

Akila Sriramulu, Jochen Wendel, Syed Monjur Murshed and Alexandru Nichersu

European Institute for Energy Research, Karlsruhe Institute of Technology (KIT)
Karlsruhe University of Applied Sciences, Karlsruhe, Germany

Web-based Interface Development for 3D Geospatial Data Visualisation *An open-source and plug-in free approach*

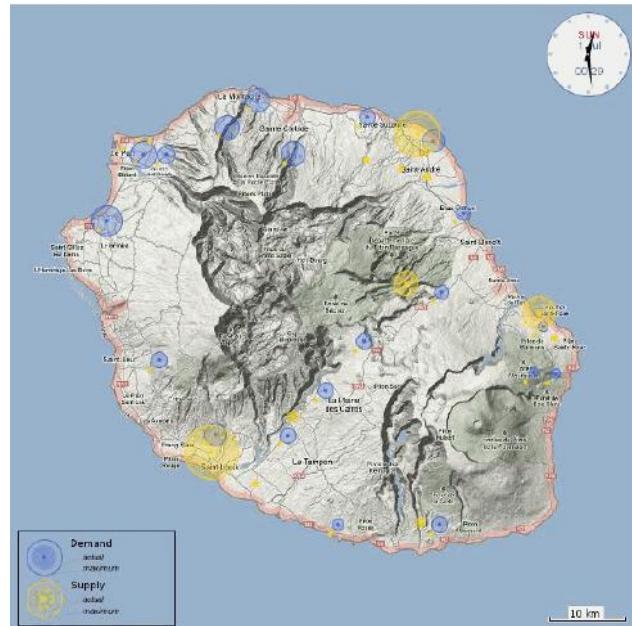
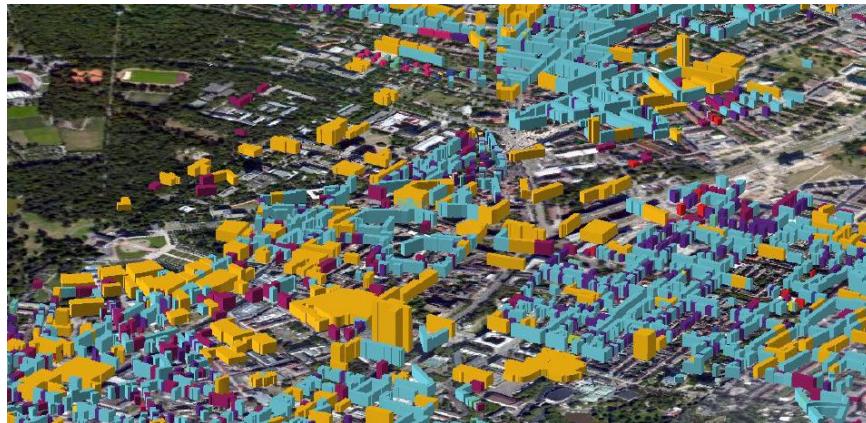




- Visualization needs for energy modelling
- Data requirements
- Current status and limitations
- Visualization interface
- Discussion

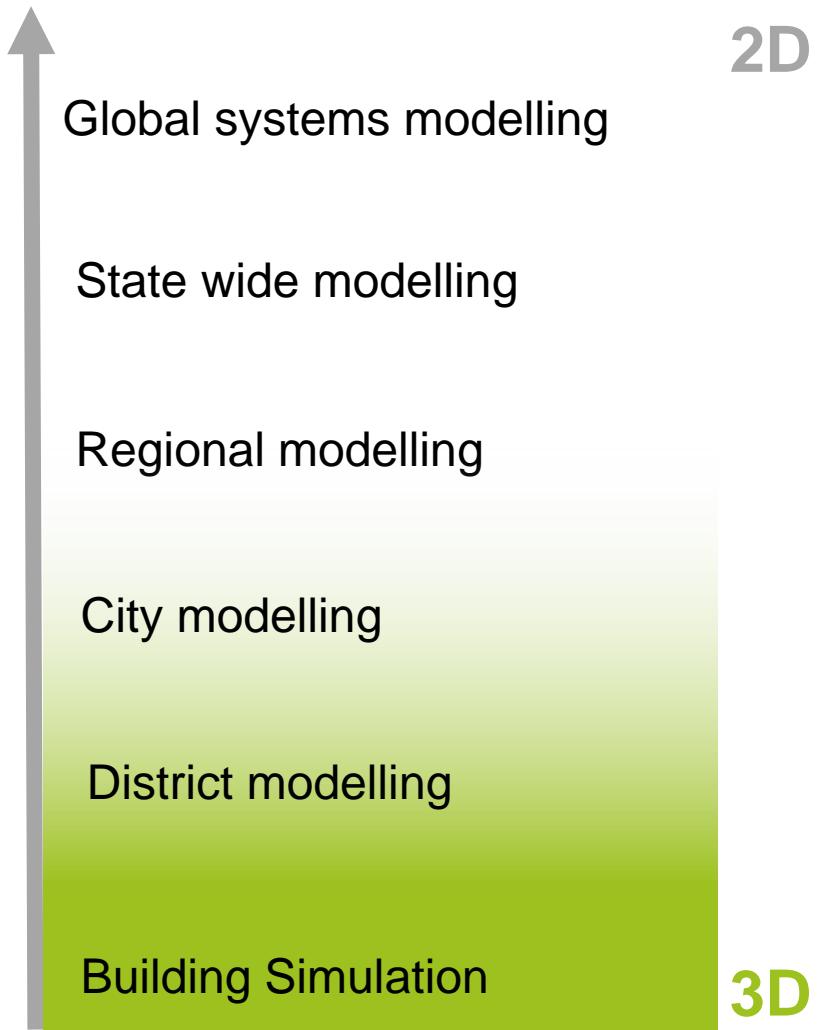
Frameworks for energy modelling

- Simulation modelling
- Systems modelling (e.g. complex systems)
- Energy and environment
- Energy and urban systems
- Building modelling



