

State of the Map Asia 2018

Bangalore, 17-18 Nov. 2018

Auroville Geomatics Studio: building a FOSS GIS platform



C S R

Auroville Centre for Scientific Research Trust



Auroville

- International, intentional, spiritual **community**
- Founded in 1968, located near Pondicherry
- About 3000 residents, 50 nationalities, 10km²
- One of the goals: build an “ideal” city
- A big challenge for such a small village





Need maps for (almost) everything

- Topography, geography, geology
- Town planning, architecture, infrastructures, administrations, decision support systems
- Cadastral, land ownership, asset management
- Natural and human made features
 - Professionals & domain specialists: water, architects, botanists...
- => 300+ layers

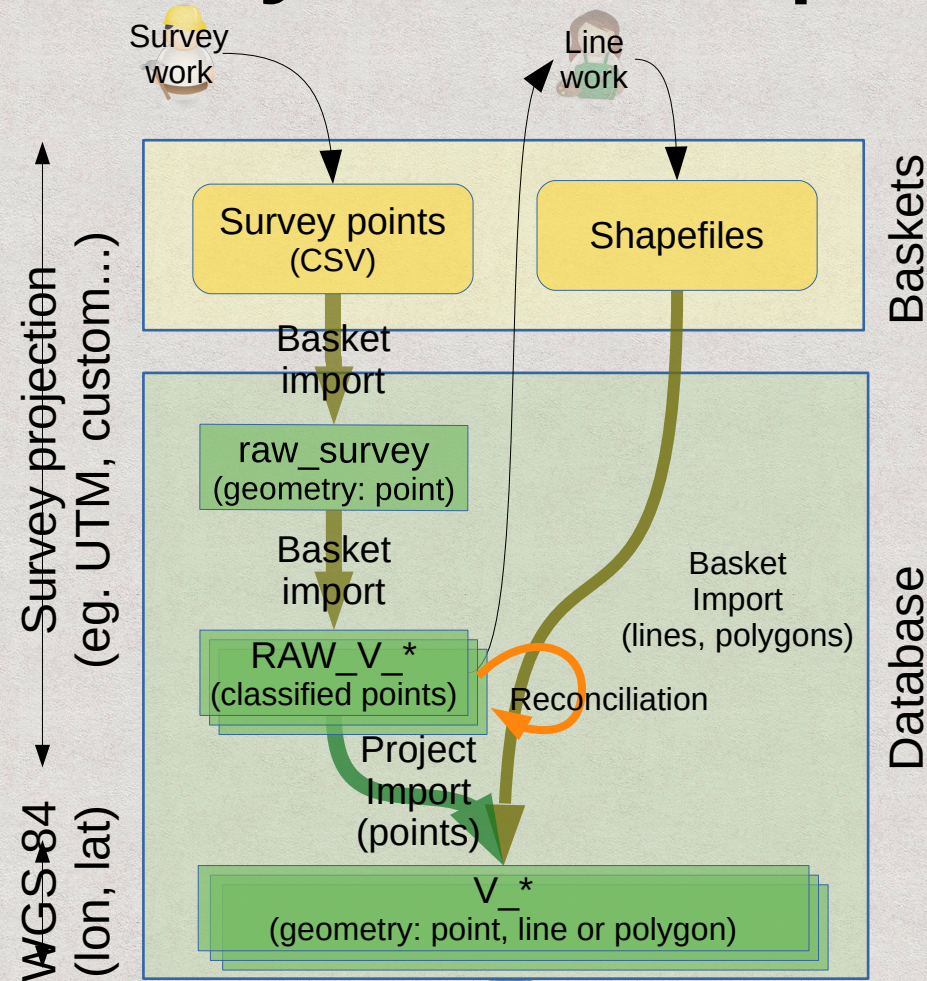


Sources

- Surveys: high accuracy DGNNS, Total Station and more (drones, ?)
- Other sources
 - Legacy in all kind of formats (Shapefiles, DXF...), coordinate systems
 - Online (OSM, aerial raster files, etc)
 - Volunteers with hand held GPS, other Geo-aware tools...
- Related data: beyond GIS
 - Chronological: wells, rain gauges, weather stations readings...
 - Infrastructures, administrations... in all kind of relations



