Creating 3D Tiles

Pirmin Kalberer
@implgeo
Sourcepole, Zurich
www.sourcepole.com

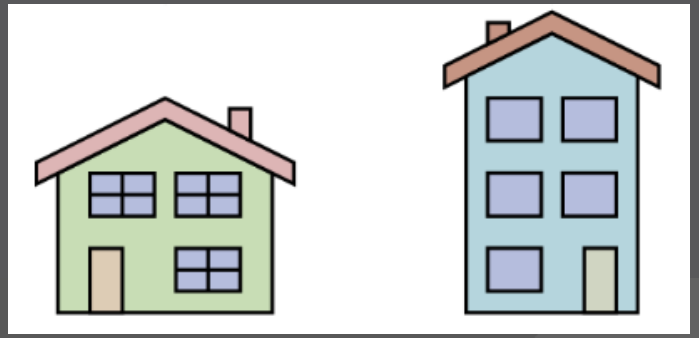
Sourcepole

- **Web GIS**
  - QGIS Webclient (QWC)
  - qgiscloud.com
  - BBOX OGC API & vector tile server
- **QGIS**
  - 4 C++ core developers
  - QGIS server, printing, plugins, ...
  - QGIS Enterprise (maintenance und support)
- **Other OSGeo project contributions**
  - OGR/GDAL, PostGIS, MapServer, Openlayers, OSGeoLive...

3D Tiles

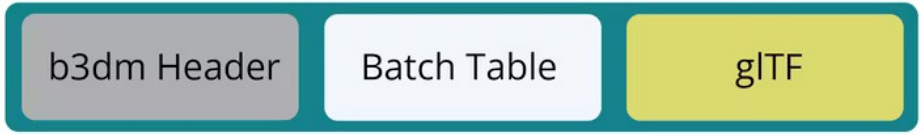
- **OGC 3D Tiles Community Standard**
  - Version 1.1, 12.1.2023 (1.0, 31.1.2019)
- **https://www.ogc.org/standards/3DTiles**
- **https://github.com/CesiumGS/3d-tiles/**

