Netherlands Enterprise Agency

kadaster

National Institute for Public Health and the Environment
Ministry of Health, Welfare and Sport

hetWaterschapshuis

Gemeente Amsterdam

Gemeente Utrecht

GEONOVUM

3D BAG Developers' meeting 13-11-2024 online 13:00-15:30

3DGI

TU Delft 3Dgeoinfo

3DBAG Team

Balázs Dukai
Ravi Peters

3DGI

Hugo Ledoux
Ivan Pađen
Gina Stavropoulou
Jantien Stoter

TU Delft
3Dgeoinfo

Part 0 –

Introduction to 3DBAG and this meeting

3DGI

TU Delft 3Dgeoinfo

House Rules

- Make sure you are **muted** during the presentations.

- If you have any questions, **send them to the chat.** Then we can answer them during the QA sessions.

- There will be some **feedback sessions** where you will be asked to answer a few questions. You can use either your browser or your phone to participate.

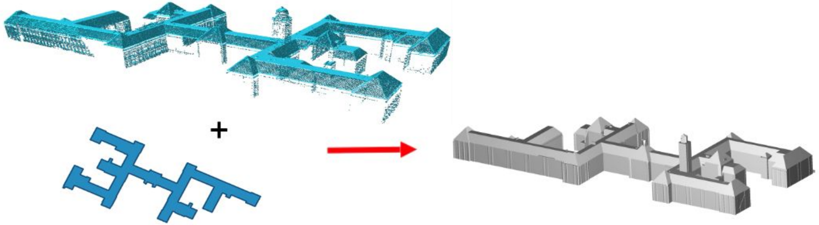
What is the 3DBAG?

Baseregister Addresses and Buildings

1. Dataset with **3D models** for **all ~ 10 million BAG-buildings** in NL
    - Reconstructed from open govern. data
        - BAG footprint + LiDAR Point cloud (AHN)
    - Available as open data

2. Software behind the dataset + hardware

Built and maintained by 3DGI TUDelft 3Dgeoinfo

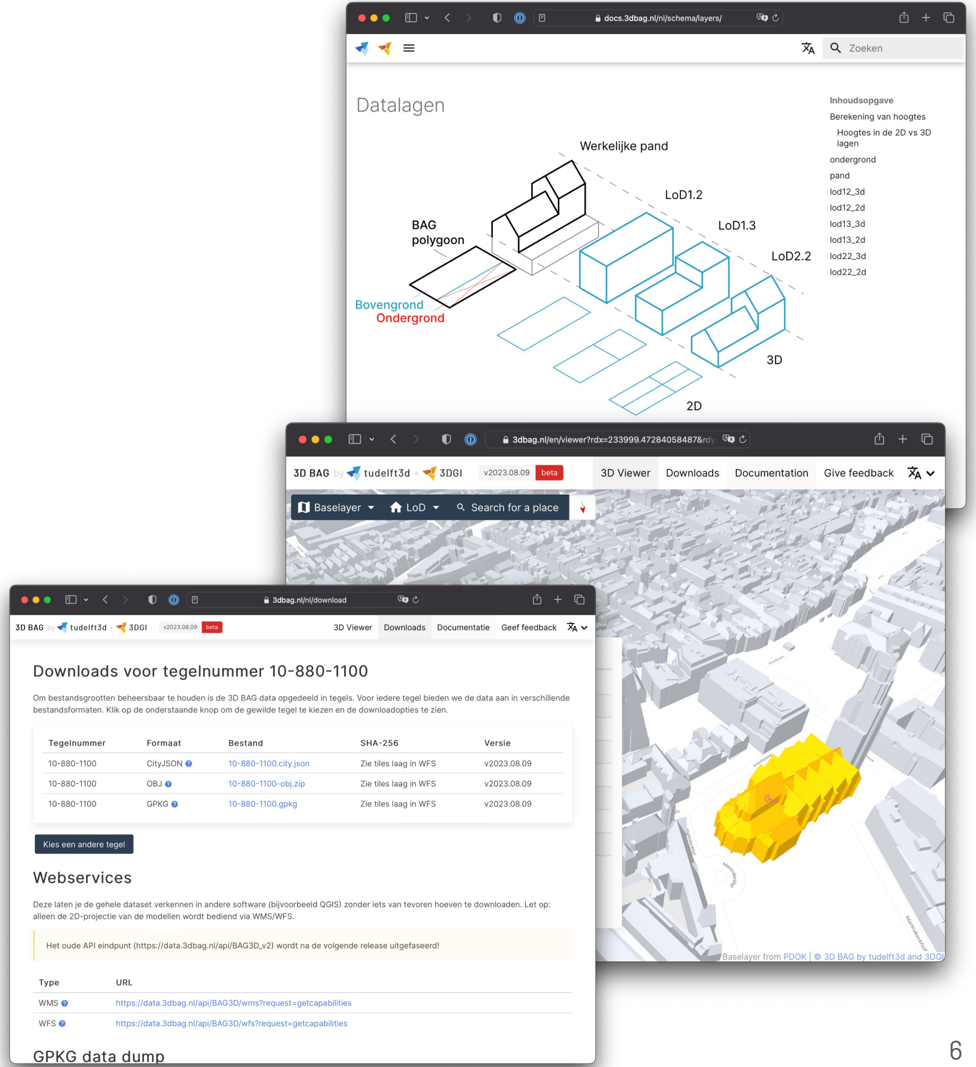


Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the netherlands.
R Peters, B Dukai, S Vitalis, J van Liempt, J Stoter.
Photogrammetric Engineering and Remote Sensing 88(3), 2022**.**

3DBAG available

Via 3dbag.nl

- 3D viewer
  - Directly view and inspect data
  - Also works on mobile
- Download data at different LoDs
  - available in common GIS/3D formats
- Documentation
  - Describing the dataset and how to use it



Downloads voor tegelnummer 10-880-1100

Om bestandsgrootten beheersbaar te houden is de 3D BAG data opgedeeld in tegels. Voor iedere tegel bieden we de data aan in verschillende bestandsformaten. Klik op de onderstaande knop om de gewilde tegel te kiezen en de downloadopties te zien.

Tegelnummer	Formaat	Bestand	SHA-256	Versie
10-880-1100	CityJSON	10-880-1100.city.json	Zie tiles laag in WFS	v2023.08.09
10-880-1100	OBJ	10-880-1100-obj.zip	Zie tiles laag in WFS	v2023.08.09
10-880-1100	GPKG	10-880-1100.gpkg	Zie tiles laag in WFS	v2023.08.09

Kies een andere tegel

Webservices

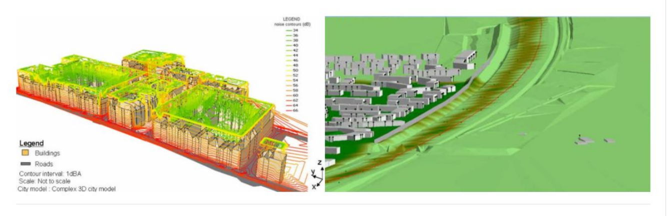
Deze laten je de gehele dataset verkennen in andere software (bijvoorbeeld QGIS) zonder iets van tevoren hoeven te downloaden. Let op: alleen de 2D-projectie van de modellen wordt bediend via WMS/WFS.

Het oude API eindpunt (https://data.3dbag.nl/api/BAG3D_v2) wordt na de volgende release uitgefaseerd!

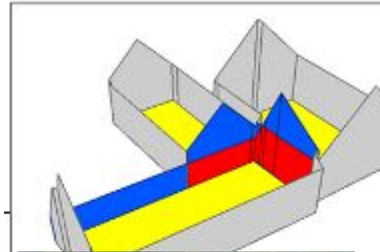
Type	URL
WMS	https://data.3dbag.nl/api/BAG3D/wms?request=getcapabilities
WFS	https://data.3dbag.nl/api/BAG3D/wfs?request=getcapabilities

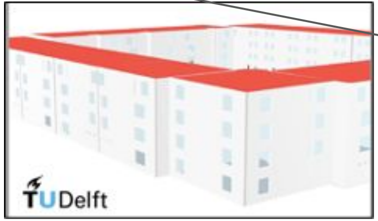
GPKG data dump

History and status 3DBAG



- Started as research projects (since 2018)
- Testbed for 3D geoinformation
  - Push state-of-the-art 3D building reconstruction
  - Push 3D innovations in urban applications (by us + others)
  - Push use of open data standards (CityJSON)
- First nationwide version 2021, several releases per year
  - Updates + improvements +
  - New features from research: volume,areas wall/roof/floor,nr floors, party walls
- Ongoing research:
  - Add openings from oblique images, use other data sources, use in BIM
- Wide use: per month:
  - ~ 100 000 tiles downloads
  - ~ 850 000 webservice queries

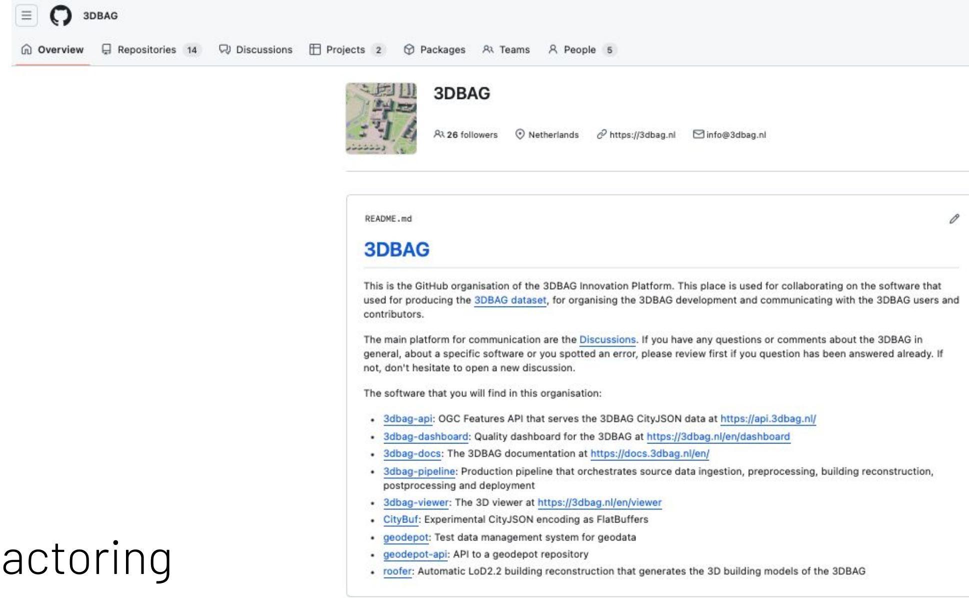




Heating energy demand
Computational Fluid Dynamics
Solar panel potential

3DBAG status

- Widely adopted; passed research phase



3DBAG

This is the GitHub organisation of the 3DBAG Innovation Platform. This place is used for collaborating on the software that used for producing the 3DBAG dataset, for organising the 3DBAG development and communicating with the 3DBAG users and contributors.

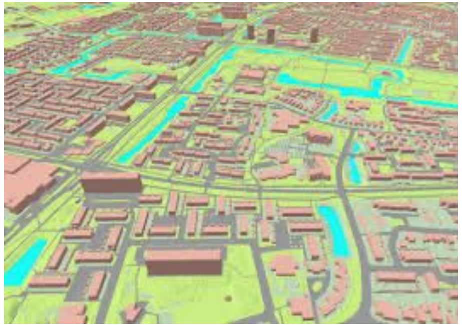
The main platform for communication are the Discussions. If you have any questions or comments about the 3DBAG in general, about a specific software or you spotted an error, please review first if you question has been answered already. If not, don't hesitate to open a new discussion.

The software that you will find in this organisation:

- 3dbag-api: OGC Features API that serves the 3DBAG CityJSON data at https://api.3dbag.nl/
- 3dbag-dashboard: Quality dashboard for the 3DBAG at https://3dbag.nl/en/dashboard
- 3dbag-docs: The 3DBAG documentation at https://docs.3dbag.nl/en/
- 3dbag-pipeline: Production pipeline that orchestrates source data ingestion, preprocessing, building reconstruction, postprocessing and deployment
- 3dbag-viewer: The 3D viewer at https://3dbag.nl/en/viewer
- CityBuf: Experimental CityJSON encoding as FlatBuffers
- geodepot: Test data management system for geodata
- geodepot-api: API to a geodepot repository
- roofer: Automatic LoD2.2 building reconstruction that generates the 3D building models of the 3DBAG

- WaU-subsidy (Kadaster) for software refactoring
  - Easier for maintenance, new releases, innovations and other developers
  - Performed by 3DGI and TU3D
  - => https://github.com/3DBAG

- Looking for stable role of data and software
  - While keeping innovation power
  - Align with 3D Basis-dataset of Kadaster



3DBAG Innovatieplatform

GEONOVUM

Utrecht

kadaster

Gemeente Amsterdam

hetWaterschapshuis

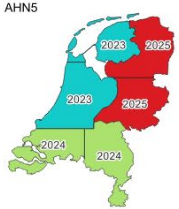
Netherlands Enterprise Agency

National Institute for Public Health and the Environment
Ministry of Health, Welfare and Sport

- Securing the future of 3DBAG:
  - Keep data (and updates) available
  - Maintain OS software
  - Collaboration in further innovations

- Exploring long term role in N-SDI landscape

- Activities 2024:
  - Start community of 3DBAG data users in NL
  - Start community of software users & developers: today!
    - also outside NL
  - AHN5 update (⅔ NL)



Today’s meeting:

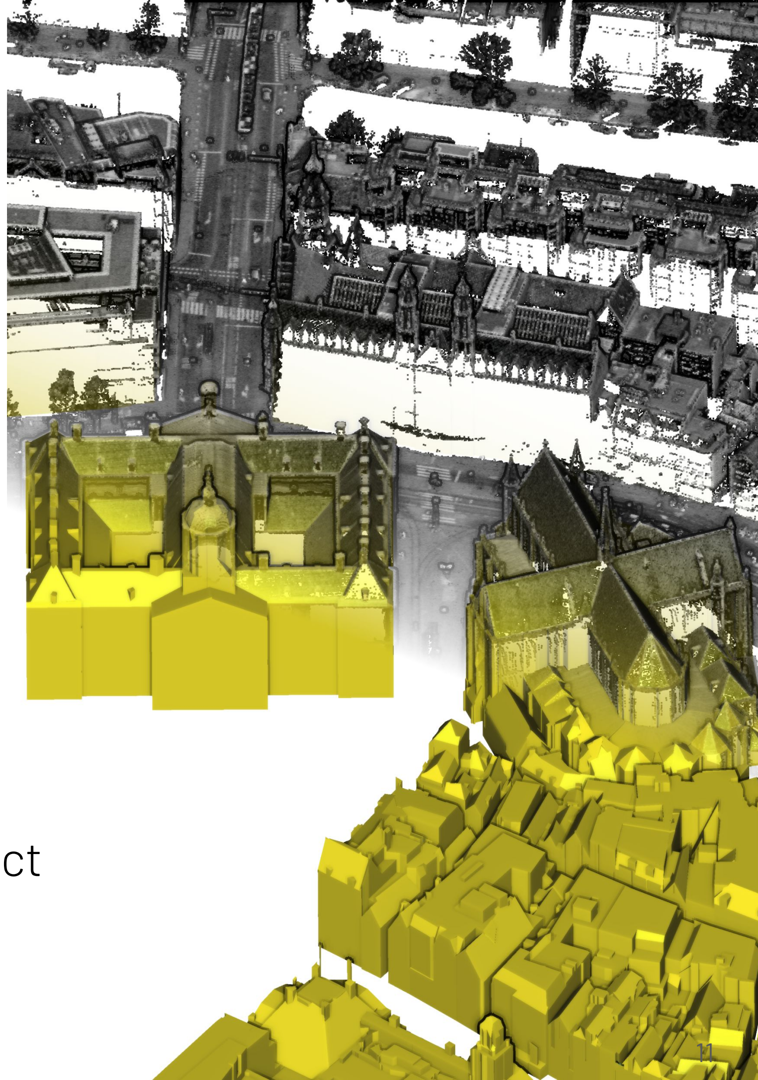
<u>Goal: start developers community</u>

- Present key components 3DBAG (plans+roadmap)
- Hear your ideas/feedback for future devs
- Discuss possible structure/organisation of dev community
  - work processes, communication platform, meetings

<u>Target audience</u>

1. Potential users of the 3DBAG software
2. Developers, potential contributors to 3DBAG project
   - More technical talk near the end of the session

- Both NL and abroad!