Survey data import



Use case: water management

- Initial funding: Department of Science and Technology of MST, GOI
- Weather data collection (rain...)
- Ground water (well monitoring...)
- Waste water plants
- Pipes network, flow analysis*
- => Goal: *Water resource management (budget, plan)*

* Planned



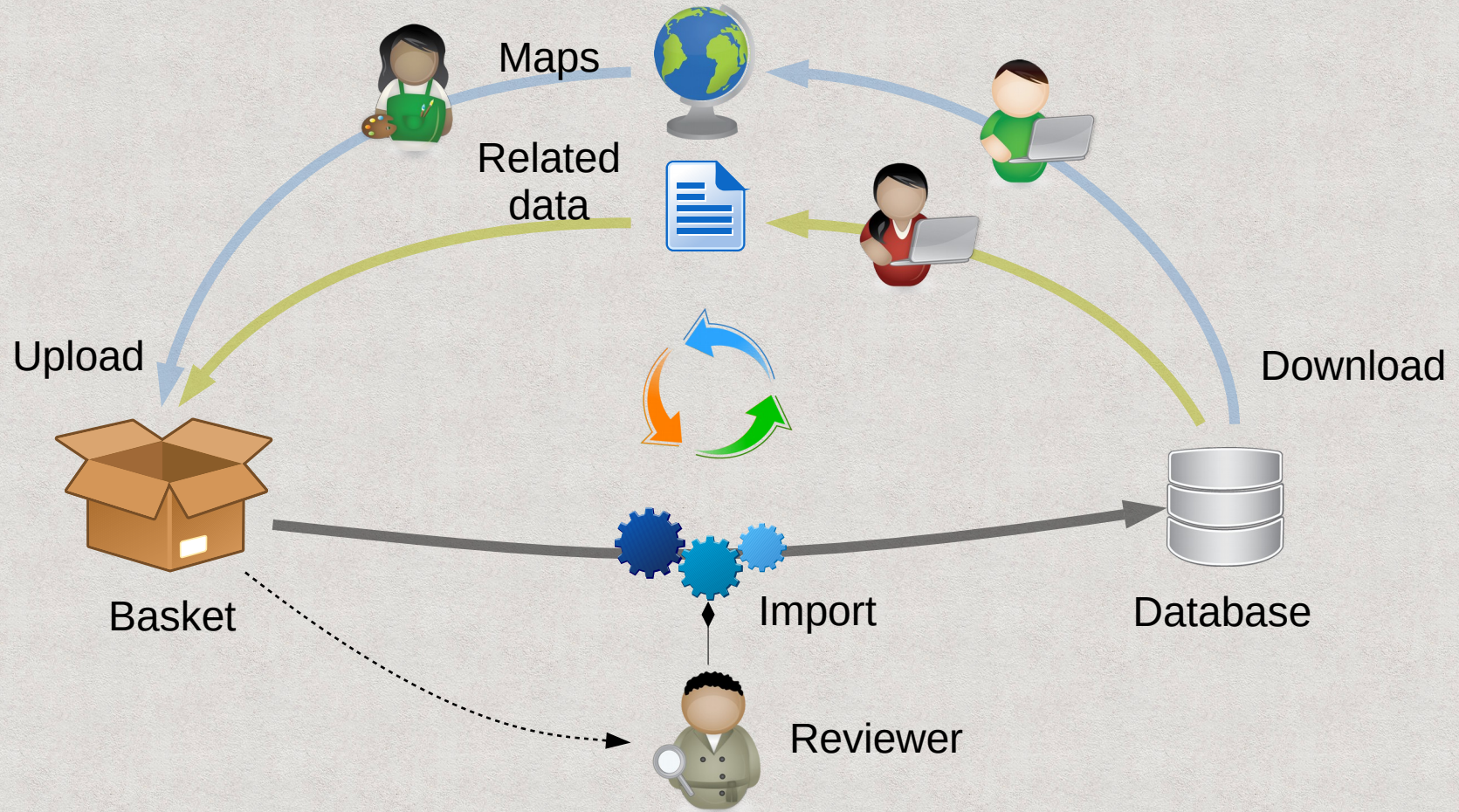
Geomatics Studio Portal

- **Geomatics:** *“the discipline of gathering, storing, processing, and delivering geographic information or spatially referenced information” (Wikipedia)*
- Online, freely accessible*: collaborative tool
- Data organized and consistent format
- Data quality continuous improvement
- Implement a workflow for the team (6-10 people: surveyors, architects, environment engineers, monitors...)
- Promote open source, open data, accessibility

* Except sensitive information



Functional workflow



Gisaf

- Code under FOSS license (GPLv3)
- > 5000 LOC (mostly Python and Typescript)
- Try to keep it as generic as possible
 - Python plugin architecture
- => Goal:
 - Make Gisaf a solution for organizations with similar needs
 - Share our experience



Software stack

- Server
 - Python, Aiohttp, Gino, Shapely, Graphene...
- Database
 - Postgis
- Browser
 - Angular, Mapbox GL
- Data analytics
 - Jupyter and Pandas