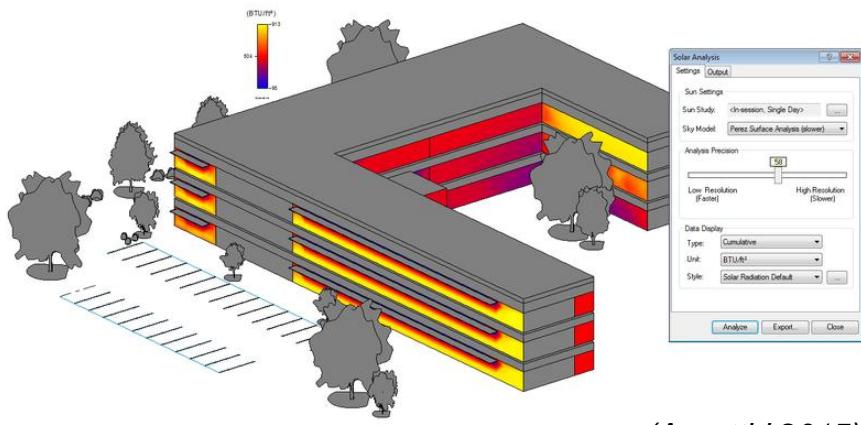
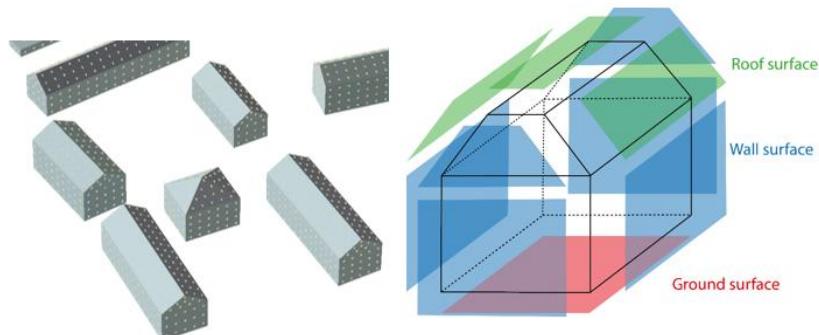


Scale dependence





Scale dependence



2D

Global systems modelling

State wide modelling

Regional modelling

City modelling

District modelling

Building Simulation

3D



Scale dependence



2D

Global systems modelling

State wide modelling

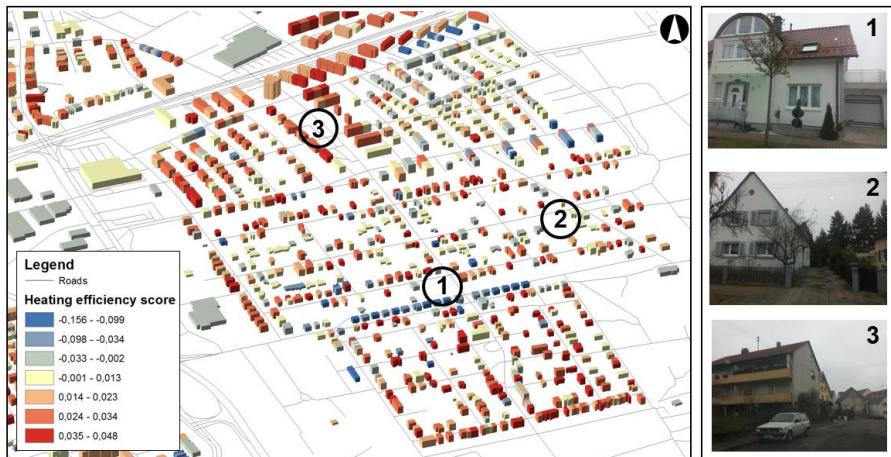
Regional modelling

City modelling

District modelling

Building Simulation

3D

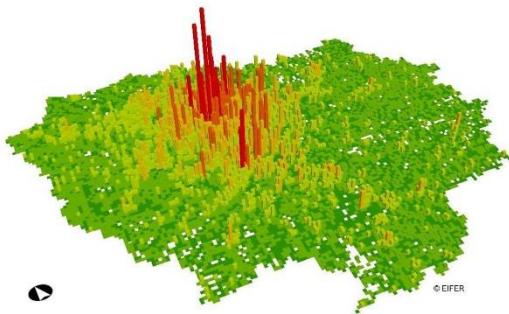


Visualization needs



EIFER

Scale dependence



(Bahu et al. 2012)



2D

Global systems modelling

Regional modelling

City modelling

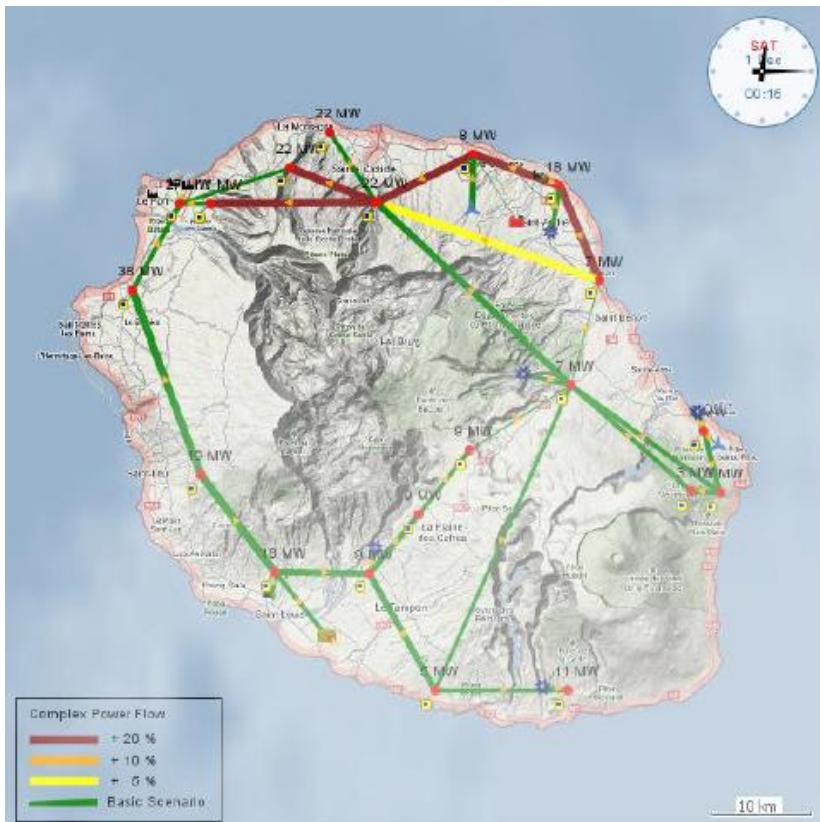
District modelling

Building Simulation

3D



Scale dependence



(Kremers et al. 2013)