Overview of this meeting

- Part 1: Build your own 3DBAG (for users of the software)
  - Overview of the different software components and how to use them

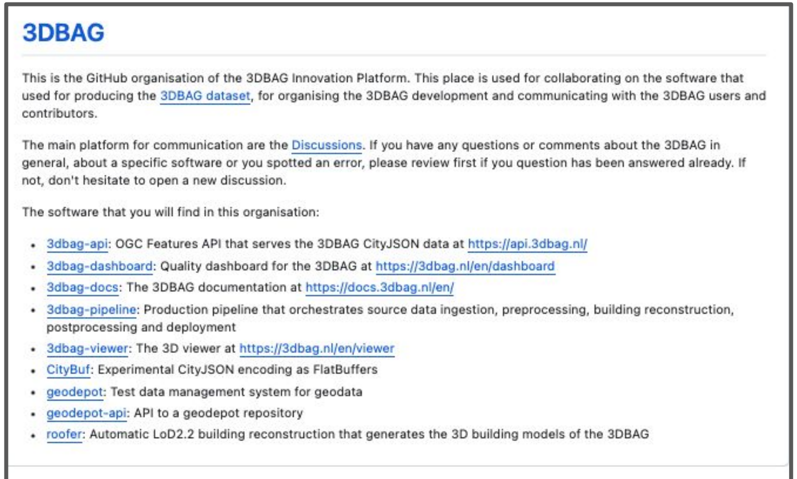
5 minute break

- Part 2: 3DBAG data access/processing for developers

5 minute break

- Part 3: Road Map and Community

During the presentations put your questions in the chat!



3DBAG

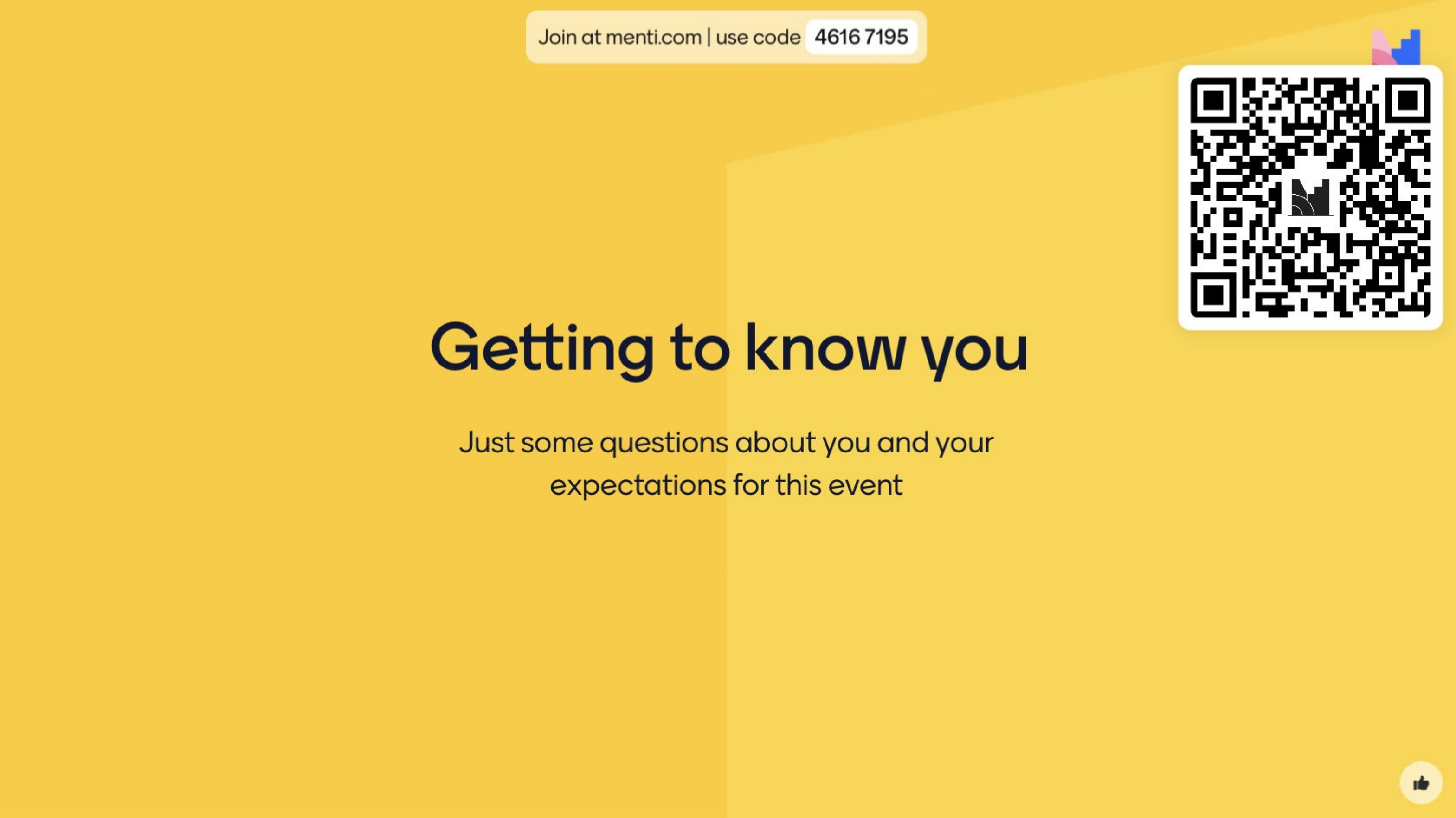
This is the GitHub organisation of the 3DBAG Innovation Platform. This place is used for collaborating on the software that used for producing the 3DBAG dataset, for organising the 3DBAG development and communicating with the 3DBAG users and contributors.

The main platform for communication are the Discussions. If you have any questions or comments about the 3DBAG in general, about a specific software or you spotted an error, please review first if you question has been answered already. If not, don't hesitate to open a new discussion.

The software that you will find in this organisation:

- 3dbag-api: OGC Features API that serves the 3DBAG CityJSON data at https://api.3dbag.nl/
- 3dbag-dashboard: Quality dashboard for the 3DBAG at https://3dbag.nl/en/dashboard
- 3dbag-docs: The 3DBAG documentation at https://docs.3dbag.nl/en/
- 3dbag-pipeline: Production pipeline that orchestrates source data ingestion, preprocessing, building reconstruction, postprocessing and deployment
- 3dbag-viewer: The 3D viewer at https://3dbag.nl/en/viewer
- CityBuf: Experimental CityJSON encoding as FlatBuffers
- geodepot: Test data management system for geodata
- geodepot-api: API to a geodepot repository
- roofer: Automatic LoD2.2 building reconstruction that generates the 3D building models of the 3DBAG

Time for some questions...



Join at menti.com | use code 4616 7195

Getting to know you

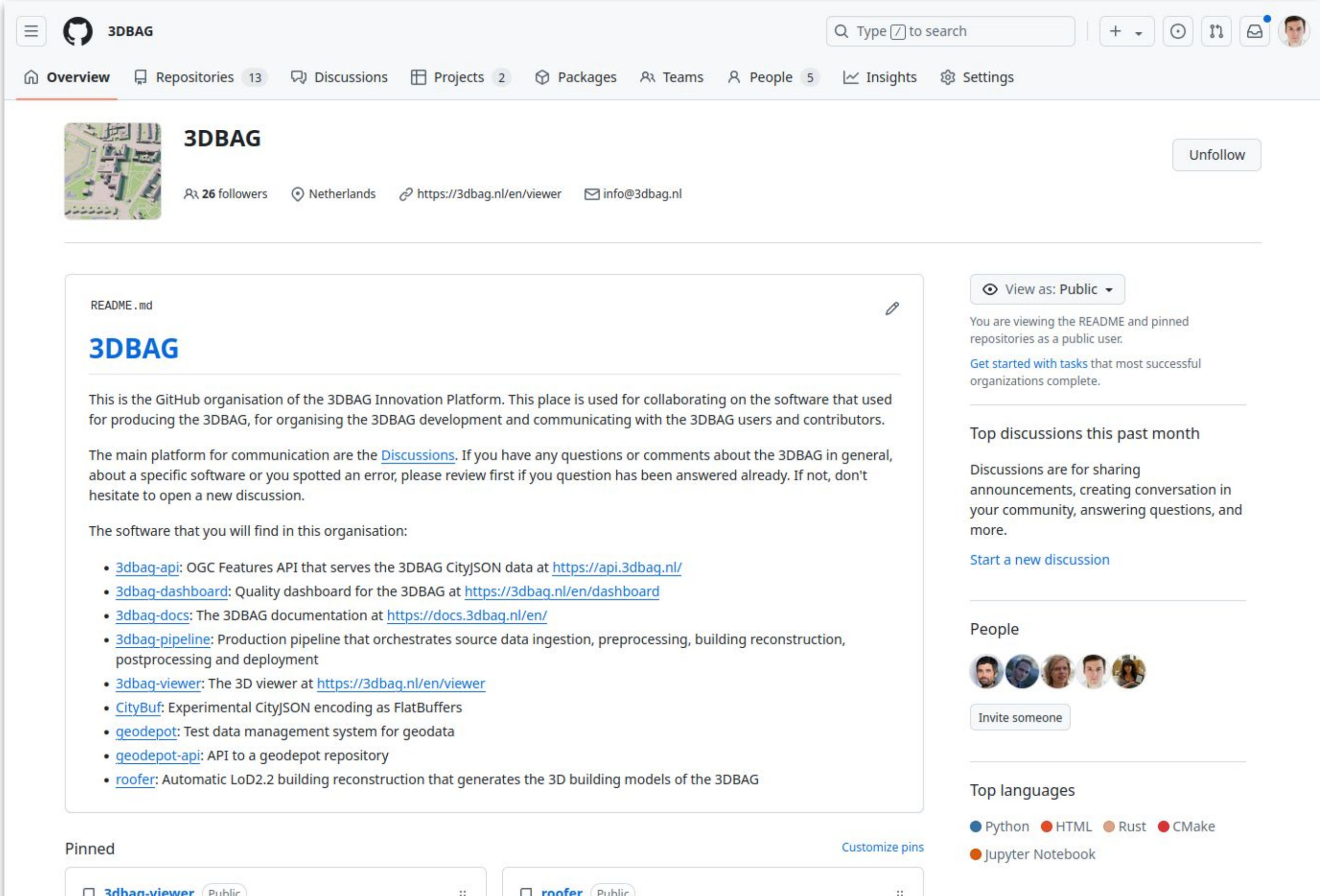
Just some questions about you and your expectations for this event

Part I – Build your own 3DBAG

3DBAG tools overview

3DBAG repositories

https://github.com/3DBAG



3DBAG

26 followers · Netherlands · https://3dbag.nl/en/viewer · info@3dbag.nl

README.md

3DBAG

This is the GitHub organisation of the 3DBAG Innovation Platform. This place is used for collaborating on the software that used for producing the 3DBAG, for organising the 3DBAG development and communicating with the 3DBAG users and contributors.

The main platform for communication are the Discussions. If you have any questions or comments about the 3DBAG in general, about a specific software or you spotted an error, please review first if you question has been answered already. If not, don't hesitate to open a new discussion.

The software that you will find in this organisation:

- 3dbag-api: OGC Features API that serves the 3DBAG CityJSON data at https://api.3dbag.nl/
- 3dbag-dashboard: Quality dashboard for the 3DBAG at https://3dbag.nl/en/dashboard
- 3dbag-docs: The 3DBAG documentation at https://docs.3dbag.nl/en/
- 3dbag-pipeline: Production pipeline that orchestrates source data ingestion, preprocessing, building reconstruction, postprocessing and deployment
- 3dbag-viewer: The 3D viewer at https://3dbag.nl/en/viewer
- CityBuf: Experimental CityJSON encoding as FlatBuffers
- geodepot: Test data management system for geodata
- geodepot-api: API to a geodepot repository
- roofer: Automatic LoD2.2 building reconstruction that generates the 3D building models of the 3DBAG

Pinned

3dbag-viewer Public

roofer Public

You are viewing the README and pinned repositories as a public user.

Get started with tasks that most successful organizations complete.

