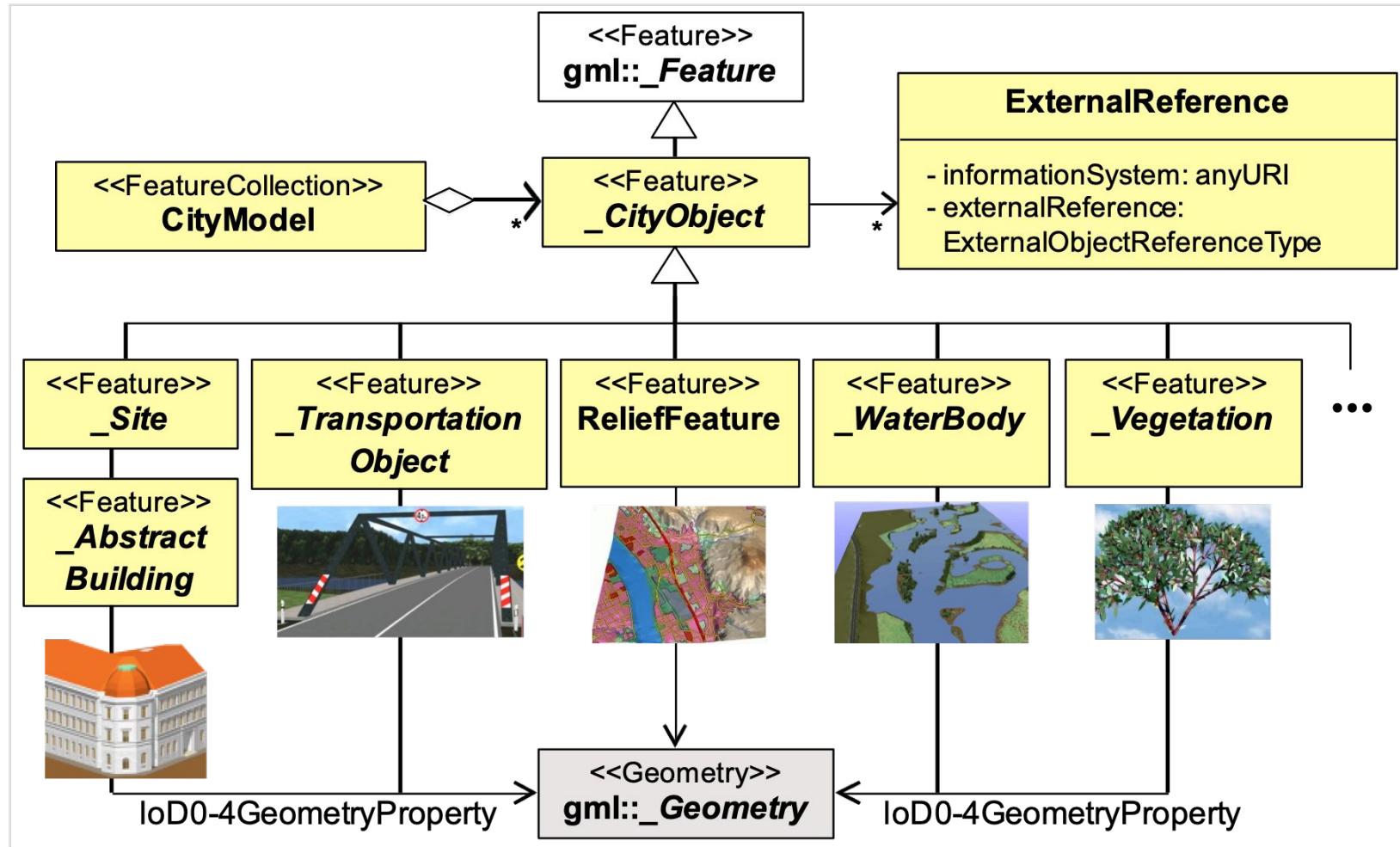




CityGML = standardisation of the data model



CityGML = standardisation of the data model



Gröger et al.