Global systems modelling

Regional modelling

City modelling

District modelling

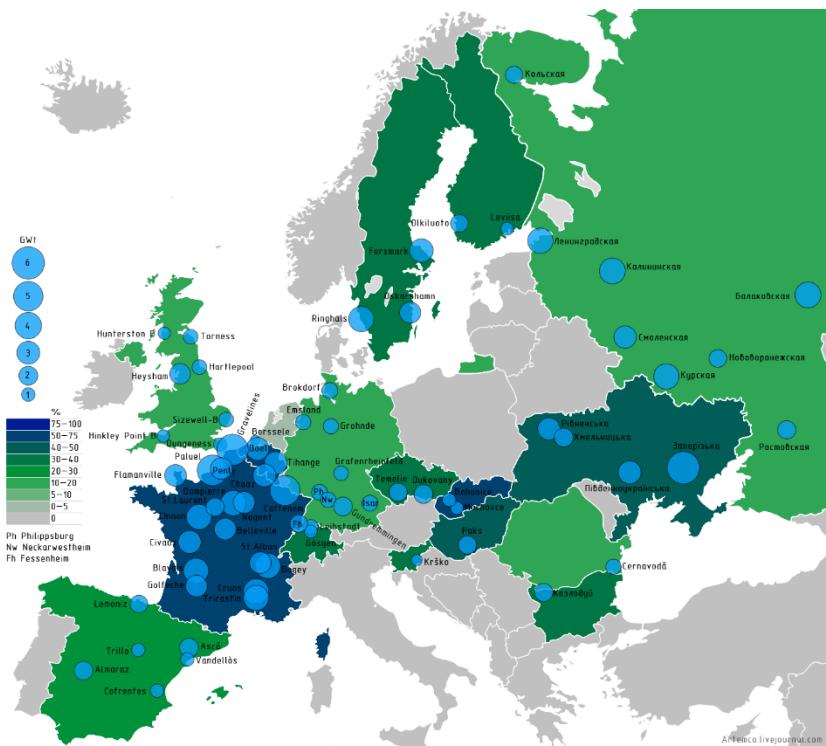
Building Simulation

2D

3D



Scale dependence



Global systems modelling

Regional modelling

City modelling

District modelling

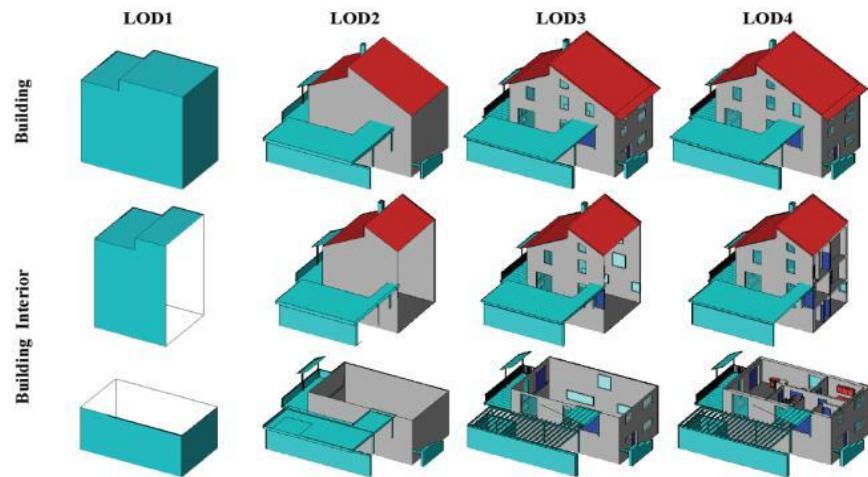
Building Simulation

2D

3D



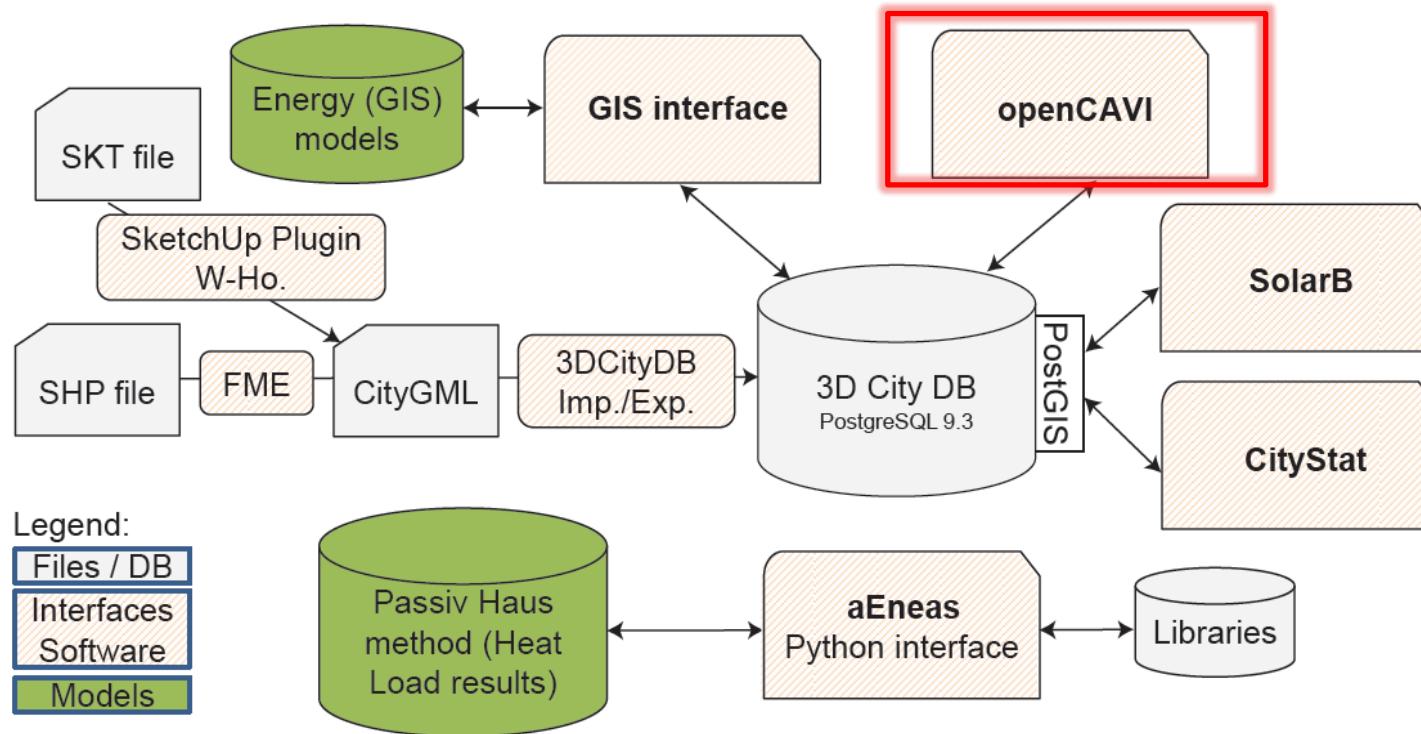
- Visualization at the urban scale
- 3D visualization of buildings
- Storage of semantic building attributes at multiple scales (CityGML)
- Access to results of energy modelling outputs stored in a PostgreSQL
- Open-source (Following OCG standards and guidelines)
- Browser and platform independent and plug-in free (no client side installations)