Top discussions this past month

Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

Start a new discussion

People

Top languages

● Python ● HTML ● Rust ● CMake ● Jupyter Notebook

3DBAG repositories

https://github.com/3DBAG

Experiments

- CityBuf

Develop

- geodepot
- geodepot-api

Generate

- *-pipeline
- roofer

Publish

- *-api
- *-dashboard
- *-docs
- *-viewer

3DBAG tech stack

LAStools

pdal

PostGIS

Spatial PostgreSQL

R

python™

GEOS

CGAL

GDAL

C++

GeoPackage

LAS

docker

OpenAPI

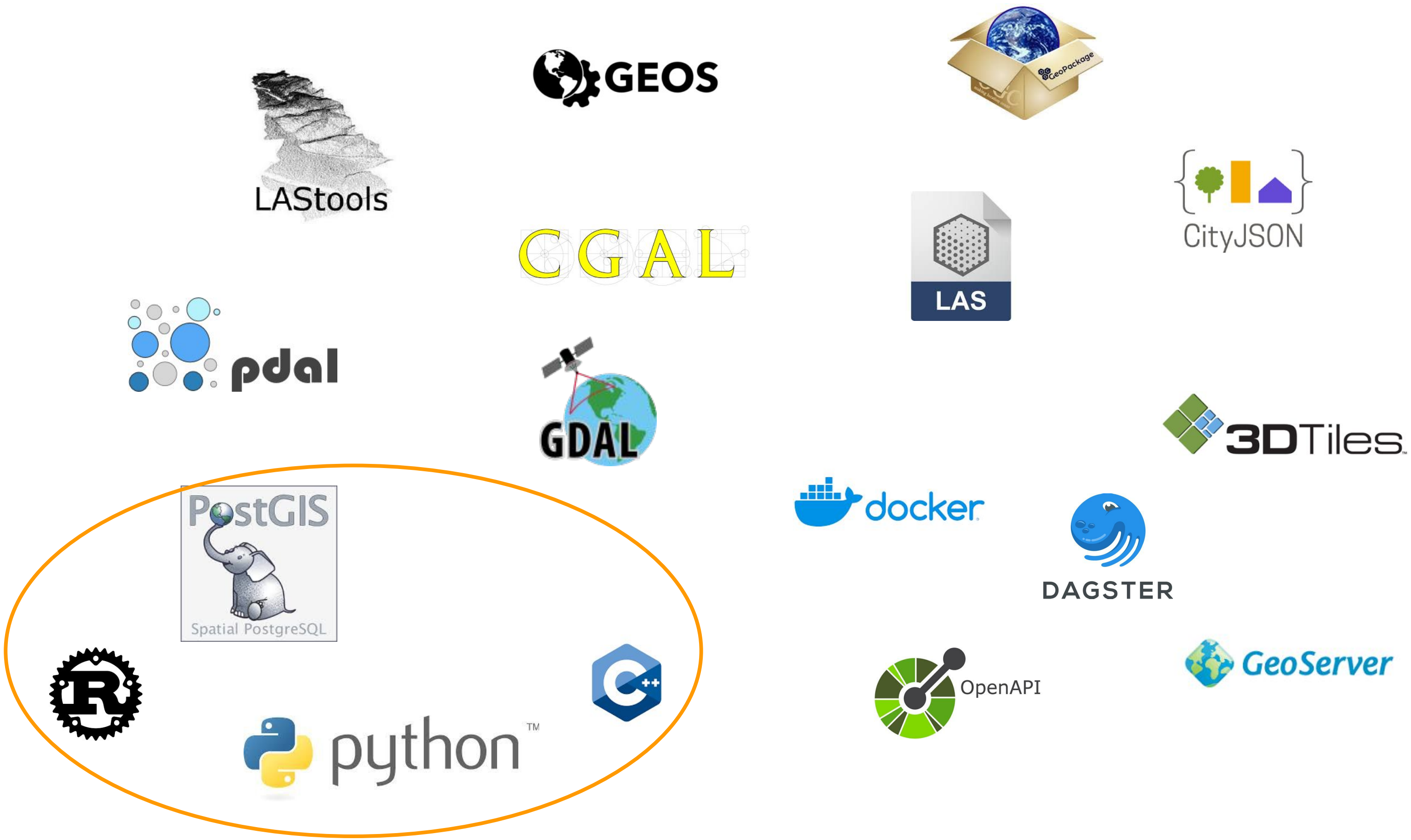
CityJSON

3DTiles

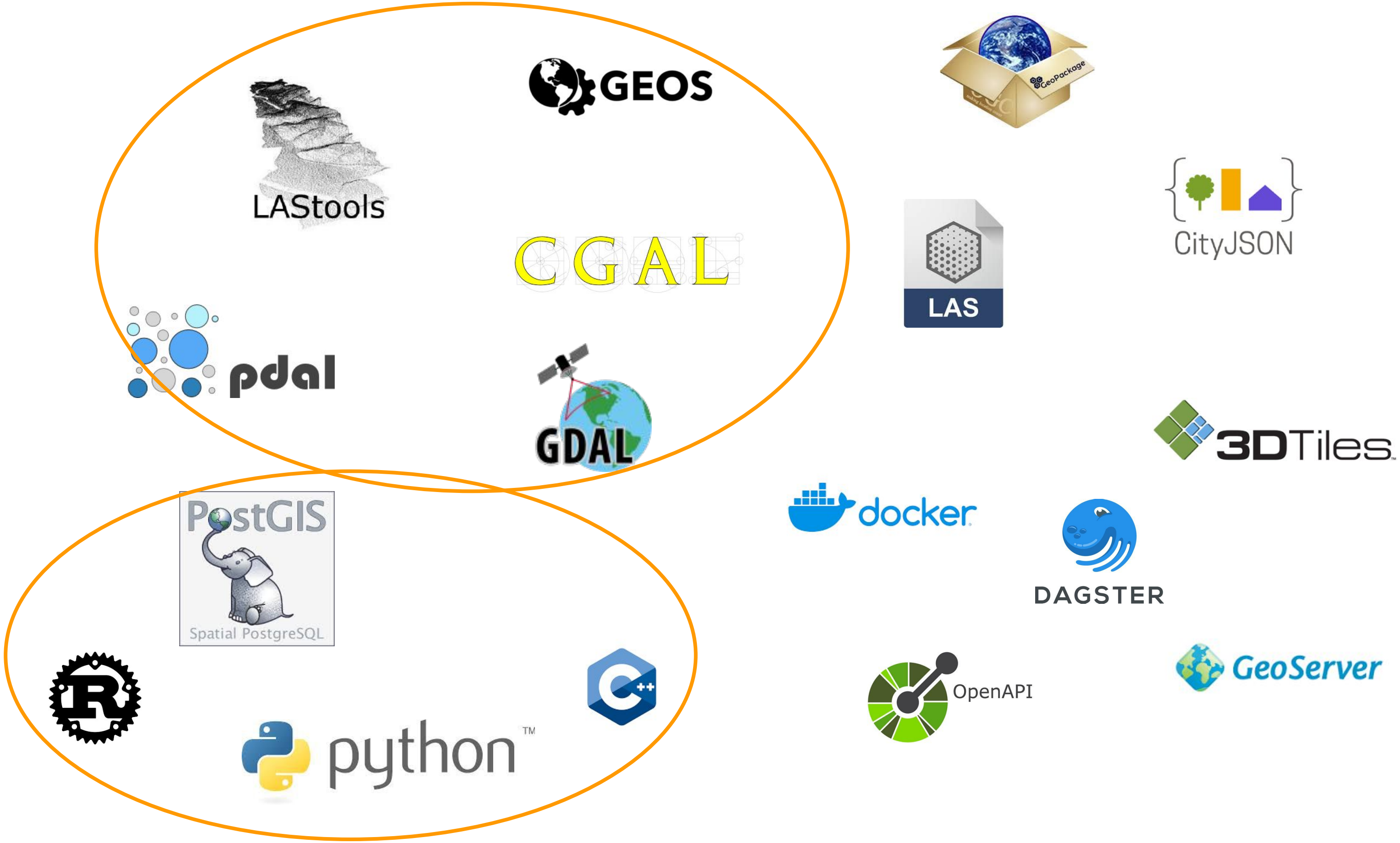
DAGSTER

GeoServer

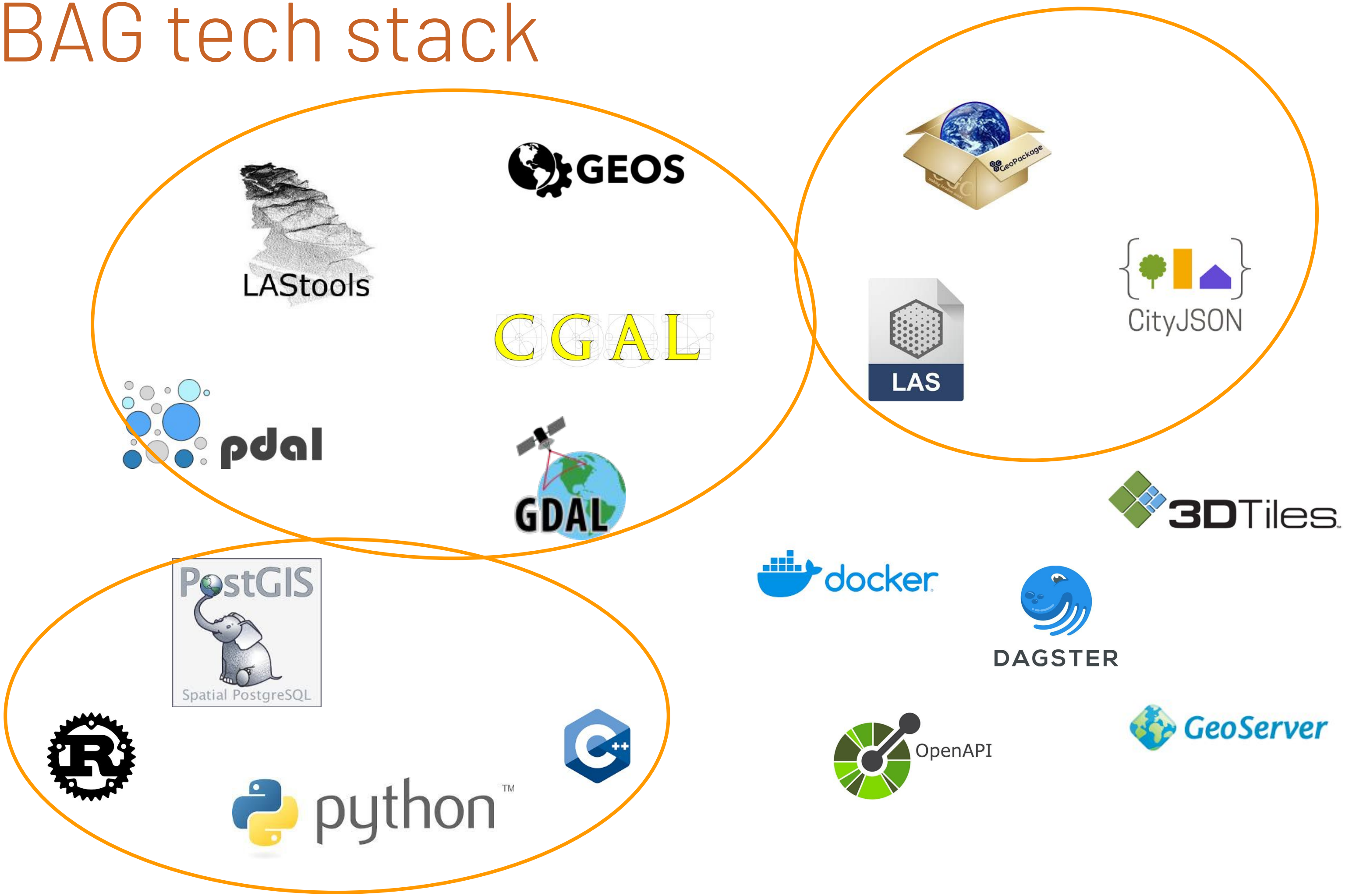
3DBAG tech stack



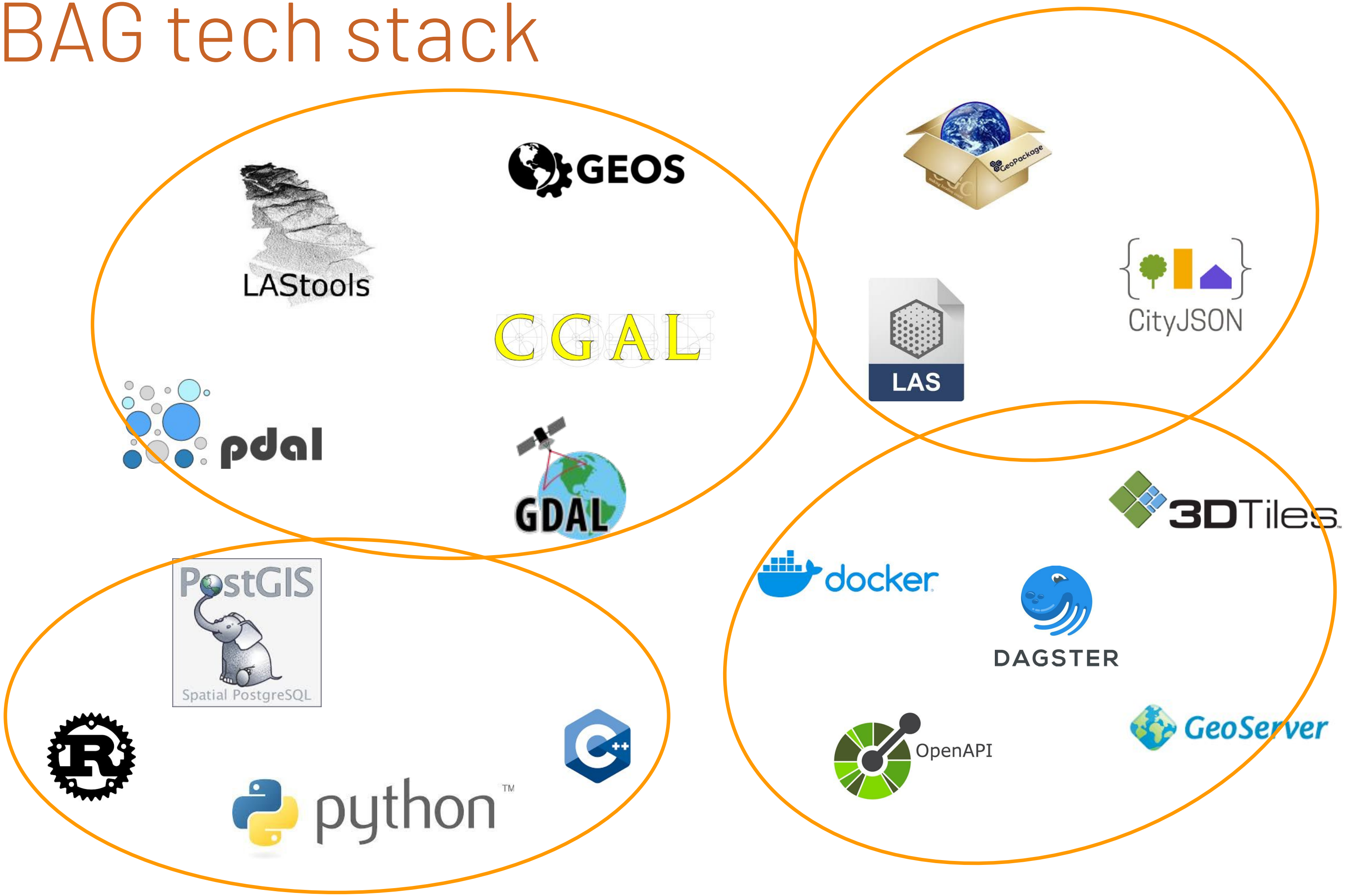
3DBAG tech stack



3DBAG tech stack



3DBAG tech stack



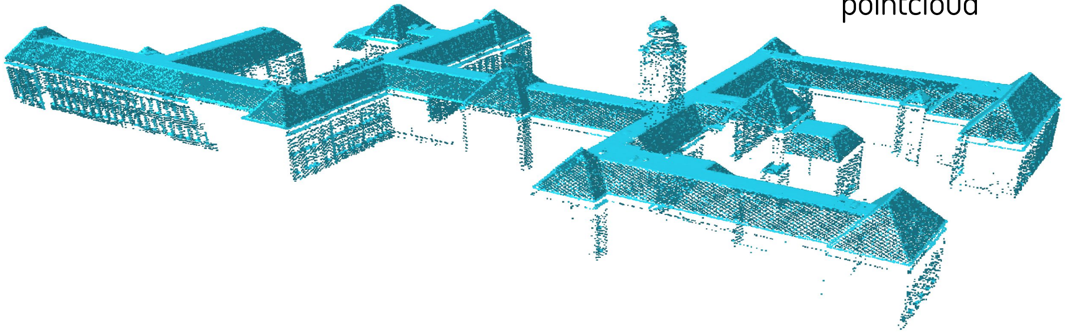
roofer

What is it?

roofer is an application to do large scale automatic LoD2 building reconstruction

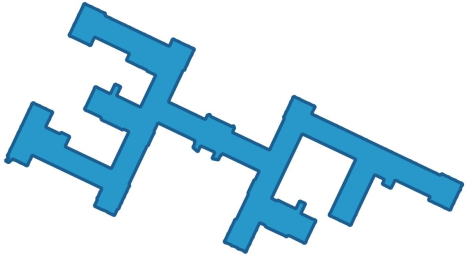
* core reconstruction algorithm also available through Python and C++ API

What does it do?

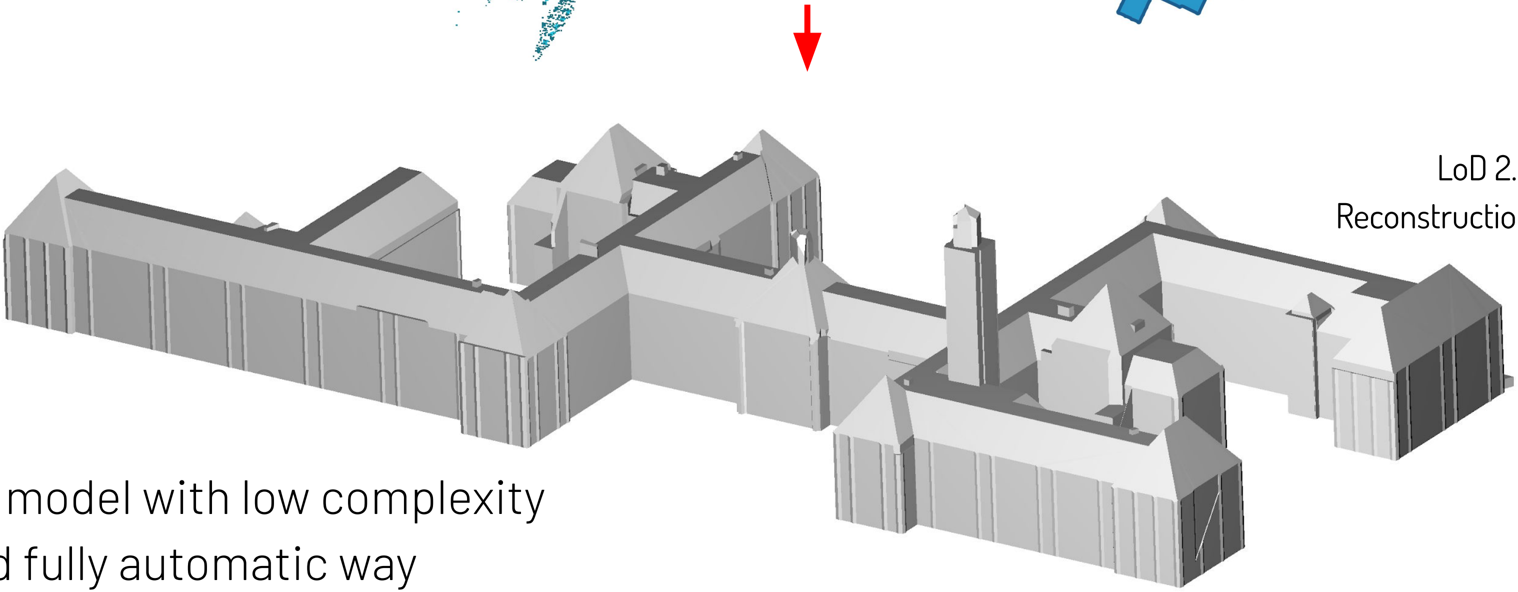


pointcloud

+



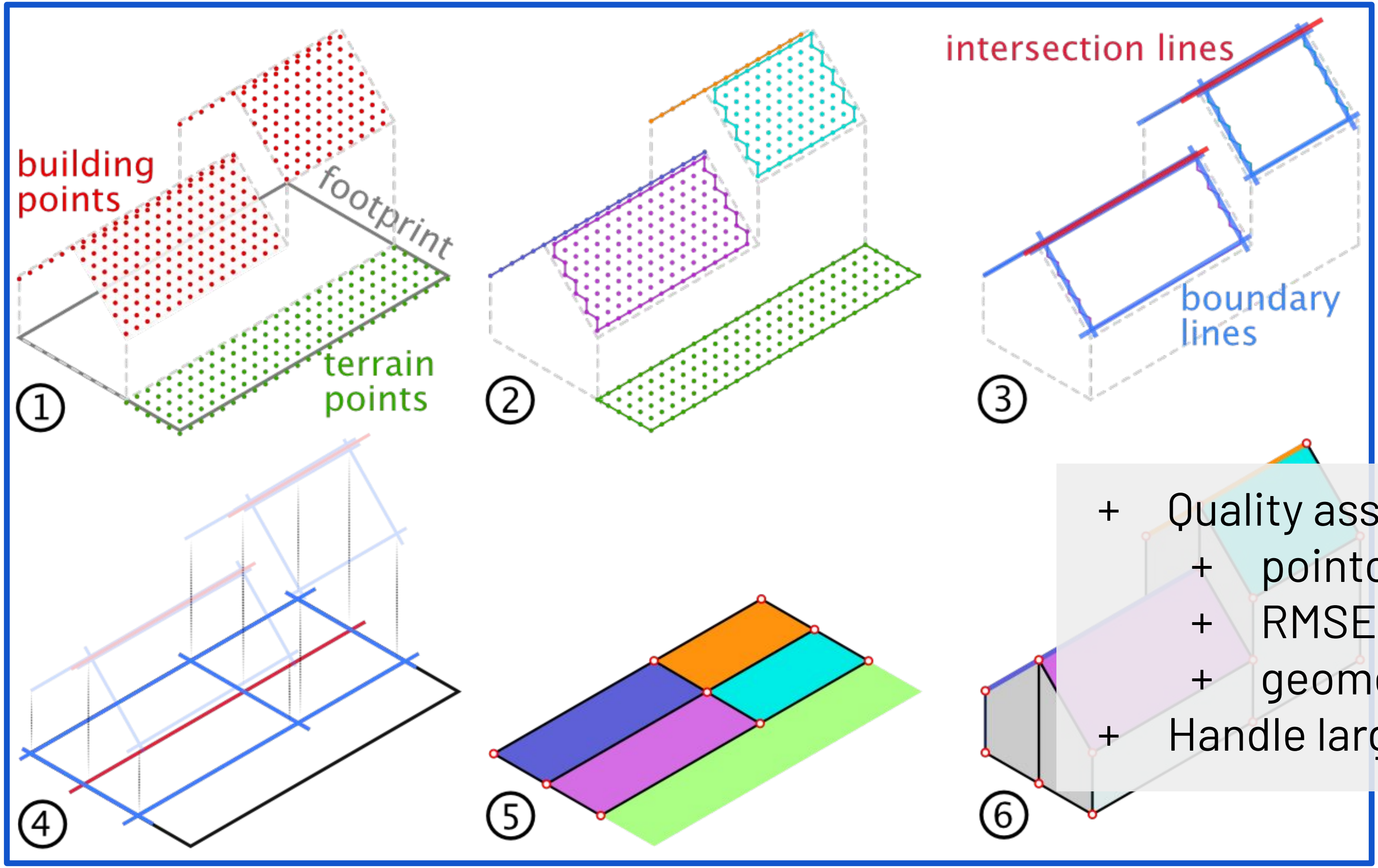
footprint polygon



LoD 2.2
Reconstruction

- Accurate 3D model with low complexity
- In robust and fully automatic way

What does it do?