(2012)



CityGML

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>
```

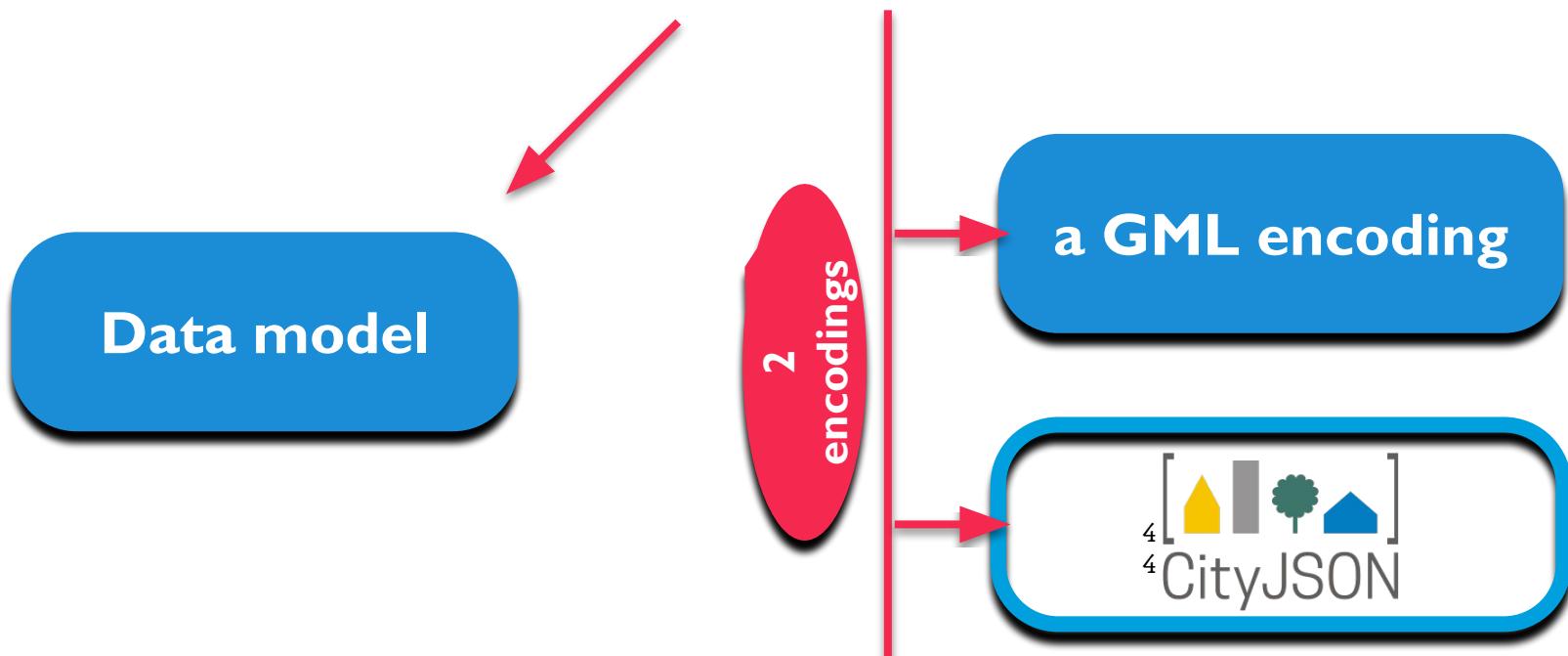
CityGML files are very complex



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



CityGML



CityGML in JSON

cityjson.org

The screenshot shows the homepage of cityjson.org. The left sidebar contains links for Home, What is CityJSON?, News, Specifications, Schemas, Datasets, Extensions, Software, and Help. The main content area features a large logo with a tree, a building, and a road icon inside curly braces, followed by the text "CityJSON". Below it is the subtitle "A JSON-based encoding for 3D city models". There are two buttons: "Getting started" (highlighted in blue) and "Specifications (v1.0.1)". A descriptive paragraph explains that CityJSON is a JSON-based encoding for 3D city models. It highlights its simplicity, compactness, and developer-friendliness, noting support from several software tools and APIs. A list of seven reasons to use CityJSON follows. At the bottom left, there's a link for contributing to the project.

Want to contribute? Spotted an error?