Source : Gröger et al., (2012)



- Open-source data infrastructure



(Simons and Nichersu 2014; Wieland et al. 2015, Saed and Wendel 2015)



Overview of the state of the art at the start of the project in 2014

- Wide use of 3D browser plug-ins.
 - Google Earth
 - Unity Web Engine
 - Virtual City Systems
- Limited open-source 3D solutions.
 - OSM Buildings, Open3DGlobe, Heidelberg 3D
- Cesium.js was still in an early stage
- Limited documentation on 3D visualization using open-source software
- Limitations in WebGL standard adoption by web browsers
- ESRI lacking in 3D



Evolution of Web Mapping

3D

- Internet
- Xerox PARC Map Viewer (scale, projection)

- GRASSLinks (Information Retrieval)
- TIGER Map server (toggle layers)
- MapQuest (Routing)

- HTML, JavaScript, ActiveX
- Server based
- ArcIMS
- GeoMedia
- Web Standards (WFS, WMS, WCS)

- 3D NASA World Wind
- Google Earth

- AJAX
- Google, OpenStreetMap, Yahoo, Bing maps (Microsoft Virtual Earth)

- GoogleEarth Plug-in
- X3D (VRML)
- Unity

- WebGL
- OSM3D

- W3DS web standard
- Cesium.js
- Three.js
- ESRI ArcGIS Online 3D
- ESRI City Engine
- QGIS2three.js
- OSM Buildings

2D

1989 – 94

1995 – 99

2000-03

2004-05

2006-09

2009-11

2012-present



Steps to generate the 3D city model for the city of Karlsruhe

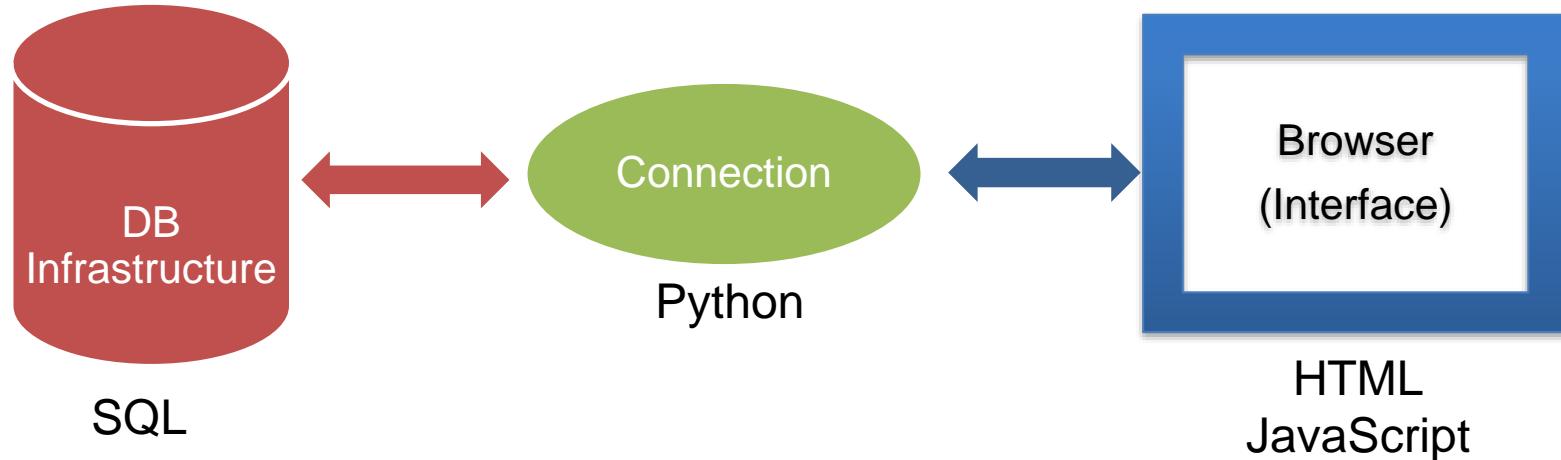
- OpenStreetMap (OSM)
 - Building footprints (Polygons)
 - Building types
 - Roads (Lines)
 - Natural features (Vegetation, Water - Polygons)
- Custom generation
 - Height (Number of floors), SRTM data
 - Attributes (Socio-demographic (Census, INFAS) and energy model outputs)
 - CityGML LoD1 (total of 80k buildings)