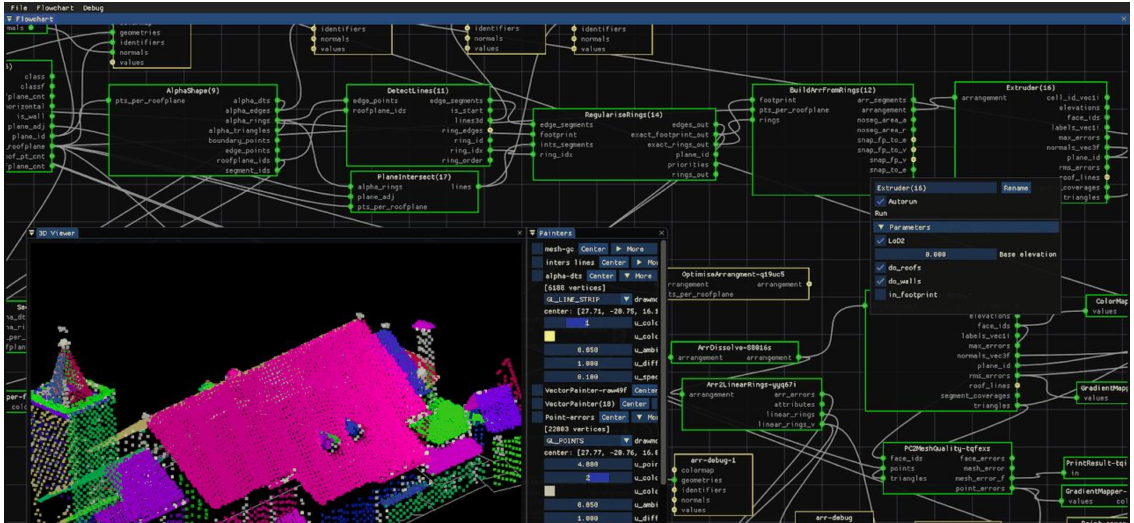


The core building reconstruction algorithm

- Quality assessment
  - pointcloud metrics
  - RMSE to pointcloud
  - geometry validation
- Handle large areas

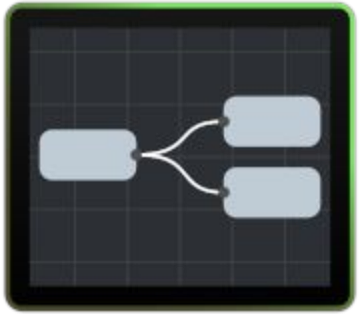
History of building reconstruction algorithm

- Originally developed as a plugin to **Geoflow**
- Geoflow = research/algorithm development tool for 3D geodata.
  - Mix of low level 'building blocks' and high level flowchart logic to develop an algorithm
- Geoflow allowed for interactive and visual experimentation with single buildings.
- Great for *developing* the reconstruction algorithm.



Geoflow GUI

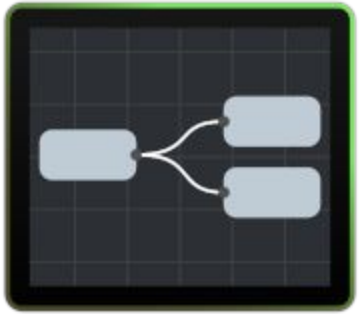
Geoflow limitations



But…

- Geoflow itself was a prototype
  - not really designed for production use
- Difficult to reconstruct large areas due to architectural limitations.
  - Pre-tiling data was needed
  - Multithreading through external tools.
- We made it work, but it was very brittle due to many 'hacks' to work around limitations
  - Silent failures
  - Flowchart architecture had little added value in production use
- Development and maintenance was not sustainable

Geoflow limitations



But…

- Geoflow itself was a prototype
  - not really designed for production use
- Difficult to reconstruct large areas due to architectural limitations.
  - Pre-tiling data was needed
  - Multithreading through external tools.
- We made it work, but it was very brittle due to many 'hacks' to work around limitations
  - Silent failures
  - Flowchart architecture had little added value in production use
- Development and maintenance was not sustainable

We learned a lot, but *time to decouple the building reconstruction algorithm….*

geoflow-bundle → **roofer**

roofer is a single application focused purely on automatic building reconstruction

- in a reliable way
  - each input feature → one output feature
  - no silent failures
- focus on usability
  - minimise preprocessing needs,
    - smartly select subset of large inputs
    - no pre-tiling
  - proper validation of input data and arguments
  - clear error messages

geoflow-bundle → **roofer**

roofer is a single application focused purely on automatic building reconstruction

- for efficient large scale reconstruction
  - built-in multithreading
  - memory efficient
- capable but simple design that is easy to maintain
  - single code base, no separate plugins
  - minimal external dependencies

roofer architecture

modules and external dependencies

Core
Reconstruction algorithm

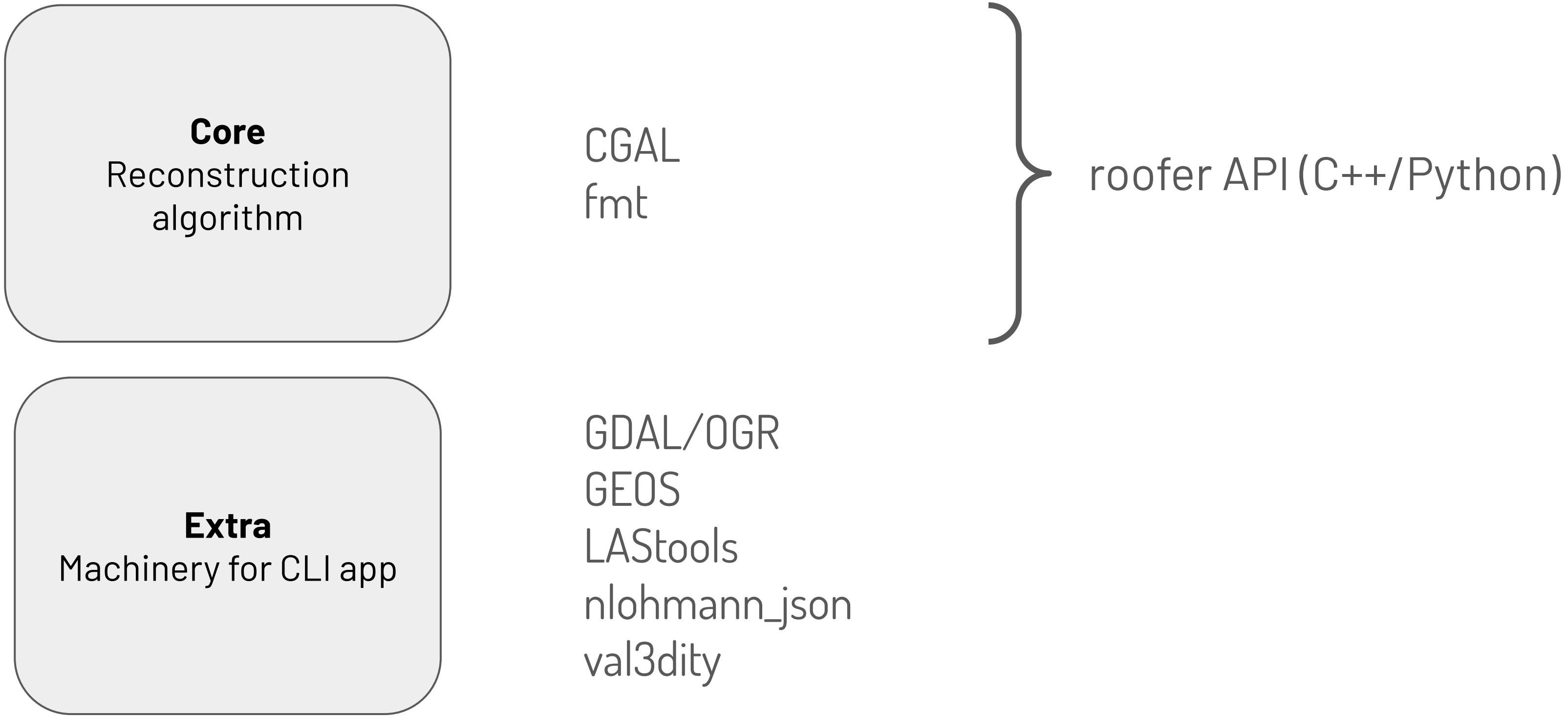
CGAL
fmt

Extra
Machinery for CLI app

GDAL/OGR
GEOS
LAStools
nlohmann_json
val3dity

roofer architecture

modules and external dependencies



Core
Reconstruction algorithm

CGAL
fmt

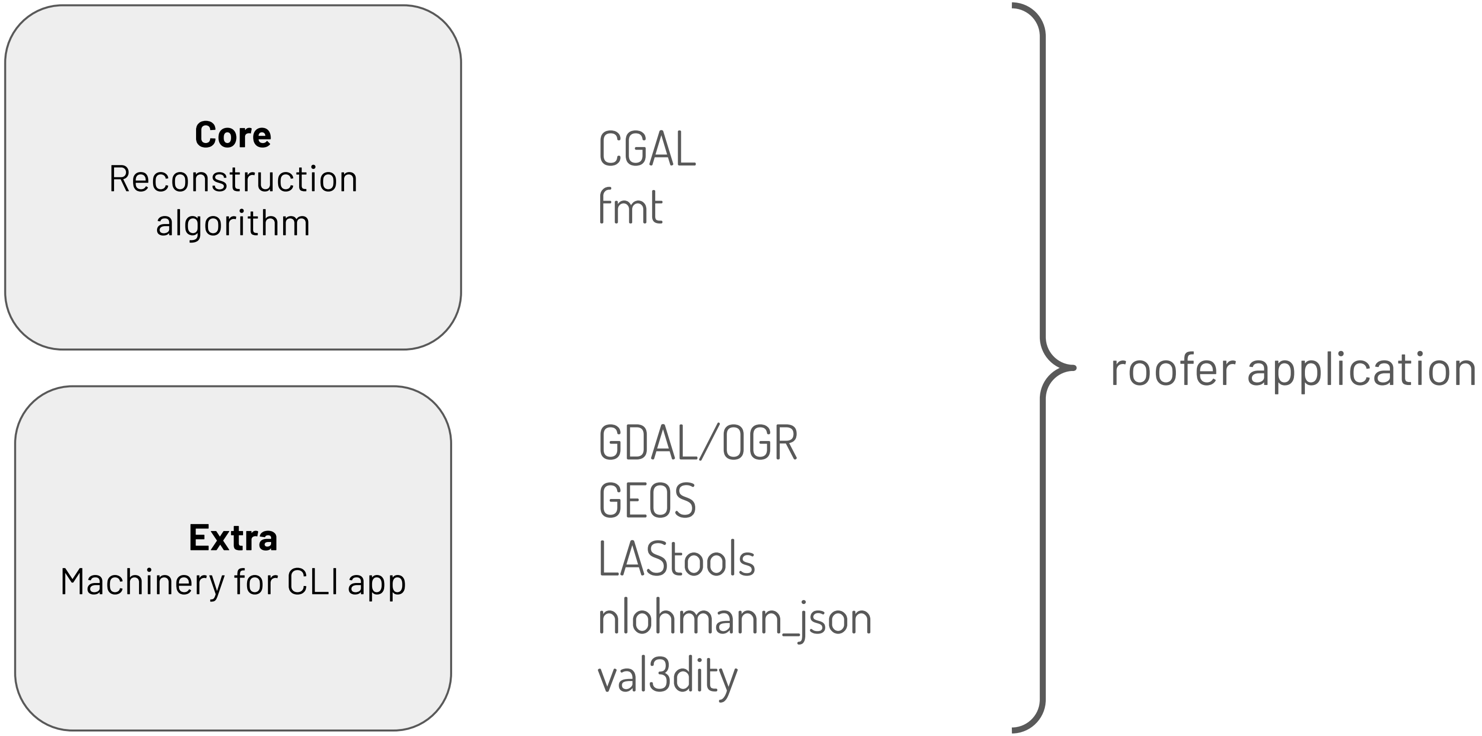
roofer API (C++/Python)