3D Tiles: Batched 3D Model



3D Tiles 1.0

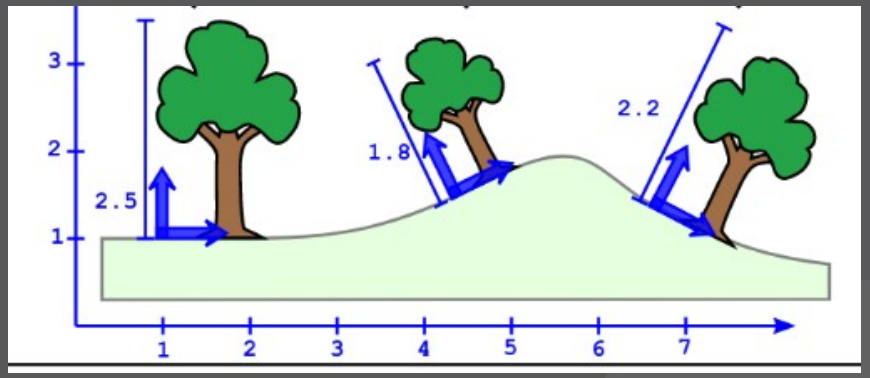
.b3dm file



3D Tiles Next

.glb file + extension

3D Tiles: Instanced 3D Model



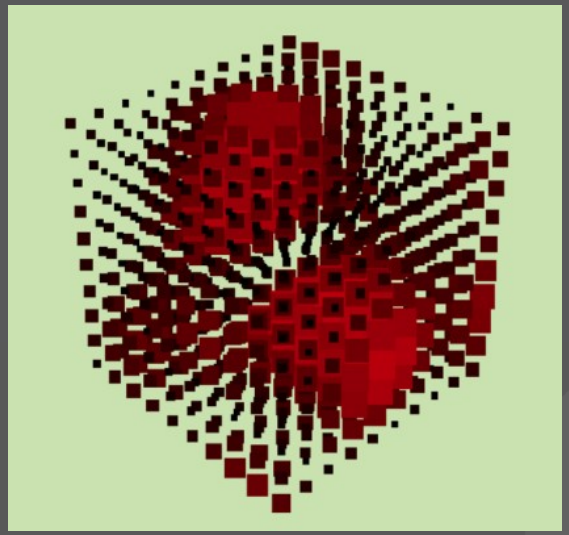
3D Tiles 1.0

.i3dm file

3D Tiles Next

.glb file + extension

3D Tiles: Point Cloud



3D Tiles 1.0

.pnts file

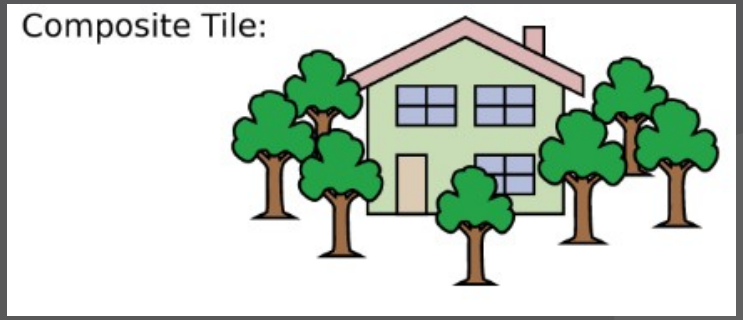
pnts Header | Feature Table | Batch Table

3D Tiles Next

.glb file + extension

glTF | EXT_mesh_features

3D Tiles: Composite Tiles



- glTF (V1.1) or b3dm, i3dm, points

3D Tiles: Declarative Styling

```
{
  "color": {
    "conditions": [
      ["${height} >= 300", "rgba(45, 0, 75, 0.5)"],
      ["${height} >= 200", "rgb(102, 71, 151)"],
      ["${height} >= 100", "rgb(170, 162, 204)"],
      ["${height} >= 50", "rgb(224, 226, 238)"],