Technologies used

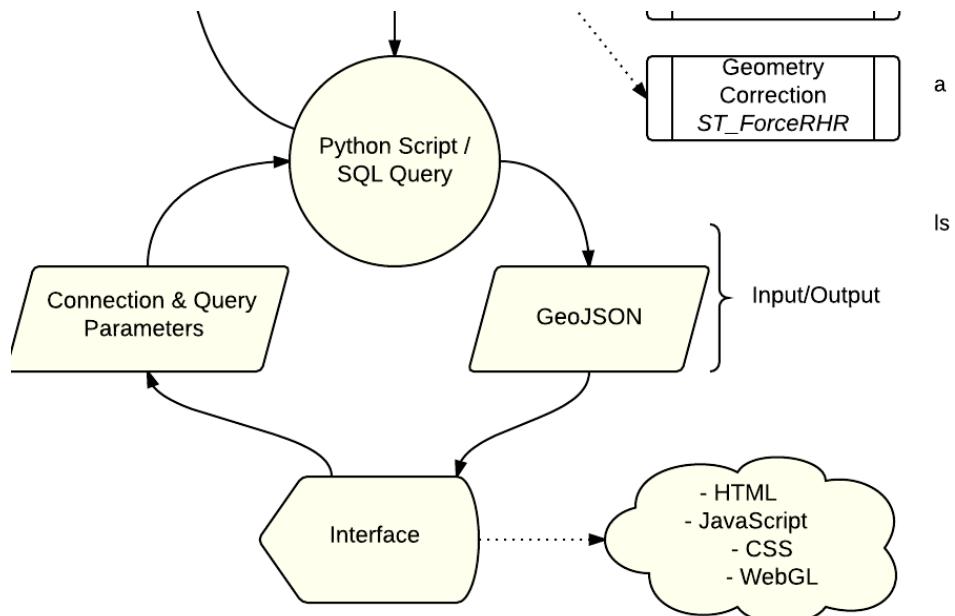
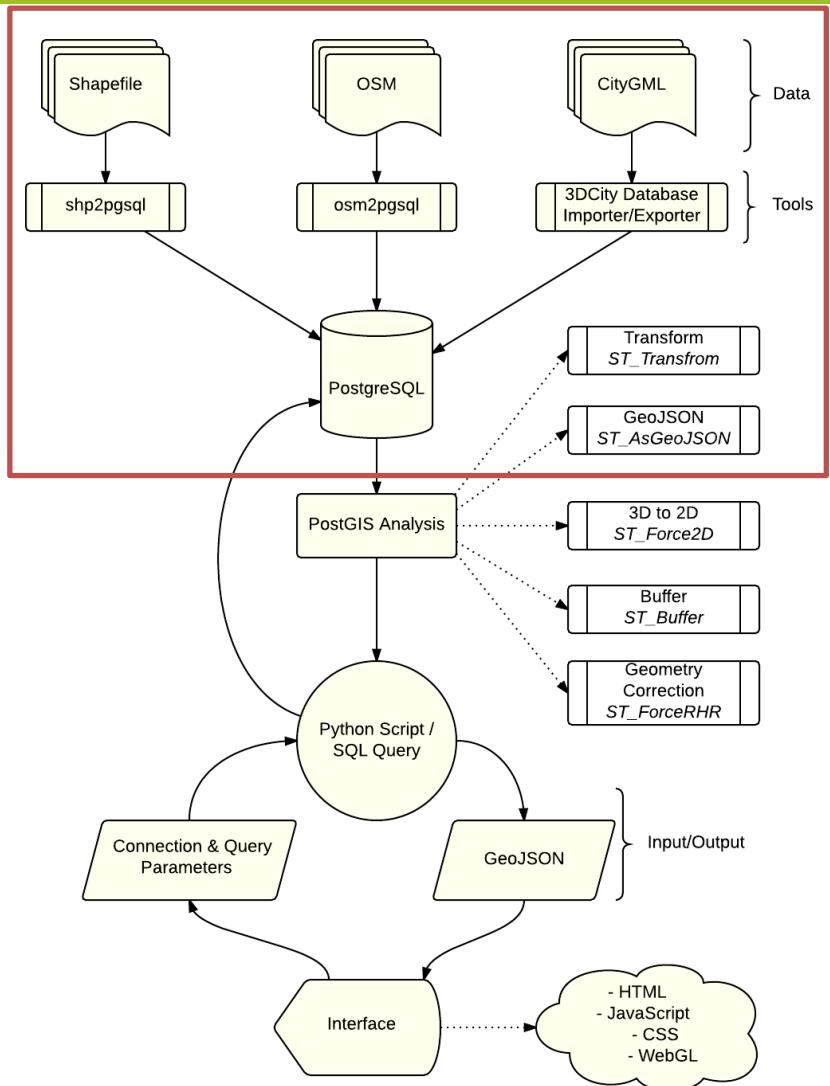
- Programming Languages
 - HTML – Web page
 - JavaScript – Control
 - Python – Connection
 - SQL – Database Control
- JavaScript Libraries
 - ExtJs
 - Three.js
 - D3.js
 - d3-threeD.js
- Data Formats
 - GeoJSON
 - Scalable Vector Graphics (SVG)
 - Shapefile
 - CityGML



Methodology - Workflow

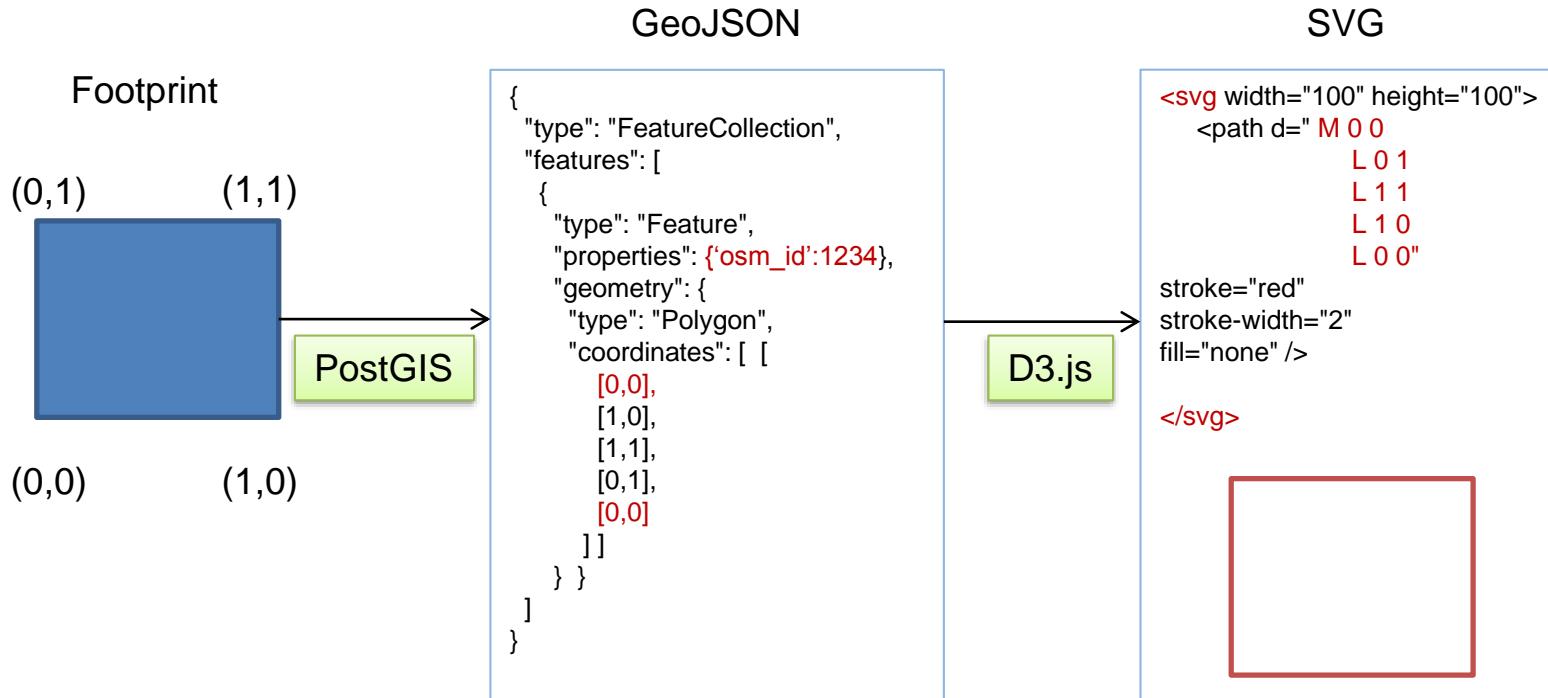


EIFER





Polygons – Buildings & Natural features





Polygons – Buildings & Natural features

SVG

```
<svg width="100" height="100">  
  <path d=" M 0 0  
         L 0 1  
         L 1 1  
         L 1 0  
         L 0 0"  
    stroke="red"  
    stroke-width="2"  
    fill="none" />  
  
</svg>
```