Extra
Machinery for CLI app

GDAL/OGR
GEOS
LAStools
nlohmann_json
val3dity

roofer architecture

modules and external dependencies



Core
Reconstruction algorithm

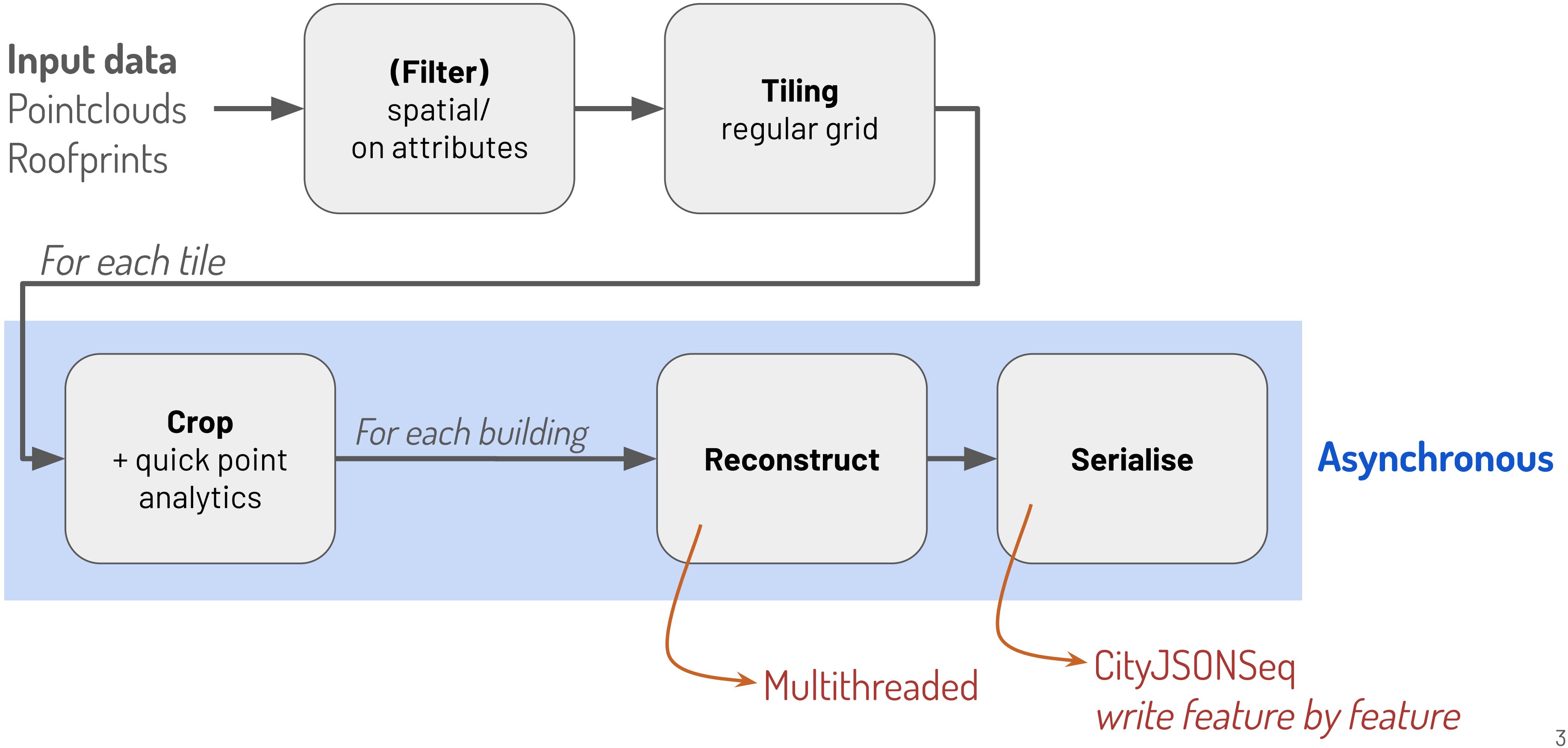
CGAL
fmt

Extra
Machinery for CLI app

GDAL/OGR
GEOS
LAStools
nlohmann_json
val3dity

roofer application

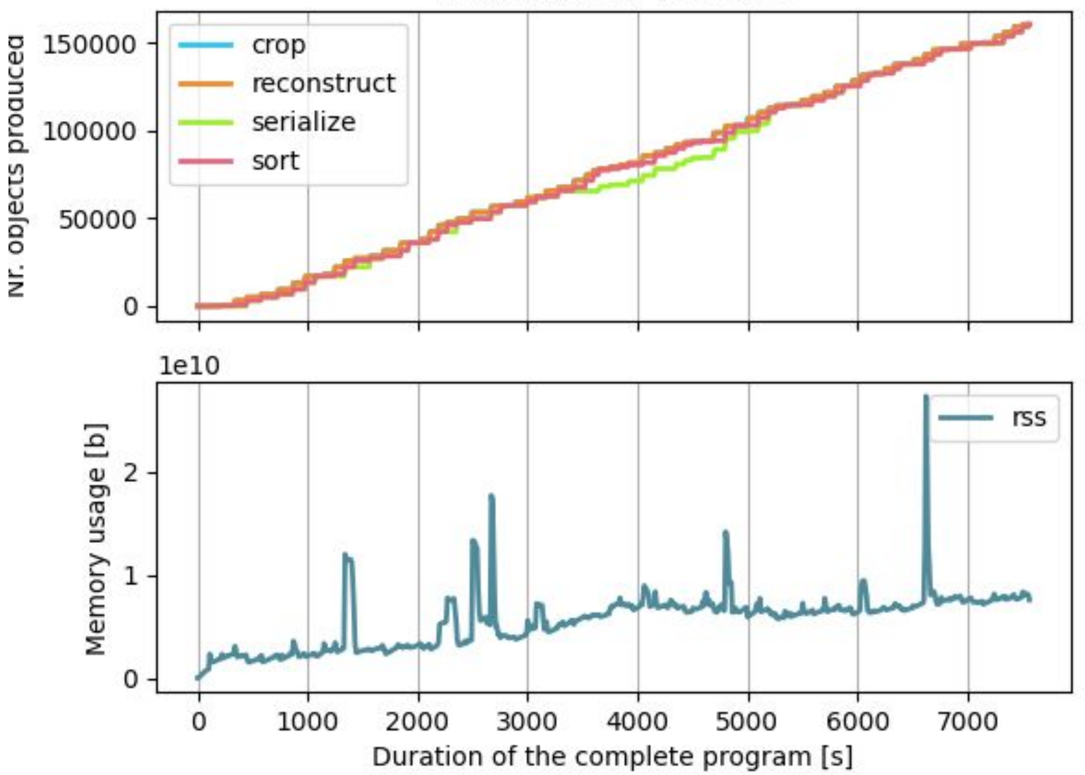
roofer processing pipeline (CLI app)



roofer data pipeline (CLI app)

In practice

- All steps happen in parallel
- Memory usage is stable



How to use roofer?

1. Through API (Python/C++)
    a. developers
2. Through CLI application
    a. most people

Roofer Python API

3DBAG/roofer

Python example:

```
import rooferpy

# ... get the input pointcloud and footprint polygon for your building

# Set the reconstruction configuration
roofer_config = rooferpy.ReconstructionConfig()
roofer_config.complexity_factor = 0.7 # Change the default values if needed

# Reconstruct
print("Reconstructing building...")
meshes = rooferpy.reconstruct(points_roof, points_terrain, footprint, roofer_config)
```

`pip install roofer` is coming soon

Roofer C++ API

3DBAG/roofer

C++ example:

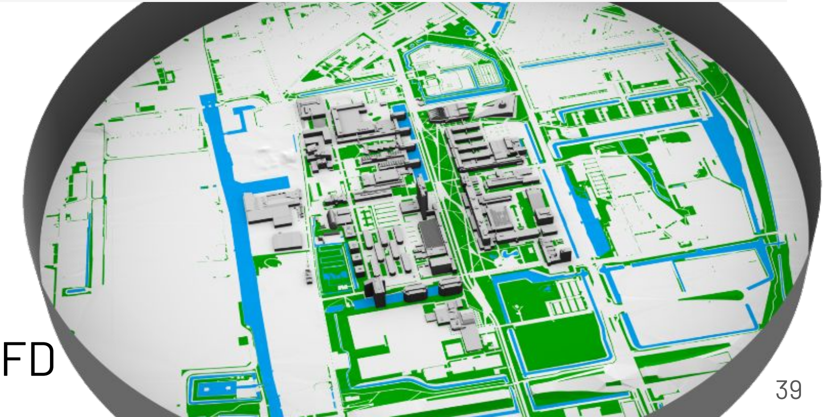
```
#include <roofer/roofer.h>

// ... get the input pointcloud and footprint polygon for your building

auto meshes =
     roofer::reconstruct(points_roof, points_terrain, footprints.front(),
{.complexity_factor = 0.7});
```

Add to your project via CMake

Used in tudelft3d/City4CFD



Roofer CLI application

3DBAG/roofer

Designed to efficiently reconstruct large areas with many buildings

- Capable of handling large input datasets
- Using multithreading for efficient computation
- Outputs 3D models as CityJSONSequence

Basic usage:

```
$ roofer pointcloud.las roofprints.gpkg output_folder/
```

Creates `output_folder/` with CityJSONSeq files

Roofer CLI application

3DBAG/roofer

Designed to efficiently reconstruct large areas with many buildings

- Capable of handling large input datasets
- Using multithreading for efficient computation
- Outputs 3D models as CityJSONSequence

Usage select subset of large input datasets:

```
$ roofer
  /folder/with/all/LAZ/of/NL/
  “PG:dbname='NL' host='localhost' tables='roofprints'”
  output_folder/
  --box 0 0 100 100
  --filter “YearOfConstruction>2000”
```

Roofer CLI application

3DBAG/roofer

Designed to efficiently reconstruct large areas with many buildings

- Capable of handling large input datasets
- Using multithreading for efficient computation
- Outputs 3D models as CityJSONSequence

Usage select subset of large input datasets:

```
$ roofer
    /folder/with/all/LAZ/of/NL/
    “PG:dbname='NL' host='localhost' tables='roofprints'”
    output_folder/
    --box 0 0 100 100
    --filter “YearOfConstruction>2000”
```

Country wide datasets

Subset on bounding box + attribute

Roofer CLI application

3DBAG/roofer

Designed to efficiently reconstruct large areas with many buildings

- Capable of handling large input datasets
- Using multithreading for efficient computation
- Outputs 3D models as CityJSONSequence

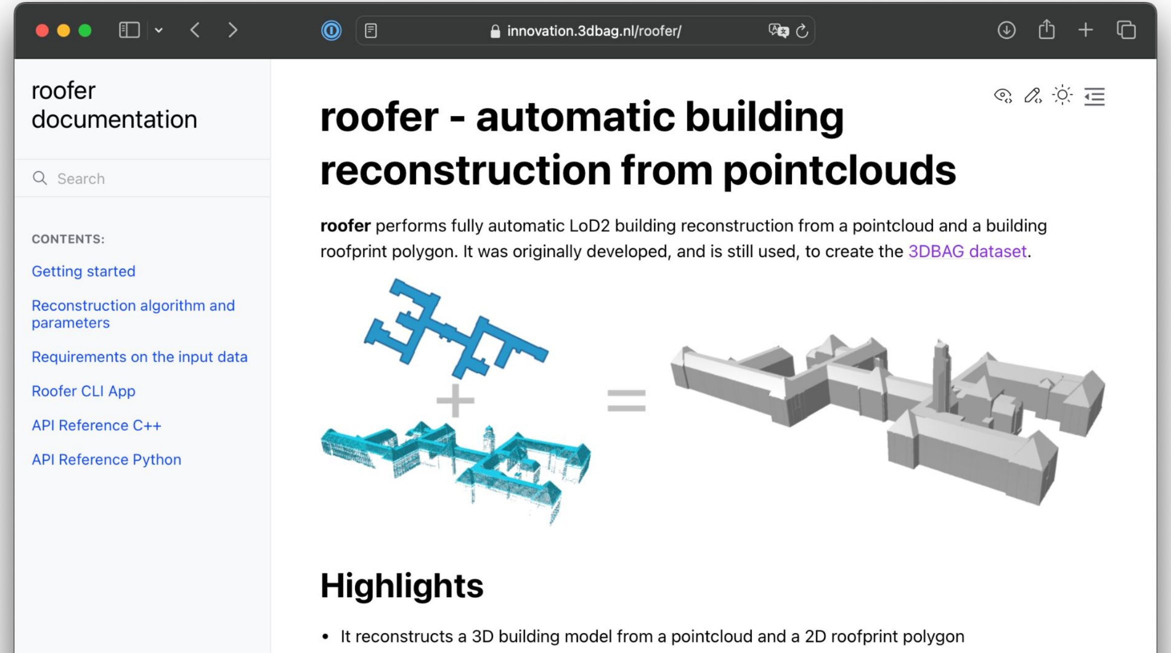
Usage ~~select subset of~~ large input datasets:

```
$ roofer
    /folder/with/all/LAZ/of/NL/
    “PG:dbname='NL' host='localhost' tables='roofprints'”
    output_folder/
```

Country wide datasets

Current status

- Still under active development, 95% done
- Documentation is available
- **develop**-branch can be used, we test before pushing



innovation.3dbag.nl/roofer/

roofer documentation

Search

CONTENTS:

- Getting started
- Reconstruction algorithm and parameters
- Requirements on the input data
- Roofer CLI App
- API Reference C++
- API Reference Python

roofer - automatic building reconstruction from pointclouds

roofer performs fully automatic LoD2 building reconstruction from a pointcloud and a building roofprint polygon. It was originally developed, and is still used, to create the 3DBAG dataset.

Highlights

- It reconstructs a 3D building model from a pointcloud and a 2D roofprint polygon

Near term roadmap: release v1.0

TODO for roofer v1.0:

- Thoroughly test with next 3DBAG data release
  - fix any bugs we encounter.
- Some more input validation prior to starting processing (CLI app)
- Set up automated build and publishing of binaries (Windows, Linux, macOS)
- Make roofer Python binding **pip**-installable
- Expected this year

Roadmap after v1.0

Usability improvements:

- Implement multithreading for point cloud reading
- More tiling options
- Ability to execute only cropping or only reconstruction, with intermediate data written to disk
- GUI
- CityBuf output

Roadmap after v1.0

Trivial improvements building reconstruction algorithm:

- Improved plane detection (allow assigning points to multiple planes)
  - Improve reconstruction result for low point density scenarios
- Merge adjacent co-planar faces in output mesh
  - Remove unnecessary faces

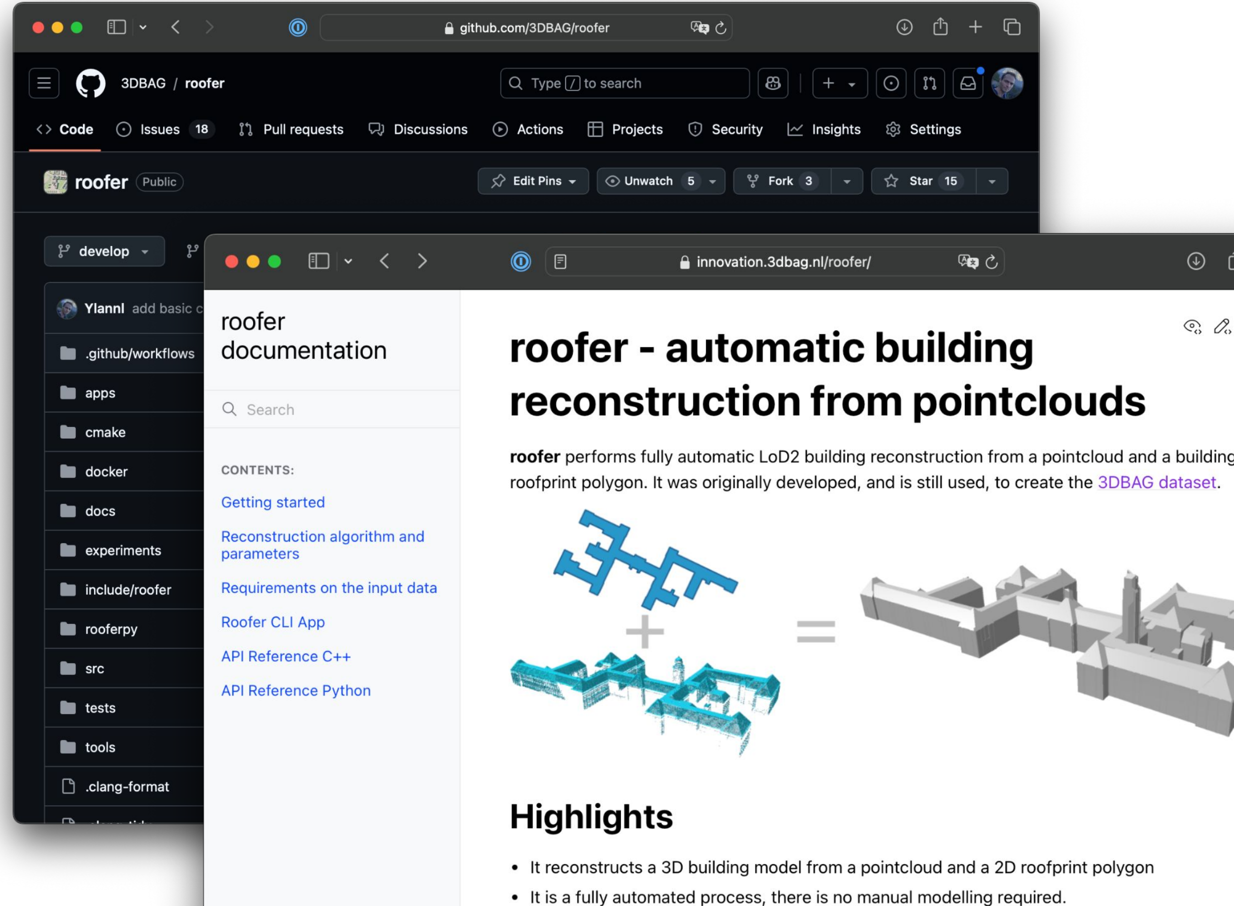
Roadmap after v1.0

Items that require (significant) research and experimentation

- Reduce reliance on accurate building roofprints
  - Rely more on the point cloud data instead
- Incorporate ideas of other building reconstruction methods (eg. City3D, SimpliCity)
  - One goal: Improve regularisation in geometry
- Enhance input elevation data using deep learning (see RoofDiffusion)
  - More robustness against occlusion gaps in pointcloud
  - possibly remove automatic vegetation over building

Try it

3DBAG/roofer



roofer - automatic building reconstruction from pointclouds

roofer performs fully automatic LoD2 building reconstruction from a pointcloud and a building roofprint polygon. It was originally developed, and is still used, to create the 3DBAG dataset.

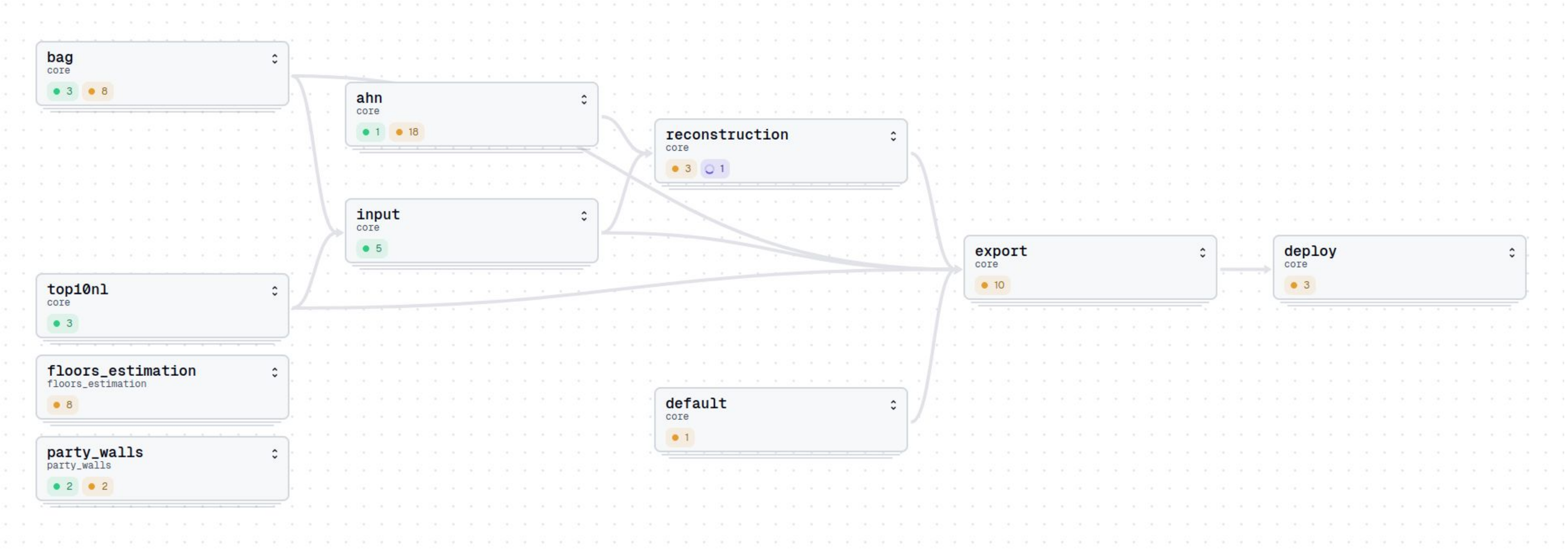
Highlights

- It reconstructs a 3D building model from a pointcloud and a 2D roofprint polygon
- It is a fully automated process, there is no manual modelling required.