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

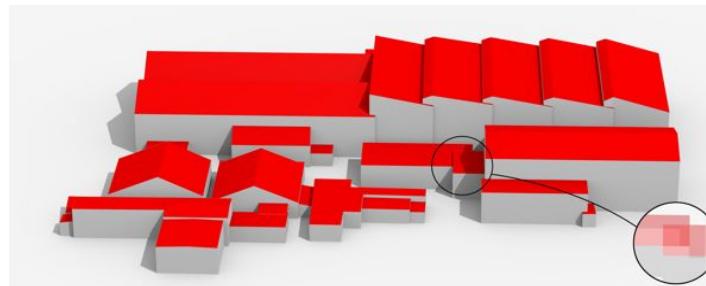
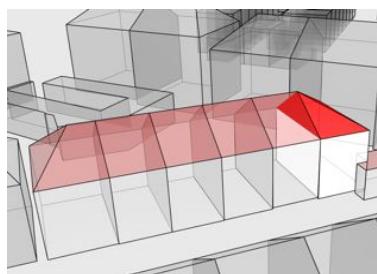
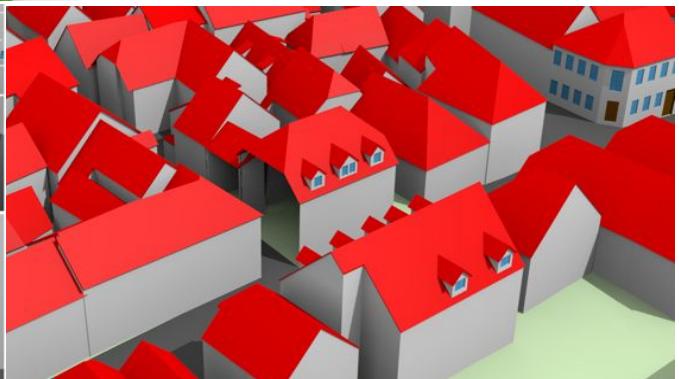
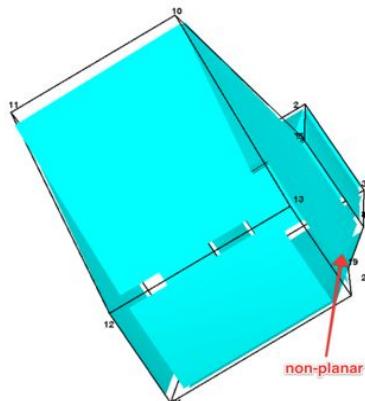
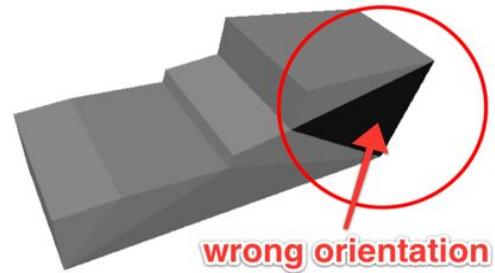
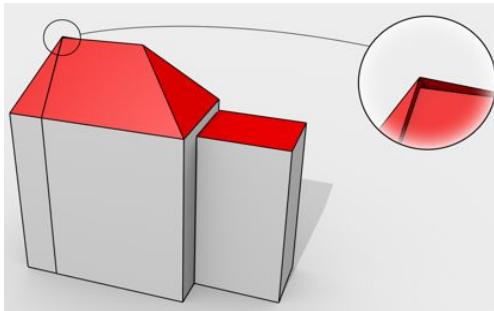
v1.0.1 released

- 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

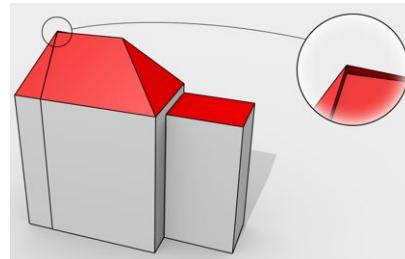
file	CityGML size (original)	CityGML size (w/o spaces)	textures?	CityJSON	CityJSON compressed	compression factor
CityGML demo "GeoRes"	4.3MB	4.1MB	yes	582KB	524KB	8.0
CityGML v2 demo "Railway"	45MB	34MB	yes	4.5MB	4.3MB	8.1
Den Haag "tile 01"	23MB	18MB	no, material	3.1MB	2.9MB	6.2
Montréal VM05	56MB	42MB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



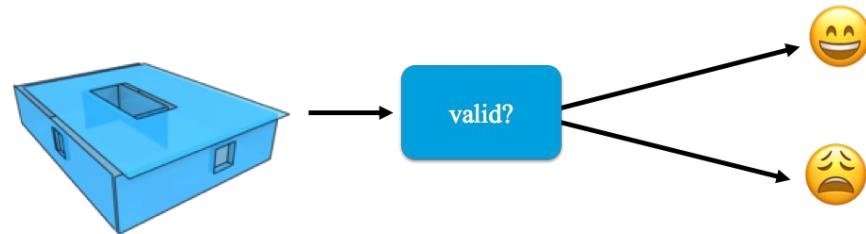
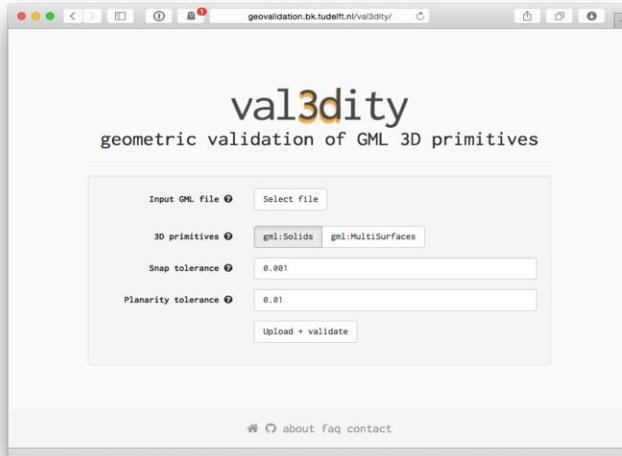
Errors in 3D models

- 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 ...

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

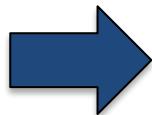
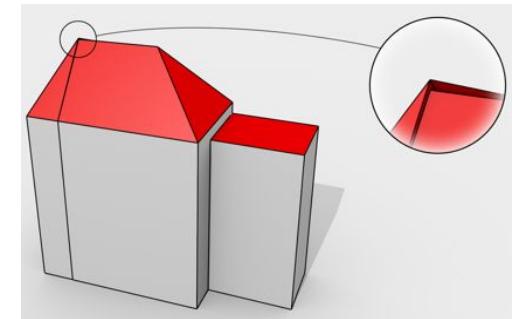


The most common geometric and semantic errors in CityGML datasets