3D Object

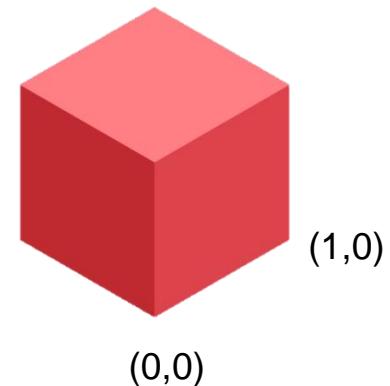


d3-threeD.js

Extruded 3D Object

Three.js

(0,1)

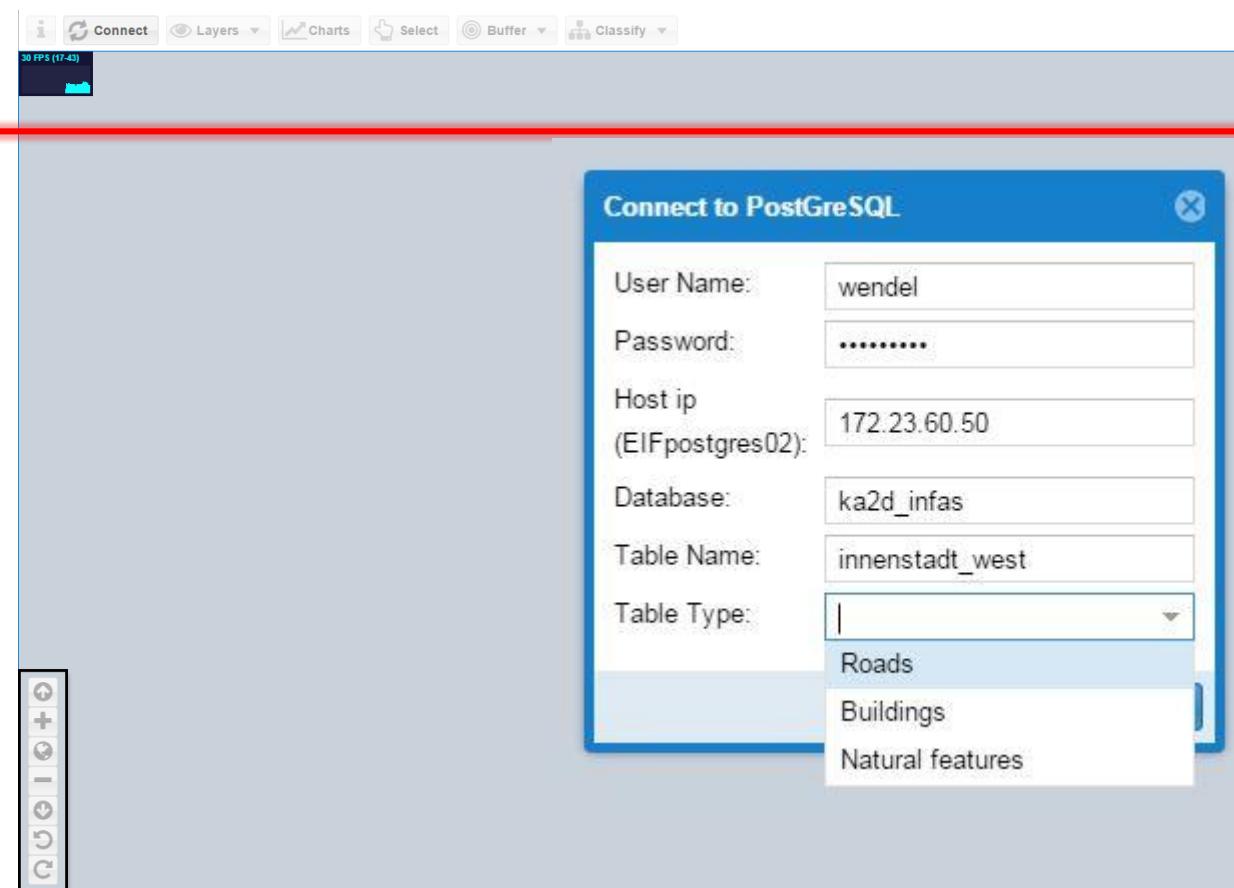


(0,0)

(1,0)

$$\begin{aligned} \text{Extrusion Value} &= \text{Number of floors} \times \text{arbitrary value} \\ &= \text{Number of floors} \times 10 \end{aligned}$$