QA session

3dbag-pipeline

What is it?

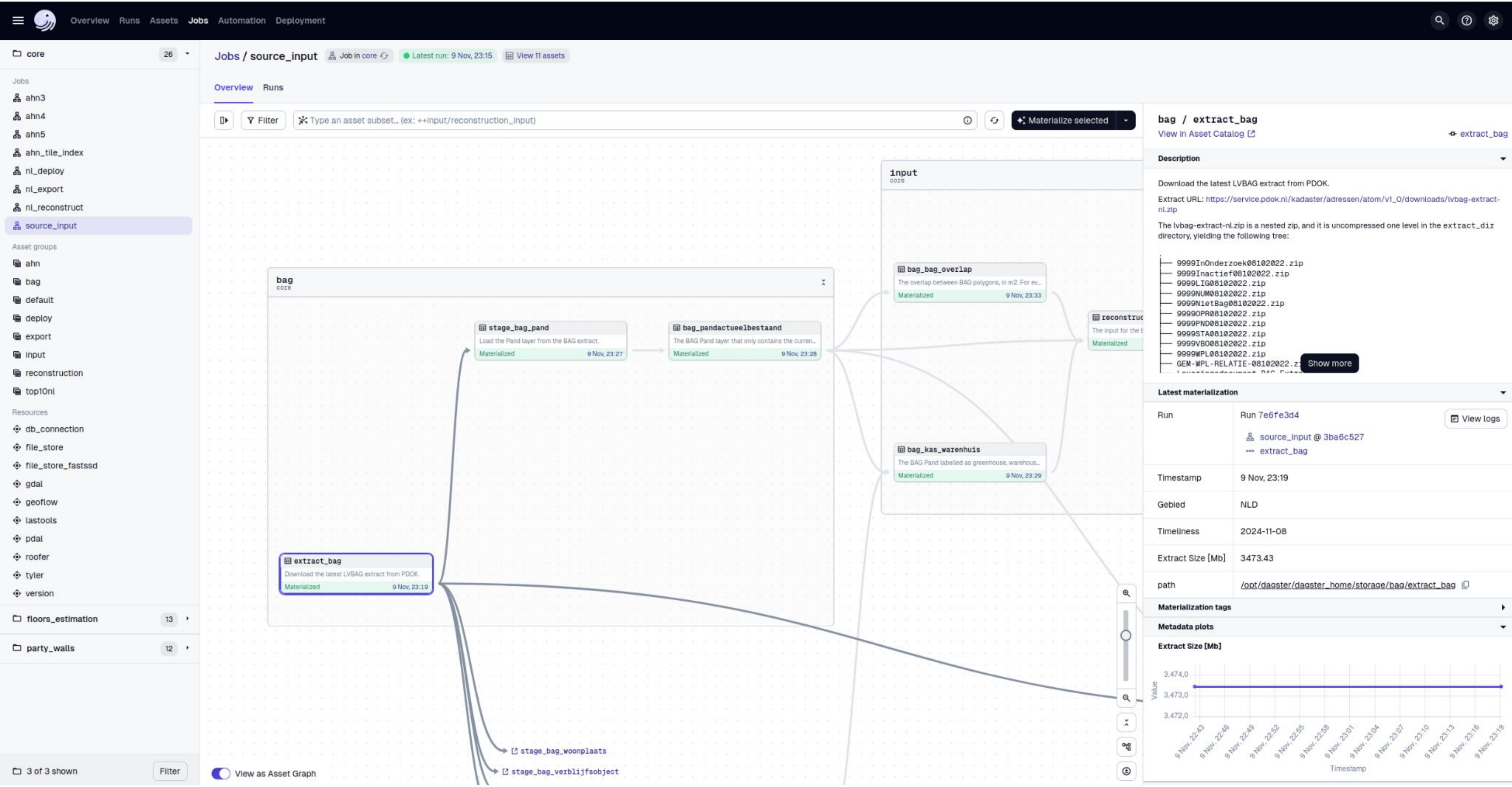

bag
core
● 3
● 8
ahn
core
● 1
● 18
input
core
● 5
reconstruction
core
● 3
○ 1
top10nl
core
● 3
floors_estimation
floors_estimation
● 8
party_walls
party_walls
● 2
● 2
default
core
● 1
export
core
● 10
deploy
core
● 3

Built with Dagster

- Open source data orchestration platform (Apache-2.0)
- Write regular python code + dagster on top
- Concepts:
  - assets
  - jobs
  - resources
  - runs

dagster

https://dagster.io



bag / extract_bag

View in Asset Catalog

extract_bag

Description

Download the latest LVBAG extract from PDOK.

Extract URL: https://service.pdok.nl/kadaster/adressen/atom/v1_0/downloads/lvbag-extract-nl.zip

The lvbag-extract-nl.zip is a nested zip, and it is uncompressed one level in the `extract_dir` directory, yielding the following tree:

```
.
├── 9999InOnderzoek08102022.zip
├── 9999Inactief08102022.zip
├── 9999LIG08102022.zip
├── 9999NUM08102022.zip
├── 9999NietBag08102022.zip
├── 9999OPR08102022.zip
├── 9999PND08102022.zip
├── 9999STA08102022.zip
├── 9999VBO08102022.zip
├── 9999WPL08102022.zip
├── GEM-WPL-RELATIE-08102022.z
```

Show more

Latest materialization

Run	Run 7e6fe3d4 source_input @ 3ba6c527 extract_bag
Timestamp	9 Nov, 23:19
Gebied	NLD
Timeliness	2024-11-08
Extract Size [Mb]	3473.43
path	/opt/dagster/dagster_home/storage/bag/extract_bag

View logs

Materialization tags

Metadata plots

Extract Size [Mb]

Source Data

BAG https://www.kadaster.nl/bag

up-to-date building polygons + attributes

largest extent → roofprint + underground

positional accuracy 30cm

AHN https://ahn.nl

classified point cloud

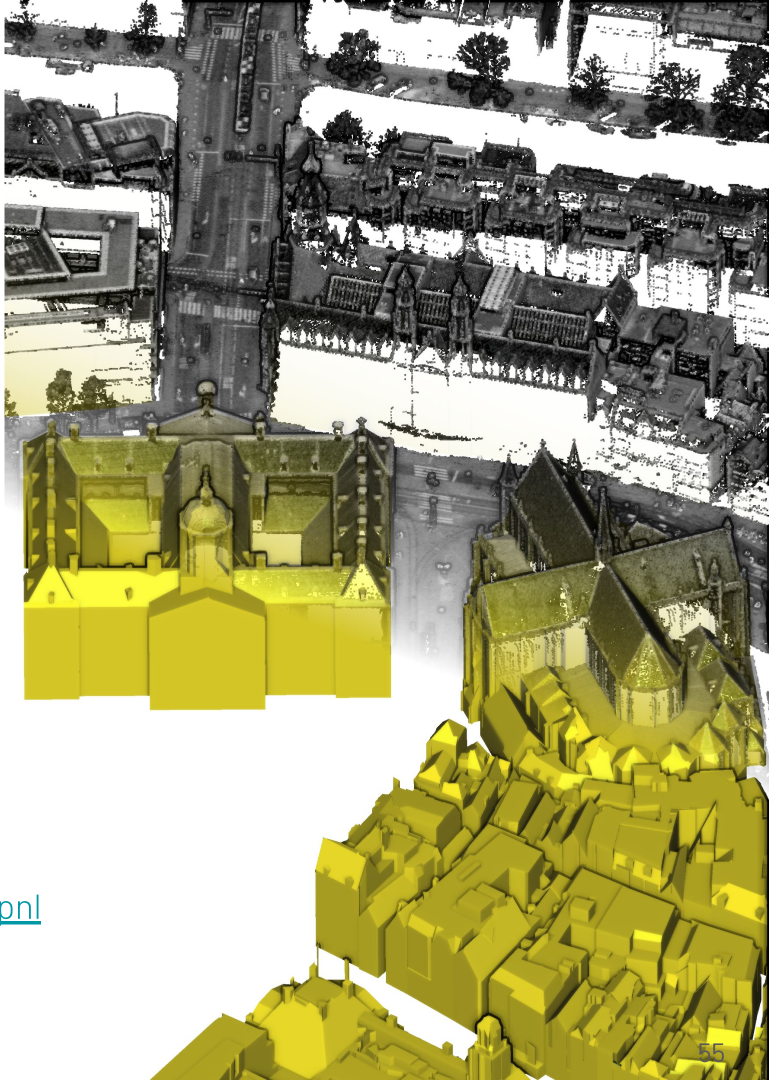
positional accuracy ~23cm (height and planimetric)

8-15 pts/m2 for buildings

occlusion and other no-data areas

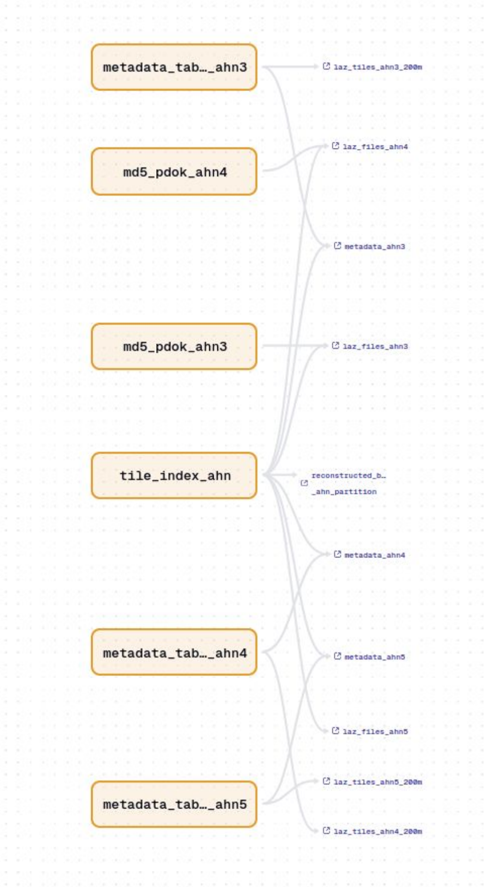
TOP10NL https://www.kadaster.nl/zakelijk/producten/geo-informatie/topnl

topographical map with greenhouse classification



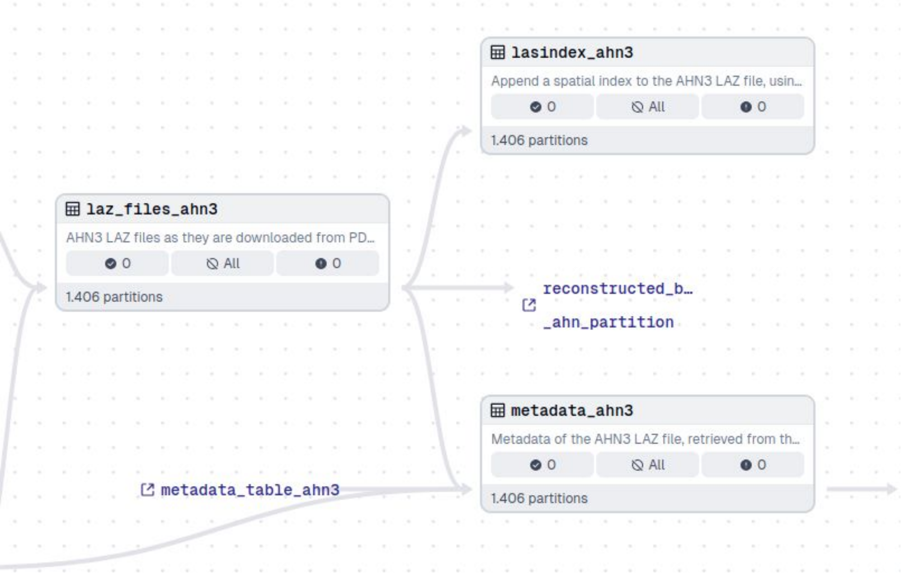
ahn_tile_index

- download AHN tile index
- prepare metadata storage
- download AHN checksums



ahn3/4/5

- download the LAZ files
- index the files with lasindex
- compute metadata
- partitioned by AHN tile



lasindex_ahn3

Append a spatial index to the AHN3 LAZ file, usin...

1.406 partitions

laz_files_ahn3

AHN3 LAZ files as they are downloaded from PD...

1.406 partitions

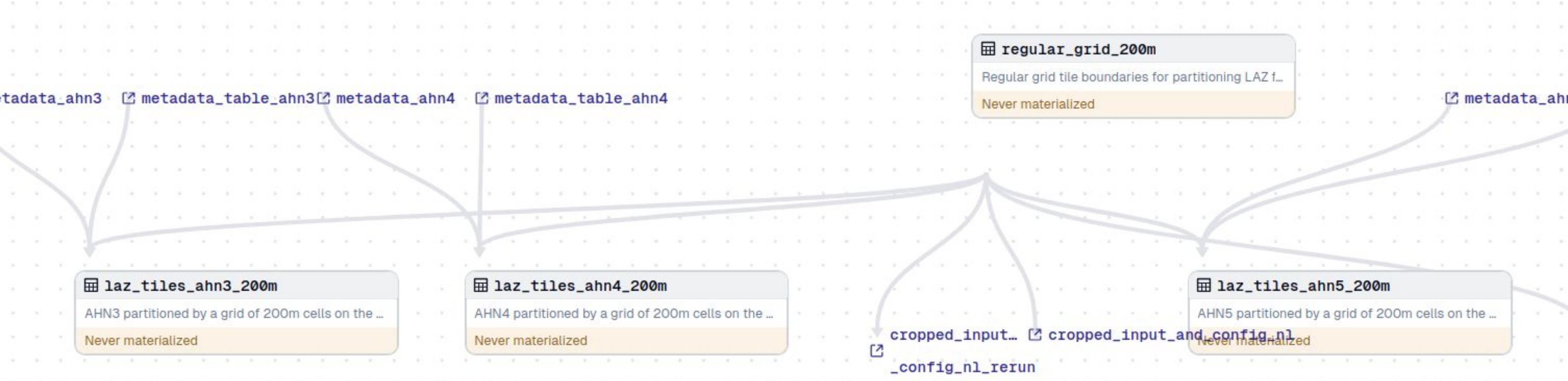
metadata_ahn3

Metadata of the AHN3 LAZ file, retrieved from th...