      ["${height} >= 25", "rgb(252, 230, 200)"],
      ["${height} >= 10", "rgb(248, 176, 87)"],
      ["${height} >= 5", "rgb(198, 106, 11)"],
      ["true", "rgb(127, 59, 8)"]
    ]
}
```

Viewer (Web)

- **CesiumJS**
- **loaders.gl → deck.gl**
- **3d-tiles-renderer (github)**
  - Three.js based renderer for 3D Tiles
- **Giro3D / Piero (Link)**
  - Framework / application for 3D visualization
- **iTowns (github)**
  - Three.js based JS/WebGL Framework
- **3DCityDB-Web-Map-Client (github)**
  - Cesium based Viewer for CityGML and 3D Tiles
- More: Awesome 3D Tiles

Viewer (Native)

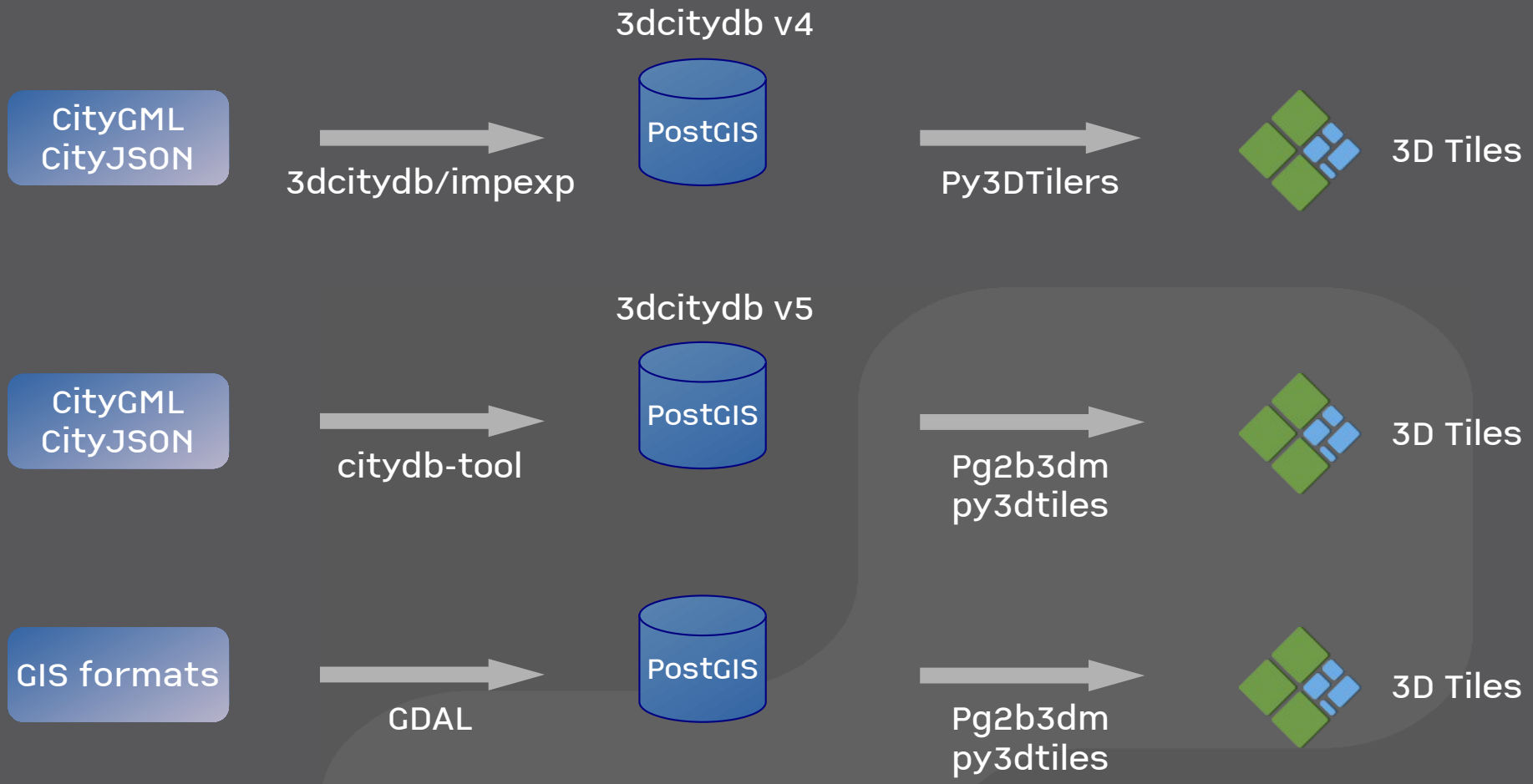
- **Desktop GIS**
  - QGIS
- **Game Engines**
  - Unreal
  - Unity
  - Godot (Windows)
  - O3DE
- **AR/VR**
  - Cesium for Omniverse
  - A-Frame component