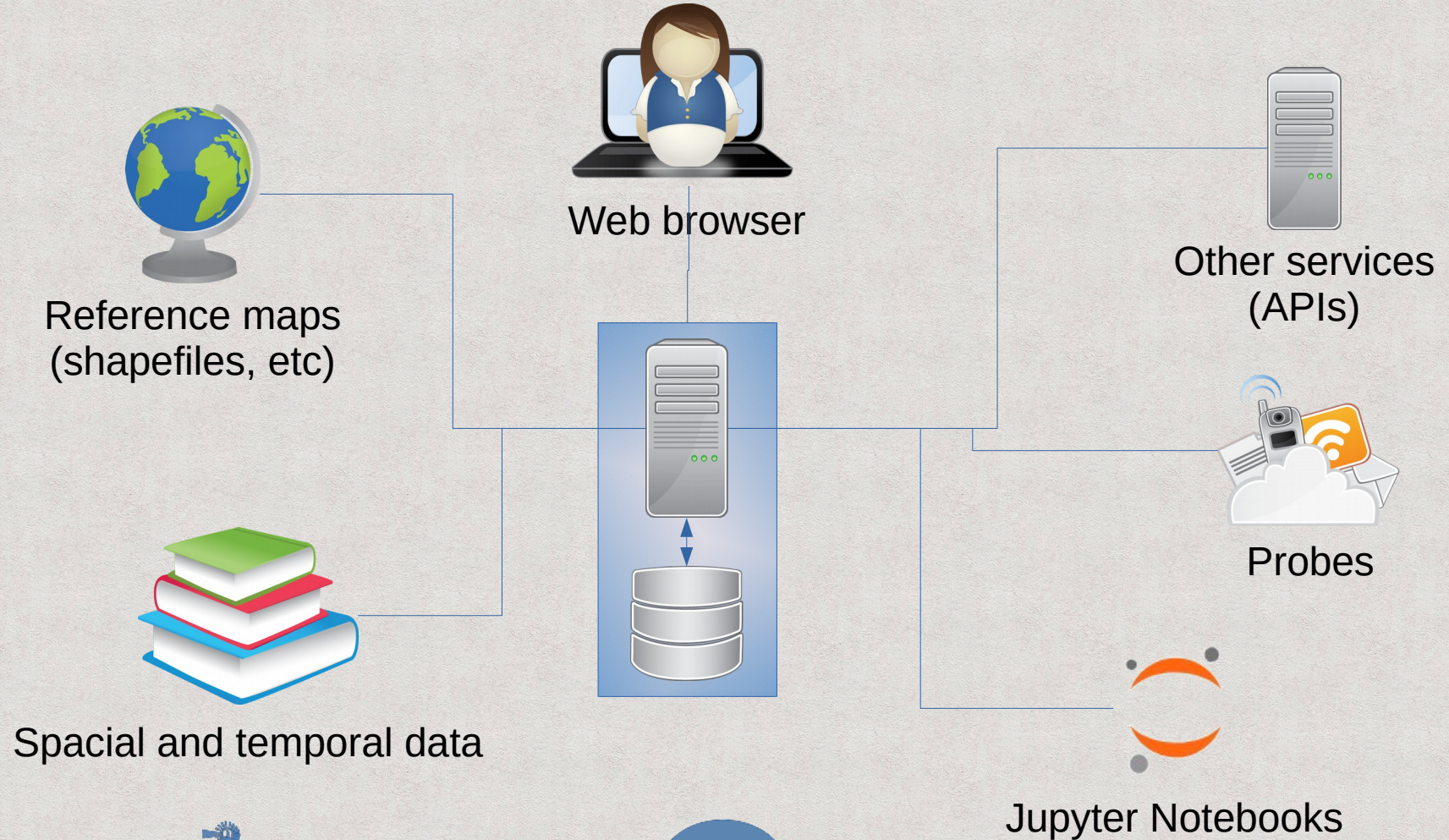
Integration

- Pandas
 - Generation of dashboards for Gisaf
- Import/export Gisaf \Leftrightarrow OSM*
 - Mapping between Gisaf database schema and OSM tags?
- Python power
 - A Swiss Army Knife that can glue heterogeneous pieces

* Planned



IO



Data analysis

- Jupyter notebooks
- Pandas and GeoPandas
 - Pandas: easy data analysis
 - Input and output from database, files, URLs...
 - Statistical and scientific analysis
 - GeoPandas is also remarkable for generating maps