1.406 partitions

ahn_tiles_200m

- tile up the LAZ files to 200m tiles



regular_grid_200m
Regular grid tile boundaries for partitioning LAZ f...
Never materialized

tadata_ahn3

metadata_table_ahn3

metadata_ahn4

metadata_table_ahn4

metadata_ahn

laz_tiles_ahn3_200m
AHN3 partitioned by a grid of 200m cells on the ...
Never materialized

laz_tiles_ahn4_200m
AHN4 partitioned by a grid of 200m cells on the ...
Never materialized

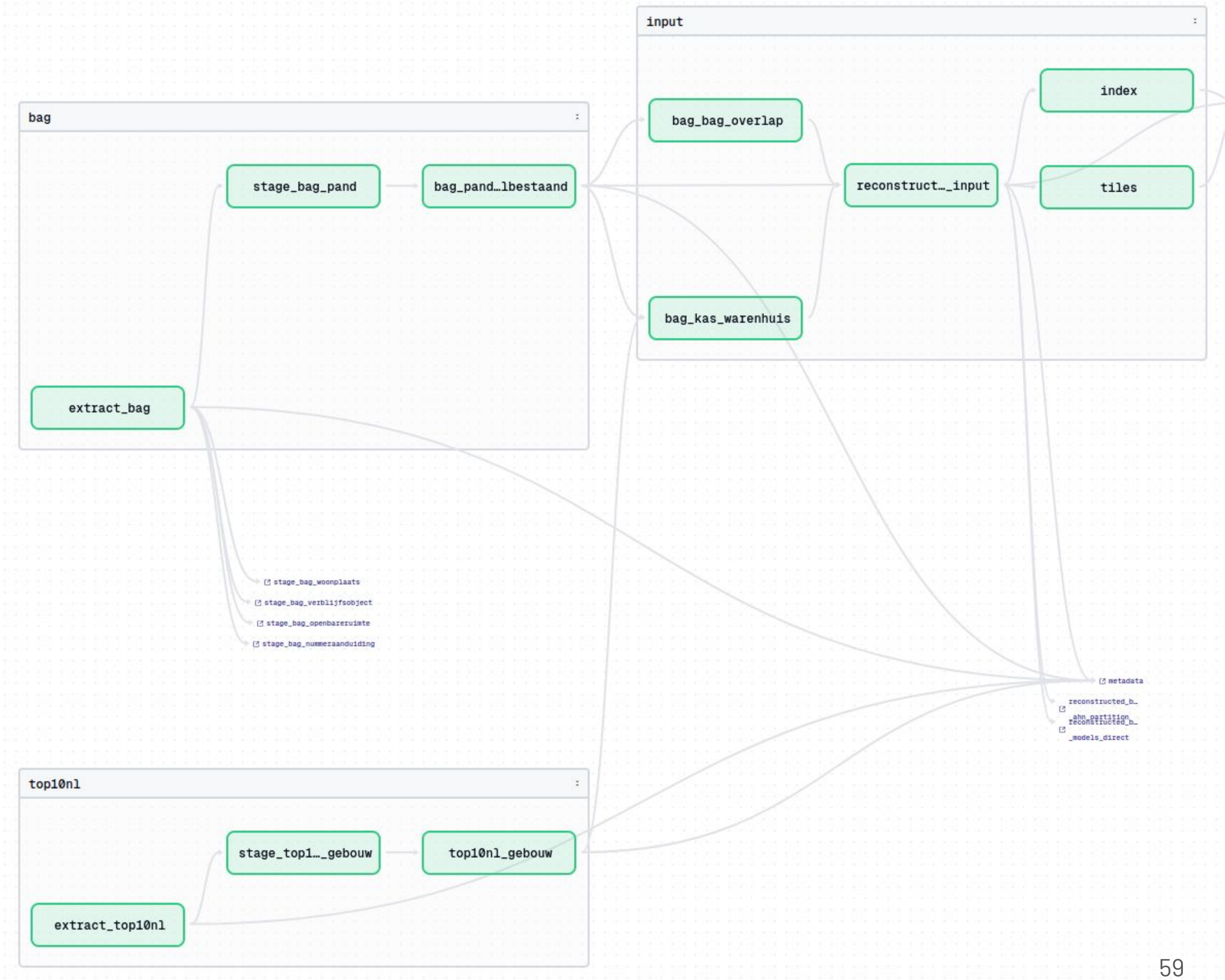
cropped_input…
_config_nl_rerun

cropped_input_and_config_nl

laz_tiles_ahn5_200m
AHN5 partitioned by a grid of 200m cells on the ...
Never materialized

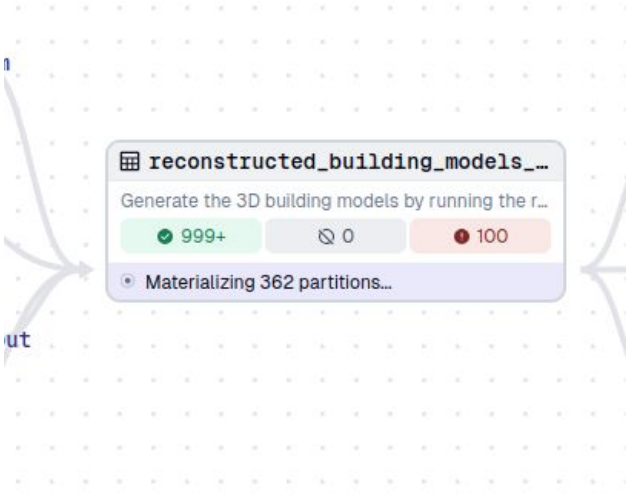
source_input

- 2D vector sources
- download, extract
- stage and transform in PostgreSQL
- combine input table
- tiling



nl_reconstruct

- run reconstruction
- partitioned by 2D input tile



reconstructed_building_models_...

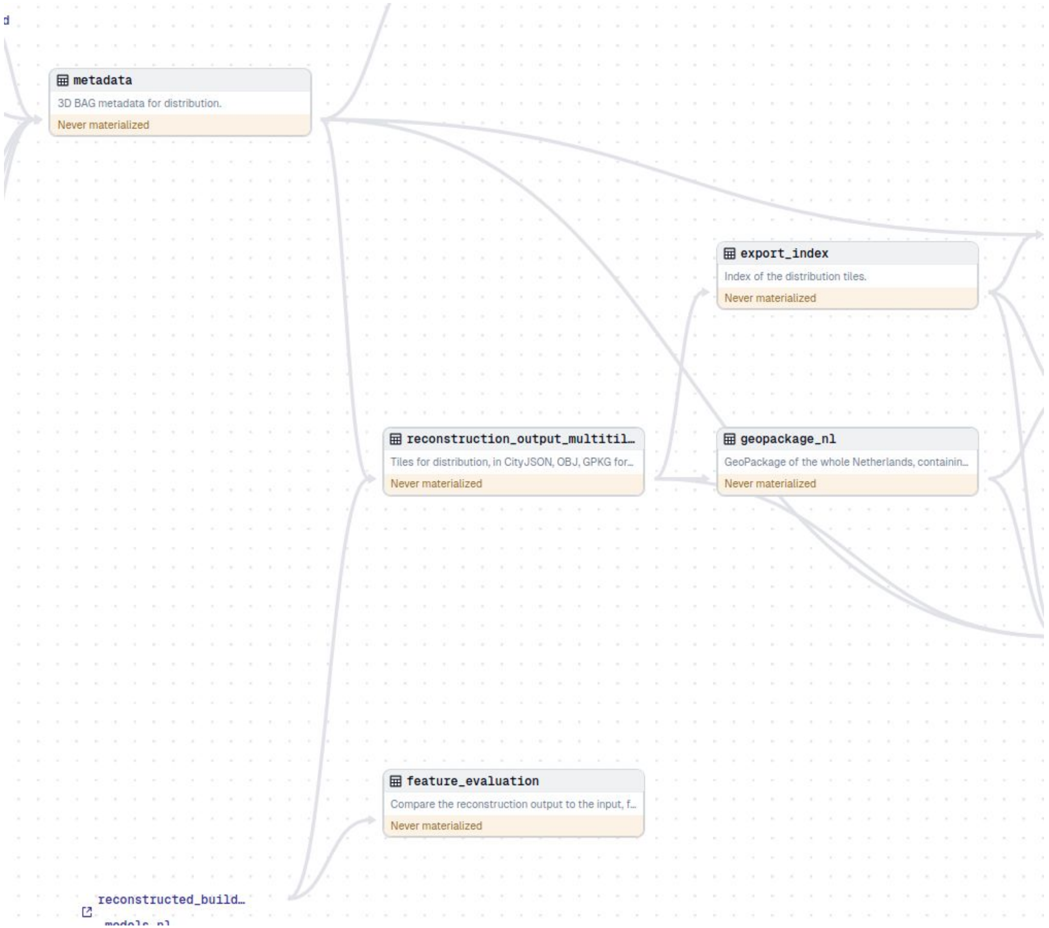
Generate the 3D building models by running the r...

✓ 999+ | ⃠ 0 | ● 100

Materializing 362 partitions...

nl_export

- format conversion
- compile metadata
- calculate reconstruction rate



metadata
3D BAG metadata for distribution.
Never materialized

export_index
Index of the distribution tiles.
Never materialized

reconstruction_output_multitil…
Tiles for distribution, in CityJSON, OBJ, GPKG for…
Never materialized

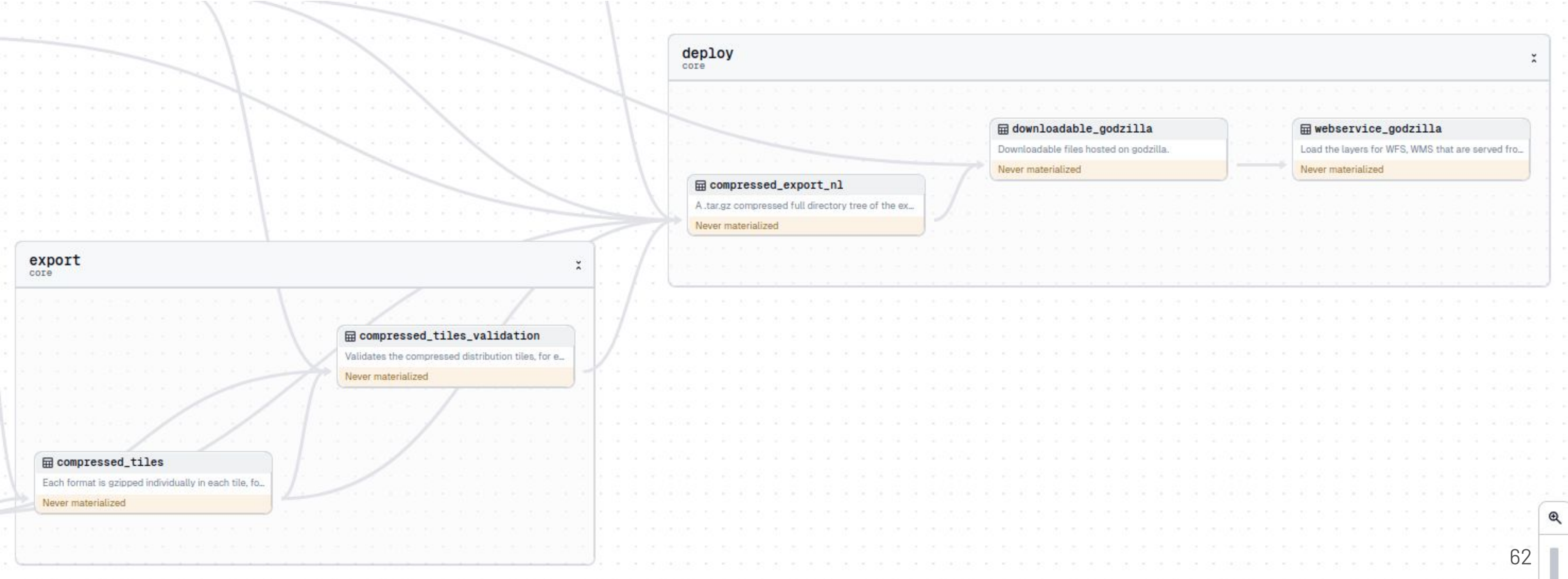
geopackage_nl
GeoPackage of the whole Netherlands, containin…
Never materialized

feature_evaluation
Compare the reconstruction output to the input, f…
Never materialized

reconstructed_build…
models_nl

nl_deploy

- compress and validate
- deploy to webserver



deploy
core

compressed_export_nl
A .tar.gz compressed full directory tree of the ex…
Never materialized

downloadable_godzilla
Downloadable files hosted on godzilla.
Never materialized

webservice_godzilla
Load the layers for WFS, WMS that are served fro…
Never materialized

export
core

compressed_tiles_validation
Validates the compressed distribution tiles, for e…
Never materialized

compressed_tiles
Each format is gzipped individually in each tile, fo…
Never materialized

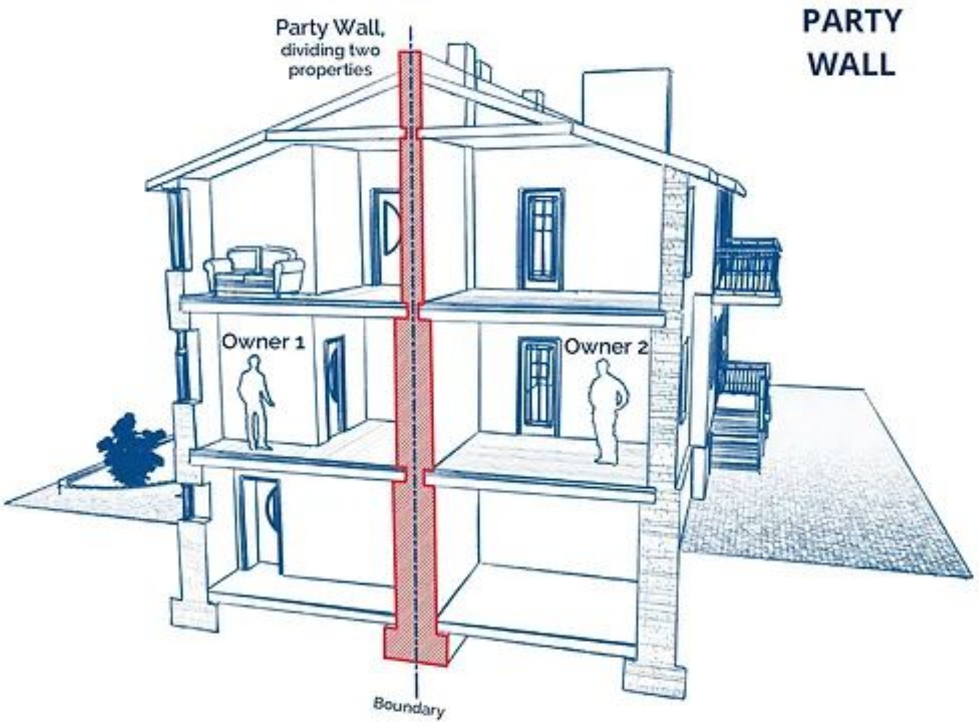
Extensions

floors_estimation

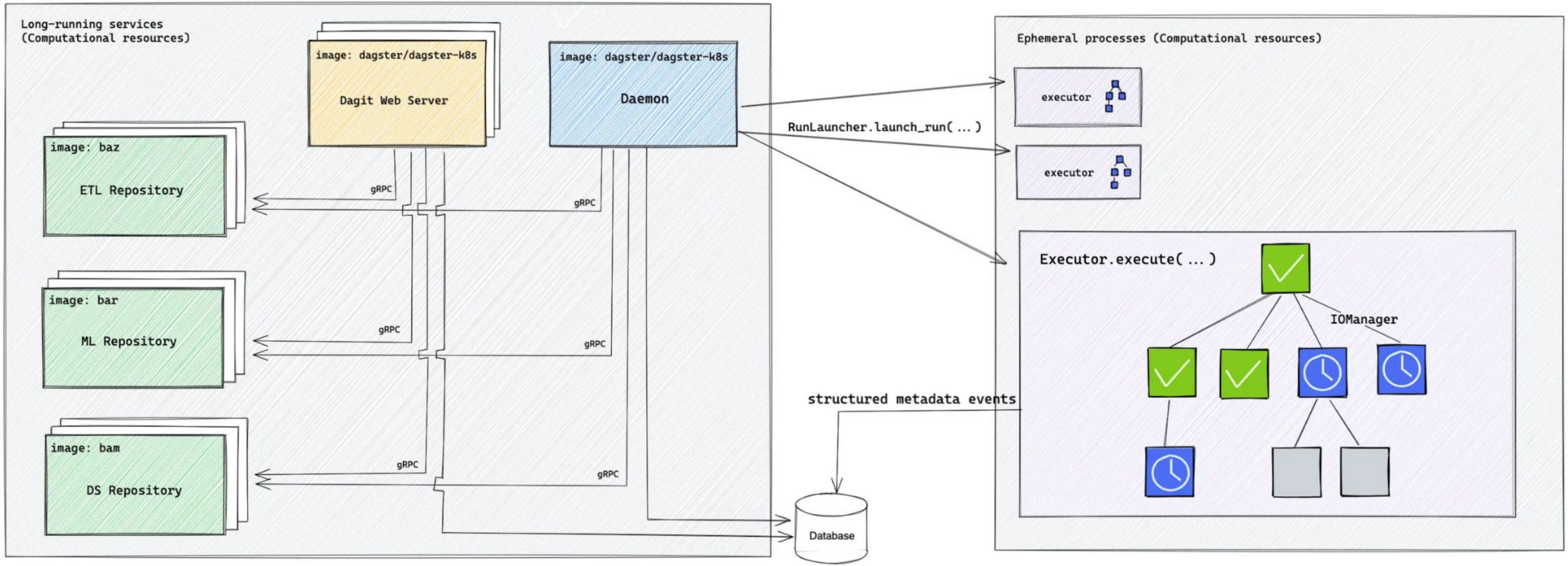
- Estimate the nr. of floors
- Accurate up to 5 floors
  - Not enough training data for tall buildings (random forest regression)
- Computed from CityJSON Features

party_walls

- Compute the area of surfaces, incl. shared walls (party walls)
- Computed from CityJSON tiles



Architecture overview (Dagster)



https://docs.dagster.io/deployment/overview

Architecture overview

Mono-repo with ***subpackages:***

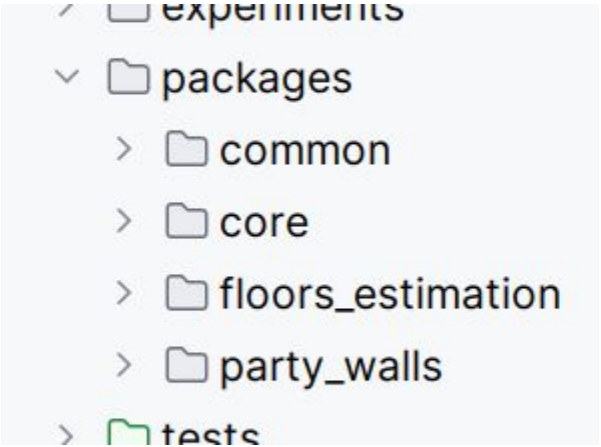
`common` package for functionality that is used by all workflow packages

Workflow packages contain the pipeline logic:

`core` The core 3DBAG data

`party_walls` Party walls calculation

`floors_estimation` Nr. of floors estimation

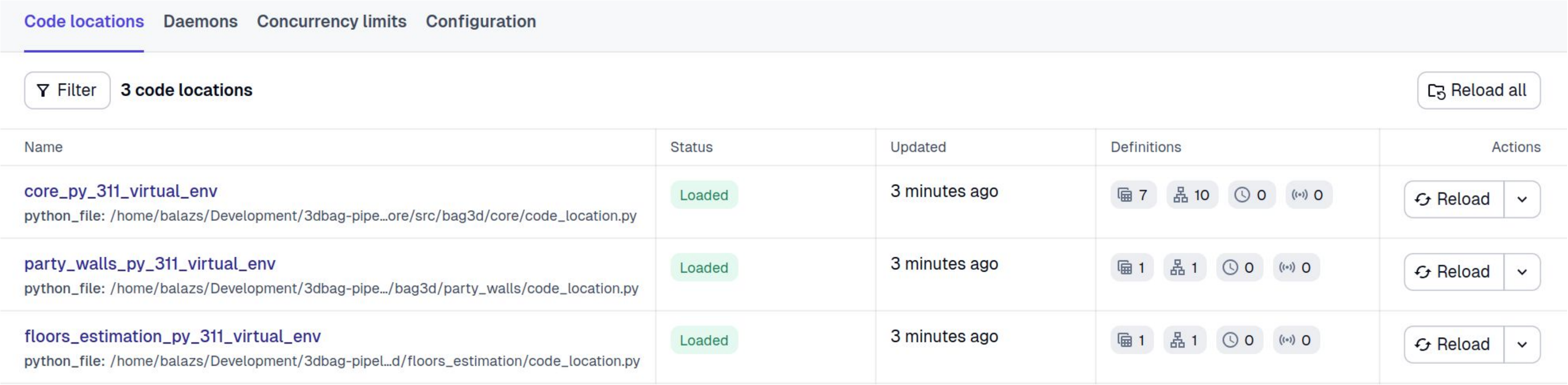


```
  experiments
∨ packages
    › common
    › core
    › floors_estimation
    › party_walls
› tests
```

Architecture overview

Each workflow package in an isolated virtual environment ("Code Location").