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 LoDI 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

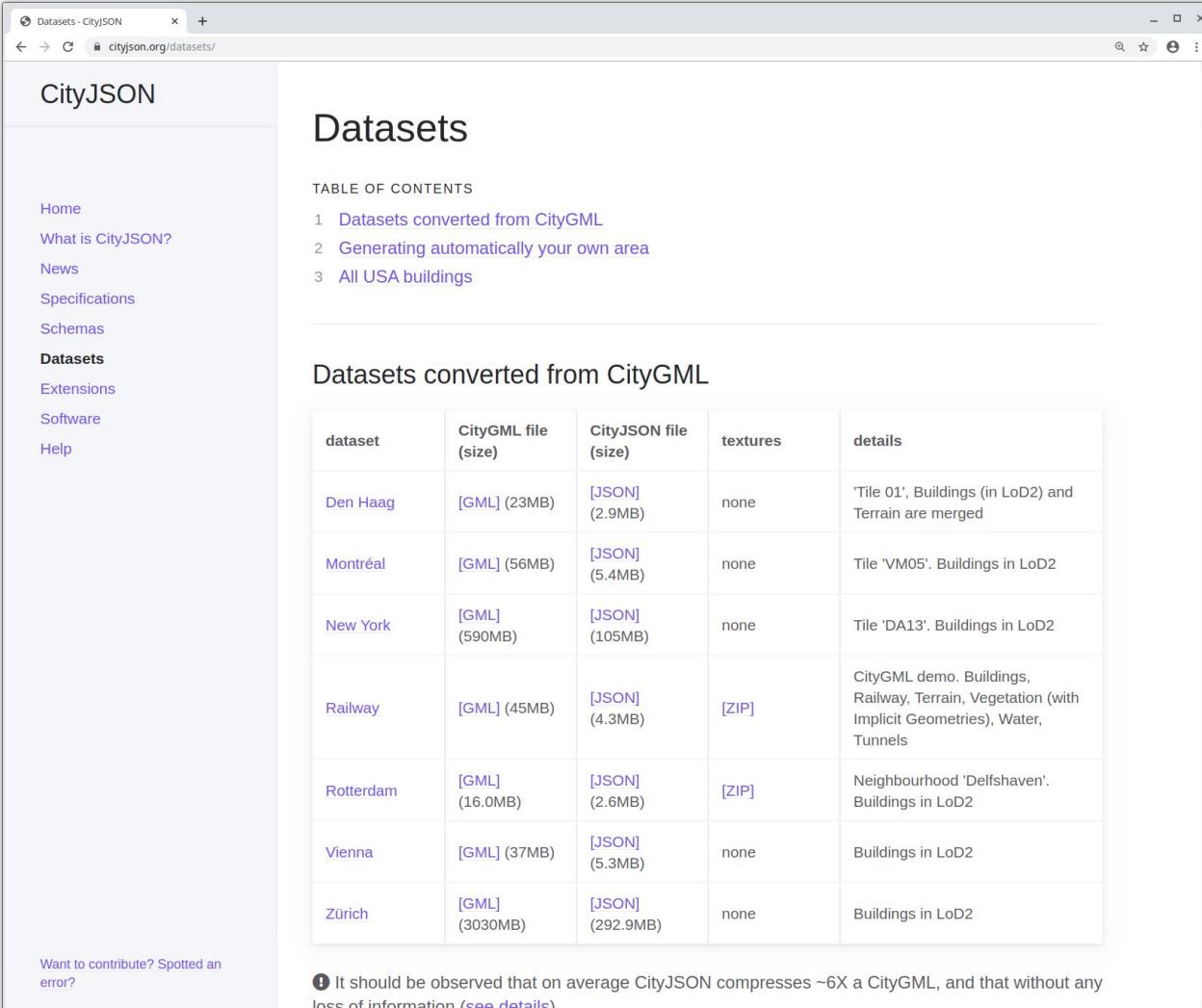
The screenshot shows the 'Open datasets' page of the 3d.bk.tudelft.nl/opendata website. The page features a dark header with the TU Delft logo and '3D geoinformation'. Below the header, there are several sections showcasing different datasets:

- Noise 3D**: Automatically generated 3D input data for noise studies. It includes a thumbnail of a 3D city model with orange buildings.
- 3D city models**: A list of cities around the world with open 3D datasets, available in various formats. It includes a thumbnail of a dense red and brown 3D city model.
- 3D city models in CityJSON**: Several 3D cities are available in CityJSON. It includes a logo with a tree, a bar chart, and a house, labeled 'CityJSON'.
- Dutch 3D city models**: Created with 3dfier, available in OBJ, CityGML, and IMGeo. It includes a logo for '3dfier' with a globe.
- Procedural multi-LoD data**: Created with Random3Dcity, available in OBJ and CityGML. It includes a thumbnail of a 3D city model with colorful buildings.
- 3D BAG**: The 3D BAG is a register of addresses and buildings of the Netherlands, including building height information and related quality metrics. It includes a thumbnail of a 3D city model with blue buildings.

At the bottom of the page, there is a section titled 'Curated lists' with a note about national lidar datasets and a social media footer with icons for Twitter, LinkedIn, and YouTube.

Where to find 3D City Models?

cityjson.org/datasets



The screenshot shows a web browser window titled "Datasets - CityJSON". The URL in the address bar is "cityjson.org/datasets/". The main content area is titled "Datasets" and includes a "TABLE OF CONTENTS" with three items:

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

Below this is a section titled "Datasets converted from CityGML" which contains a table of datasets. The table has columns for "dataset", "CityGML file (size)", "CityJSON file (size)", "textures", and "details".

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

At the bottom left of the page, there is a link: "Want to contribute? Spotted an error?". At the bottom right, there is a note: "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