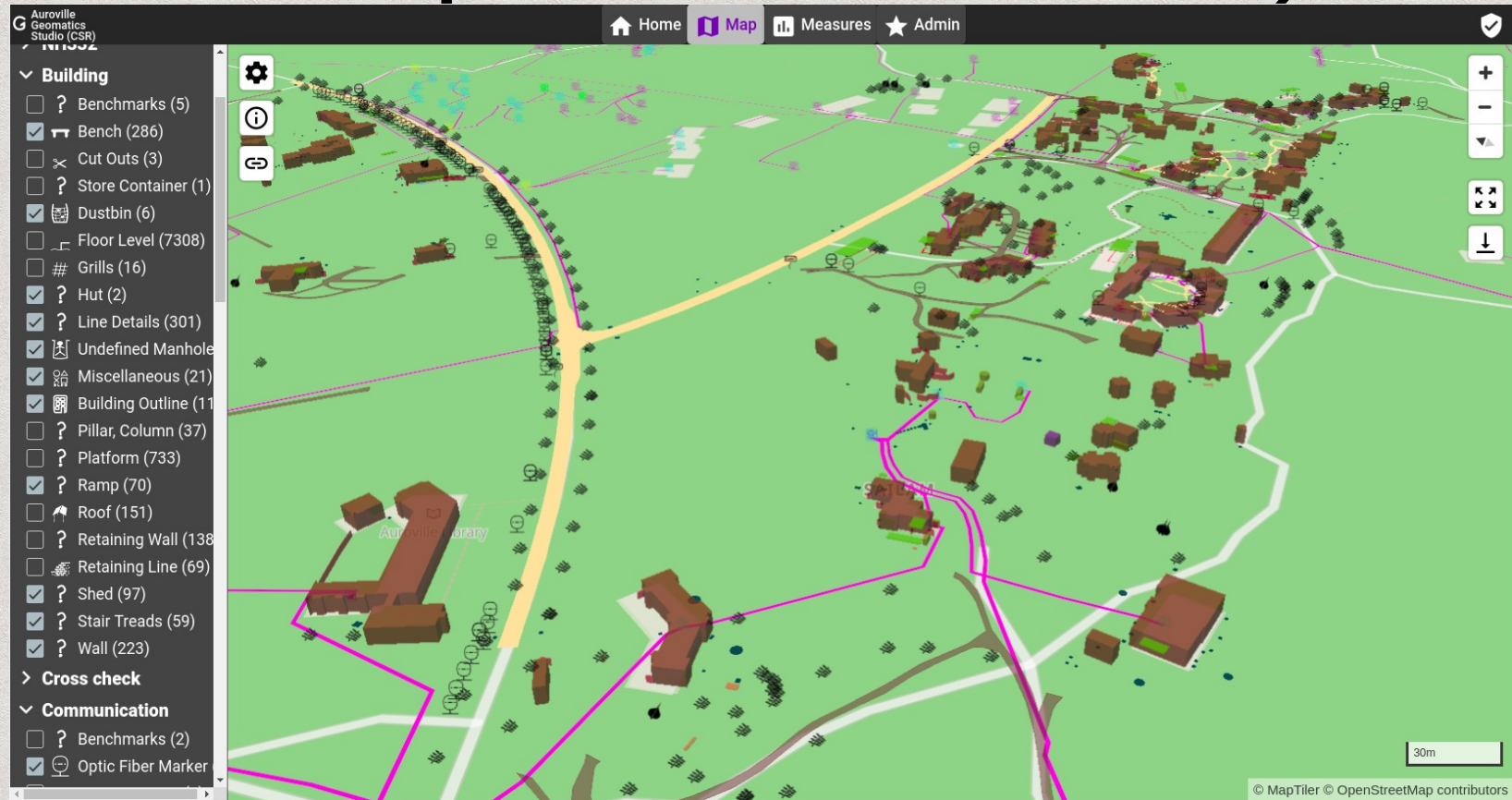


Links

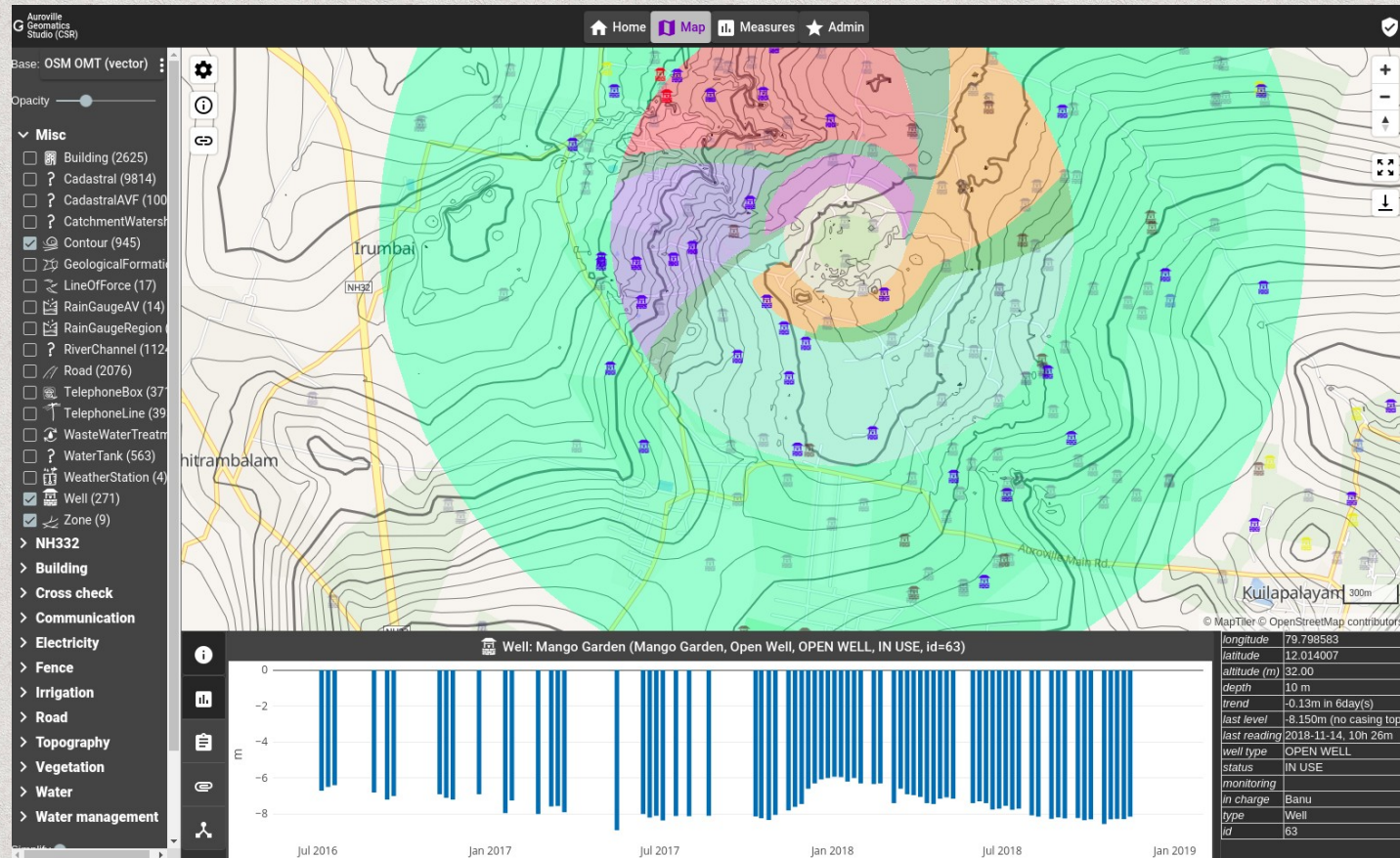
- Site: <http://gis.auroville.org.in>
- Gisaf home:
<http://redmine.auroville.org.in/projects/gisaf>