Tile creation

Our requirements

- **Use case: Urban planning**
- **Main input formats: CityGML, CityJSON**
- **Tiles in cartesian SRS (e.g. UTM)**
  - Overlay with WMTS (imagery, etc.)
  - No globe (WGS84) required
- **Overlay with imported models (IFC, GPKG)**
- **Property-based styling**
  - CityGML attributes in Batch table
- **Open Source**

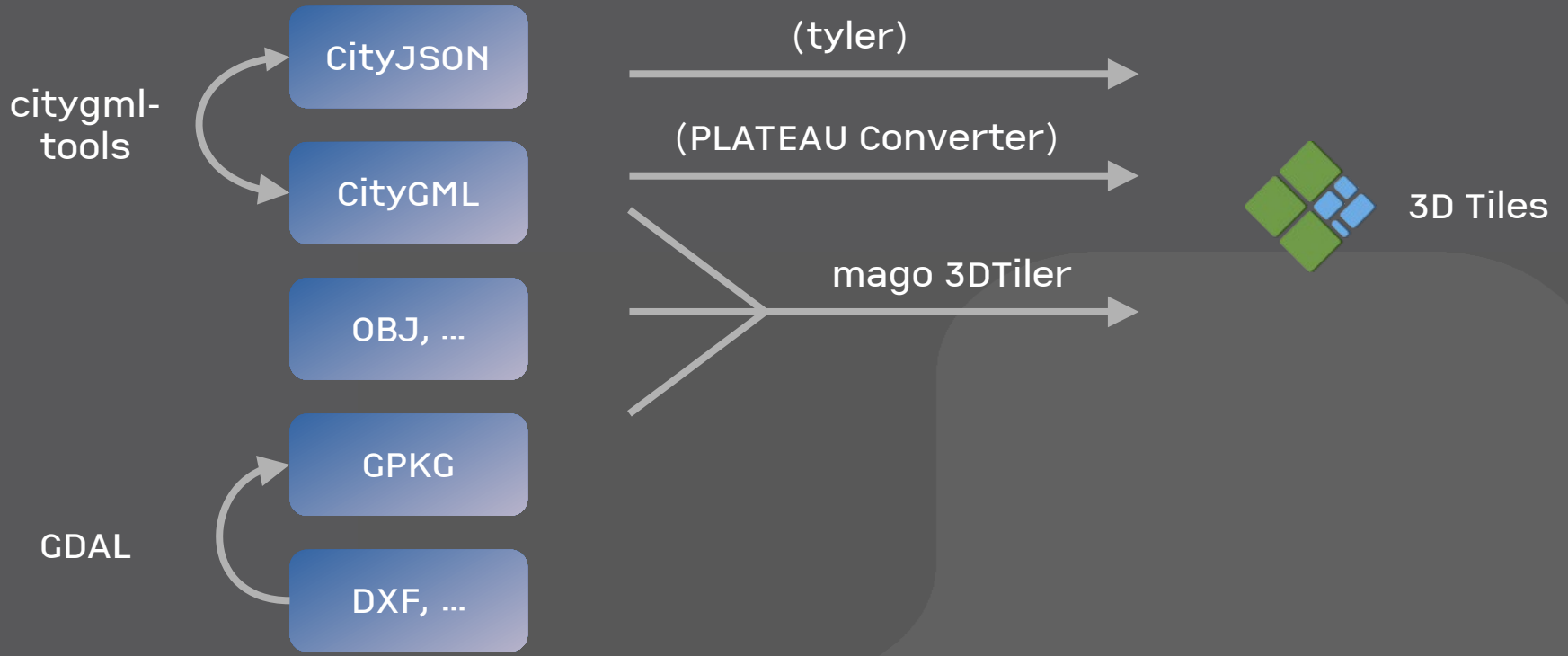
Workflows with PostGIS



Workflows with PostGIS

- **3dCityDB**
  - Schema V4
    - Many tables
    - 3D Tiles with Py3DTilers
  - Schema V5
    - Single table with all 3D geometries (multiple types)
    - May require view for selecting supported types (py3dtiles)
    - 3D Tiles with pg2b3dm or py3dtiles
- **Advantages**
  - Display and edit PostGIS geometries in QGIS
  - Spatial operations in DB after import
    - e.g. join attributes from 2D layers

File based workflows



File based workflows

- **Advantages**
  - Easier for user data imports
  - Less steps → decreased risk of information loss

Py3dTilers

- CityTiler: 3D City Database V4
- IfcTiler
- ObjTiler
- GeojsonTiler
- TilesetReader
- *Currently unmaintained*

py3dtiles

- Python library and command-line tool for generating and managing 3D Tiles datasets
- LAS / XYZ → 3D Point Cloud tiles
- WKB (PostGIS geometries) → b3dm
- PostGIS example:
  - `py3dtiles export -D "host=citydb dbname=citydb user=citydb" -t geometry_data -i id -s out`
- Test case:
  - Requires additional view for 3dCityDB v5
  - Projection ok, objects only partially converted

pg2b3dm

- Tool for converting 3D geometries from PostGIS to 3D Tiles
- Partial support for non-WGS84 tiles
- 3dCityDB export:
  - `pg2b3dm -h citydb -U citydb -d citydb -t geometry_data -c geometry --keep_projection --use_implicit_tiling false`
- Test case:
  - Regeneration of tileset.json with `3d-tiles-tools createTilesetJson` and transform of root node

pg2b3dm

tyler + PLATEAU Converter

- **tyler**
  - CityJson lines → 3D Tiles
  - Test case: No CityJson converter found
- **PLATEAU Converter**
  - Convert CityGML models of Japan into 3D Tiles, MVT and GeoPackage
  - Also part of Re:Earth Flow
  - Test case: GML reader errors

- **Input formats**
  - 3D Modeling: Wavefront obj, Collada, glTF, FBX, ...
  - BIM: IFC-STEP (partial)
  - Point Cloud: PLY, LAS, LAZ
  - City Models: CityGML 3.0 (partial)
  - GIS Vector: shp, GeoJSON, GeoPackage
- **2D to 3D Extrusion**
- **On-The-Fly CRS Conversion**
- **Test case:**
  - WGS84 only, PR for cartesian SRS

mago 3DTiler

Test results

- **CityGML**
  - 3dCityDB v4 with Py3dTilers
  - 3dCityDB v5 with pg2b3dm (custom tileset.json)
- **IFC, GPKG**
  - mago 3DTiler with cartesian PR
- **Potential alternatives:**
  - Py3dtiles
  - mago 3DTiler (also for CityGML)

3D Tiles Community

- **Awesome 3D Tiles**
  - https://github.com/pka/awesome-3d-tiles
  - Viewers
  - Tile creation
  - glTF tools
  - Demo applications & learning material
- **awesome-quantized-mesh-tiles**
  - https://github.com/bertt/awesome-quantized-mesh-tiles
- **Contribute!**
  - Blogs, Tutorials
  - Awesome 3D Tiles Discussions

Thank you!

Pirmin Kalberer
@implgeo