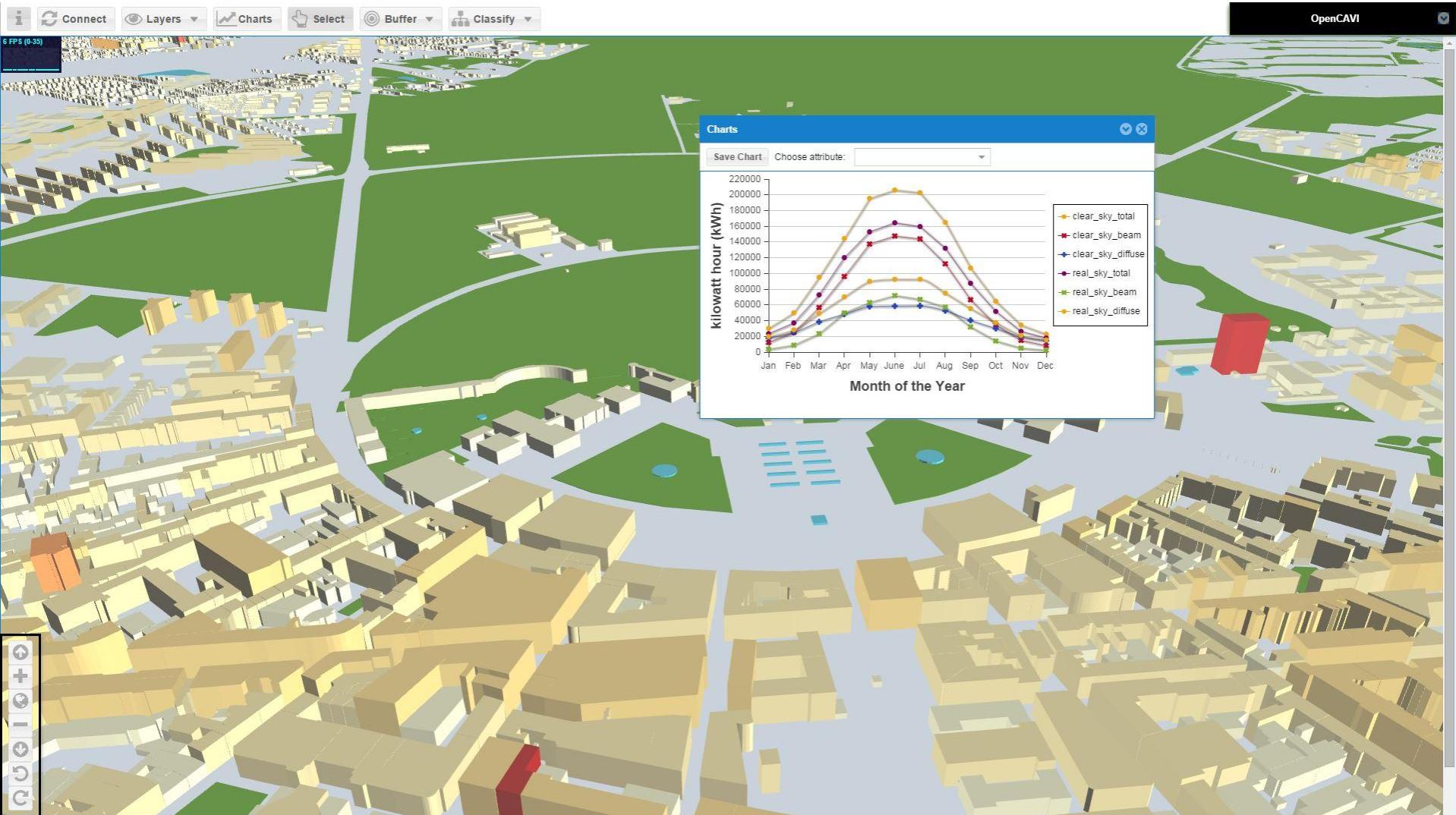
Visualization interface



Visualization interface

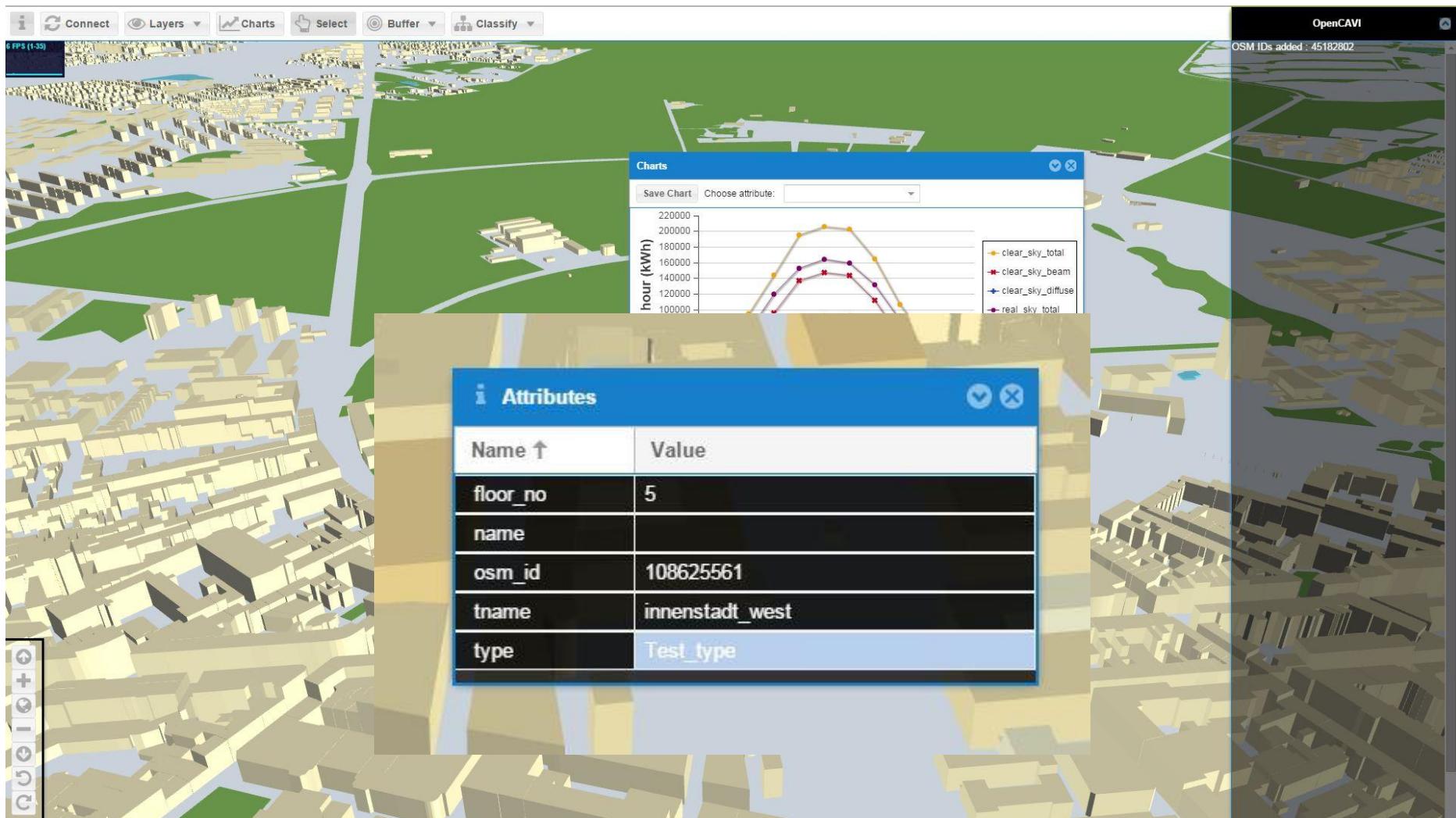


EIFER



Visualization interface

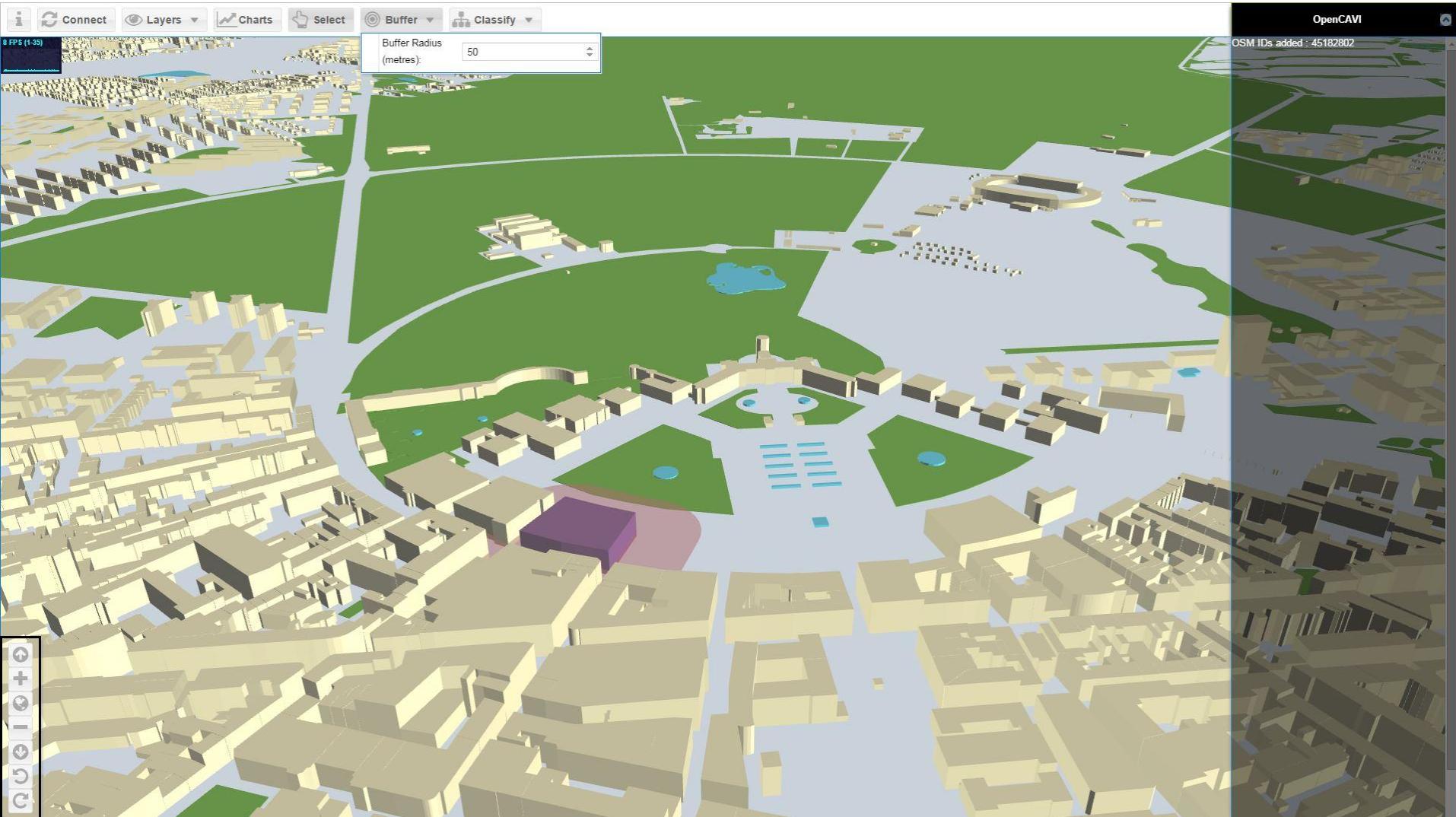
EIFER



Visualization interface



EIFER





Performance benchmarking

~1800 building objects		~14000 building objects		
<i>Browser</i>	<i>Memory</i>	<i>FPS</i>	<i>Memory</i>	<i>FPS</i>
Chrome	~200 MB	35-45	~400 MB	7
Firefox	~240 MB	57-60	~940 MB	3
Safari	~273 MB	25-40	~1.73 GB	9-15
Internet Explorer	~350 MB	24-38	~960 MB	6-10

Preferable : Lower Memory & Higher FPS (Frames per Second)

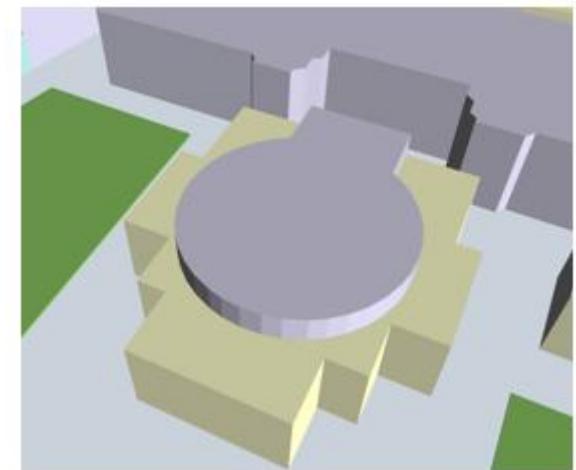
Tested on Intel i5, 8GB RAM, 128SSD, Intel HD Graphics

Conclusion



EIFER

- PostgreSQL and PostGIS enables full control over customization and functionality from different sources
- Limitations of current implementation:
 - Steep learning curve
 - Arbitrary selection of extrusion on origin values in three.js and D3.js conversions
 - Limited cartographic visualization capabilities (own implementations)
 - Currently only works for extruded values
 - Problems with textures
- Current web3D limitations for spatial data
 - Data standardization
 - W3DS (OGC Web 3D Service) – .x3d format = WFS for 2D
- Enhancement potential
 - WMS/WFS via Geosever for base layers





- Rapid advancement of 3D web technology
- Most WebGL 3D applications are heavy on the client side
 - > Streaming of data might be a solution (e.g. gaming industry)
- WebGL standardization process (OGC)
 - > 3D Portrayal Service
- Cesium.js has seen many possibilities (cartographic potential)
- Mobile developments (e.g Globe3D)
- Proprietary plug-in based APIs are not the future