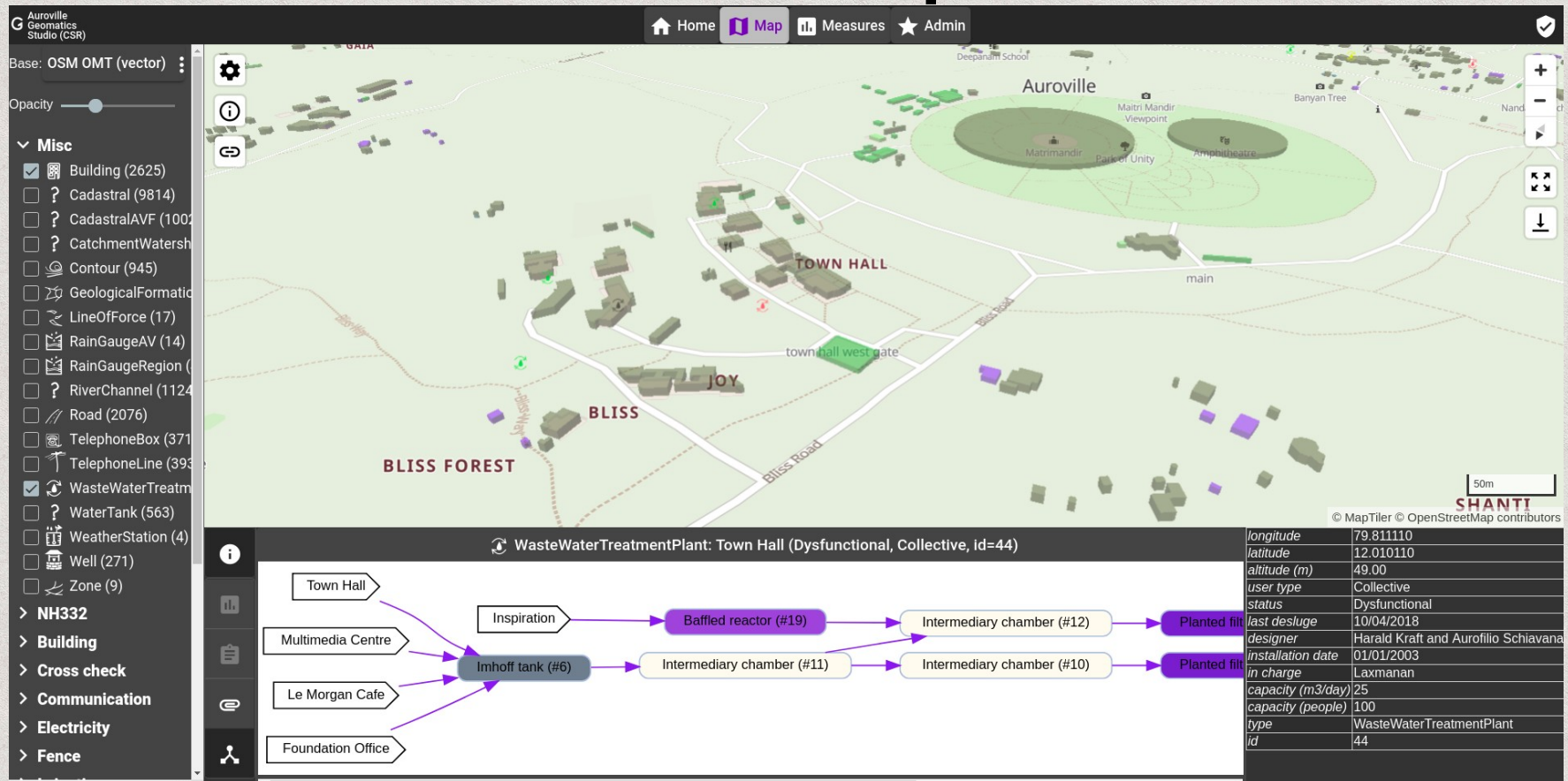
Screenshot: map (buildings, telephone lines, etc)



Screenshot: wells, with water levels



Screenshot: waste water treatment plant



Screenshot: dashboard (from Jupyter notebook)

```
File Edit View Run Kernel Tabs Settings Help
Wells.ipynb x Rain.ipynb x Geo stats.ipynb x Survey.ipynb x
+ - Copy Paste Run Cell Markdown git gisaf

Map: surveyed area

Note: the layers must be reprojected to WebMercator (EPSG 3857)

[231]: ax1.set_title('Surveyed area ({:.1f} acres)'.\
               format(surveyed_area))
       ax1.set_axis_off()
       raw_survey_points.to_crs({'init': 'epsg:3857'}).plot(ax=ax1, facecolor='#3311CC', alpha=0.2, markersize=1)
       #zones.to_crs({'init': 'epsg:3857'}).plot(ax=ax1, column='zone', alpha=0.2)
       by_project.to_crs({'init': 'epsg:3857'}).plot(ax=ax1, column='name', legend=True, alpha=0.2)
       #ax1.add_patch(PolygonPatch(survey_bounds,
       #                           fc='#99999933',
       #                           ec='#00000033',
       #                           alpha=0.3,
       #                           fill=True,
       #                           zorder=1))
       gs.add_basemap(ax1, 15, url=gs.basemaps.OSM_A)

       <Figure size 432x288 with 0 Axes>

[232]: ax1.margins(0)

Graph

[233]: raw_survey_points['time'] = pd.to_datetime(raw_survey_points['date'])

       point_histogram = raw_survey_points.groupby('time').agg({'geometry': 'count'}).rename(columns={'geometry': 'histogram'})
       point_histogram['cum'] = np.cumsum(point_histogram['histogram'])

       point_histogram.cum.plot(
           ax=ax2,
           grid=True,
           title='Number of points surveyed ({})'.format(len(raw_survey_points)),
           legend=False,
       )
```