- + Helps us avoid dependency hell
- + Dagster uses gRPC for communicating with the code locations → cannot pass execution results



3 code locations

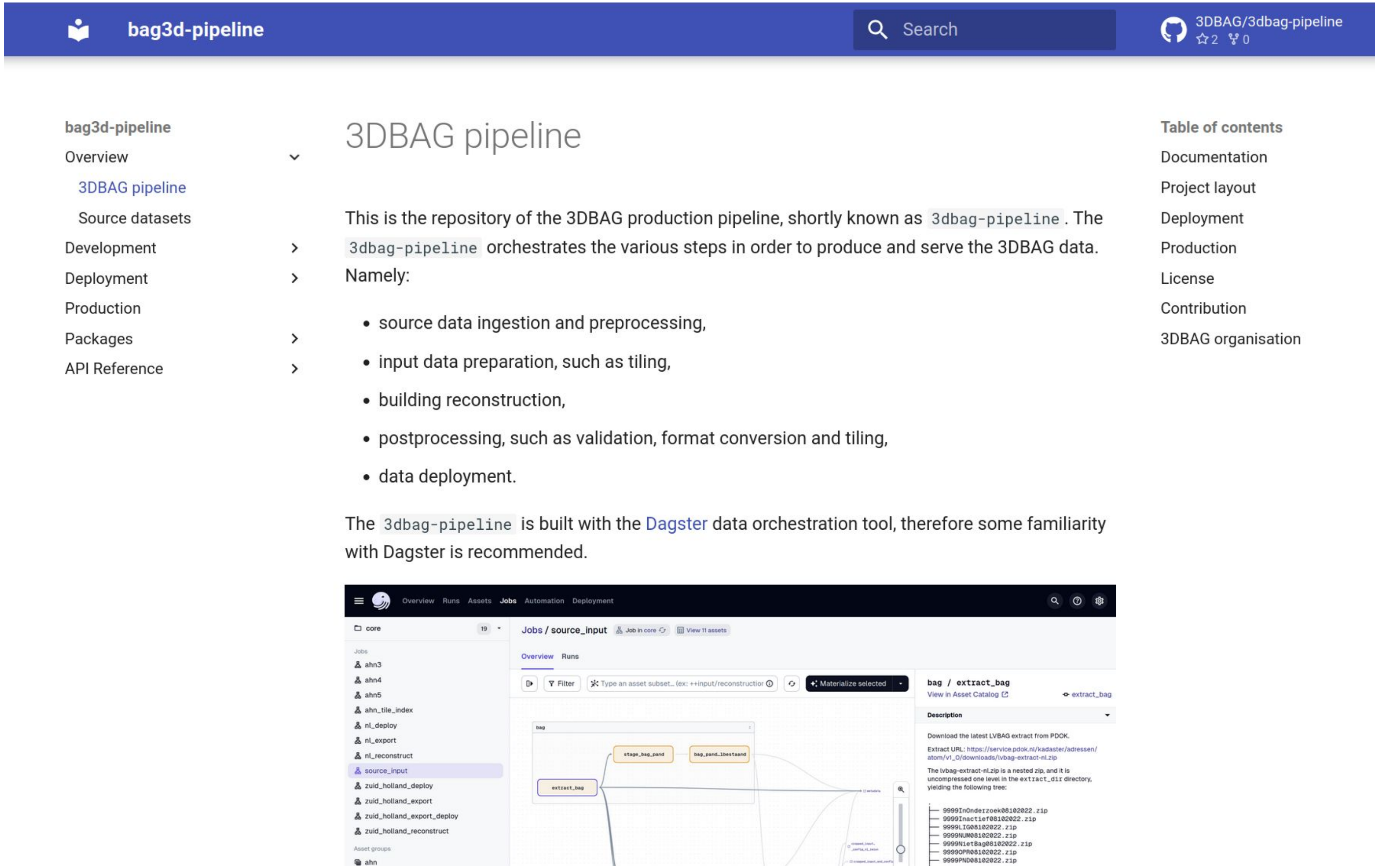
Name	Status	Updated	Definitions	Actions
core_py_311_virtual_env python_file: /home/balazs/Development/3dbag-pipe...ore/src/bag3d/core/code_location.py	Loaded	3 minutes ago	7 10 0 0	Reload
party_walls_py_311_virtual_env python_file: /home/balazs/Development/3dbag-pipe.../bag3d/party_walls/code_location.py	Loaded	3 minutes ago	1 1 0 0	Reload
floors_estimation_py_311_virtual_env python_file: /home/balazs/Development/3dbag-pipel...d/floors_estimation/code_location.py	Loaded	3 minutes ago	1 1 0 0	Reload

Deployment

- use docker
  - multi-container setup
  - all tools built into the base image
  - each package in its own image
- compose file in repo
  - used for testing on GH Actions
  - also for local dev

- native setup also possible, but complicated

Documentation



bag3d-pipeline

3DBAG/3dbag-pipeline ☆2 ⑂0

3DBAG pipeline

This is the repository of the 3DBAG production pipeline, shortly known as `3dbag-pipeline`. The `3dbag-pipeline` orchestrates the various steps in order to produce and serve the 3DBAG data. Namely:

- source data ingestion and preprocessing,
- input data preparation, such as tiling,
- building reconstruction,
- postprocessing, such as validation, format conversion and tiling,
- data deployment.

The `3dbag-pipeline` is built with the Dagster data orchestration tool, therefore some familiarity with Dagster is recommended.

bag / extract_bag

View in Asset Catalog

extract_bag

Description

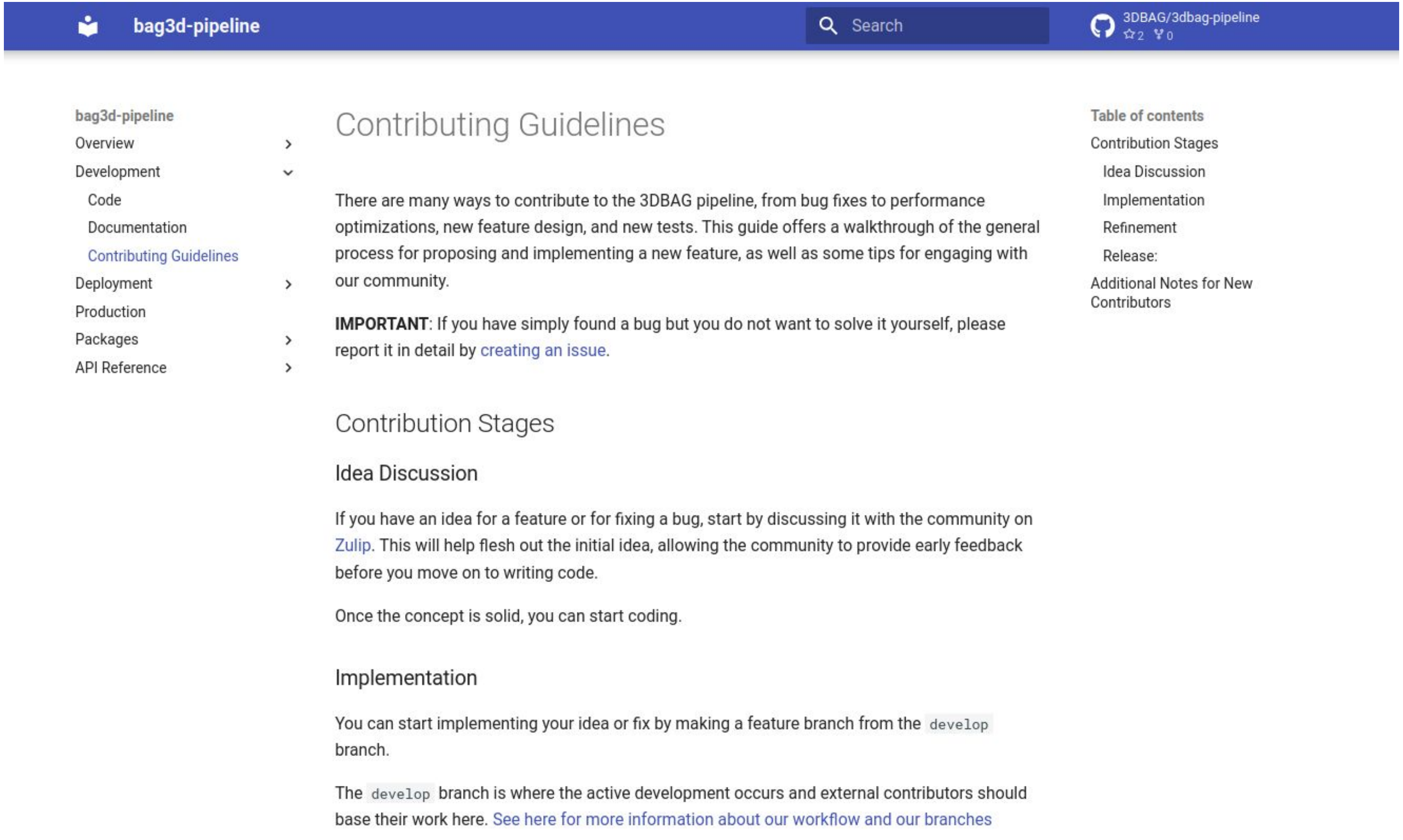
Download the latest LVBAG extract from PDOK.

Extract URL: https://service.pdok.nl/kadaster/adressen/atom/v1_0/downloads/lvbag-extract-nl.zip

The lvbag-extract-nl.zip is a nested zip, and it is uncompressed one level in the `extract_dir` directory, yielding the following tree:

```
.
├── 9999InOnderzoek08102022.zip
├── 9999Inactief08102022.zip
├── 9999LIG08102022.zip
├── 9999NUM08102022.zip
├── 9999NietBag08102022.zip
├── 9999OPR08102022.zip
├── 9999PND08102022.zip
```

Contribution guidelines



Contributing Guidelines

There are many ways to contribute to the 3DBAG pipeline, from bug fixes to performance optimizations, new feature design, and new tests. This guide offers a walkthrough of the general process for proposing and implementing a new feature, as well as some tips for engaging with our community.

IMPORTANT: If you have simply found a bug but you do not want to solve it yourself, please report it in detail by creating an issue.

Contribution Stages

Idea Discussion

If you have an idea for a feature or for fixing a bug, start by discussing it with the community on Zulip. This will help flesh out the initial idea, allowing the community to provide early feedback before you move on to writing code.

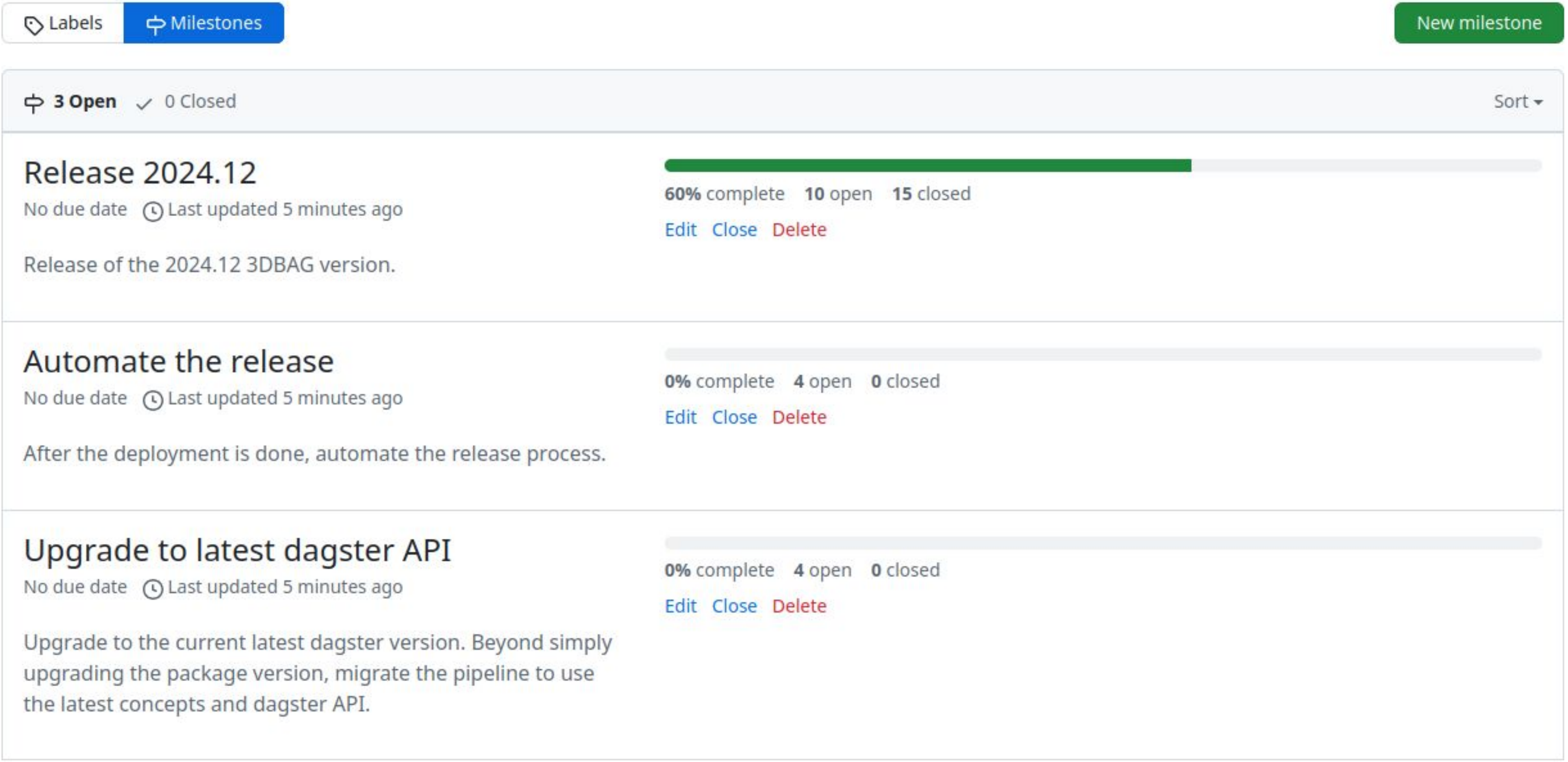
Once the concept is solid, you can start coding.

Implementation

You can start implementing your idea or fix by making a feature branch from the `develop` branch.

The `develop` branch is where the active development occurs and external contributors should base their work here. See here for more information about our workflow and our branches

Roadmap



3 Open ✓ 0 Closed

Release 2024.12

No due date Last updated 5 minutes ago

60% complete 10 open 15 closed

Release of the 2024.12 3DBAG version.

Automate the release

No due date Last updated 5 minutes ago

0% complete 4 open 0 closed

After the deployment is done, automate the release process.

Upgrade to latest dagster API

No due date Last updated 5 minutes ago

0% complete 4 open 0 closed

Upgrade to the current latest dagster version. Beyond simply upgrading the package version, migrate the pipeline to use the latest concepts and dagster API.

Roadmap - 2024.12 3DBAG release

- Integrate the new Roofer
- Integrate AHN5
- Update to the latest Dagster API
- Improve developer experience
  - various levels of testing
  - CI
  - deployment
  - guidelines and tutorials
- Improve automation

QA session

Time for a 5’ break!

We will resume at 14:35

Next up: Data Access

Part II – Data access

CityJSON

How to get the 3DBAG data?

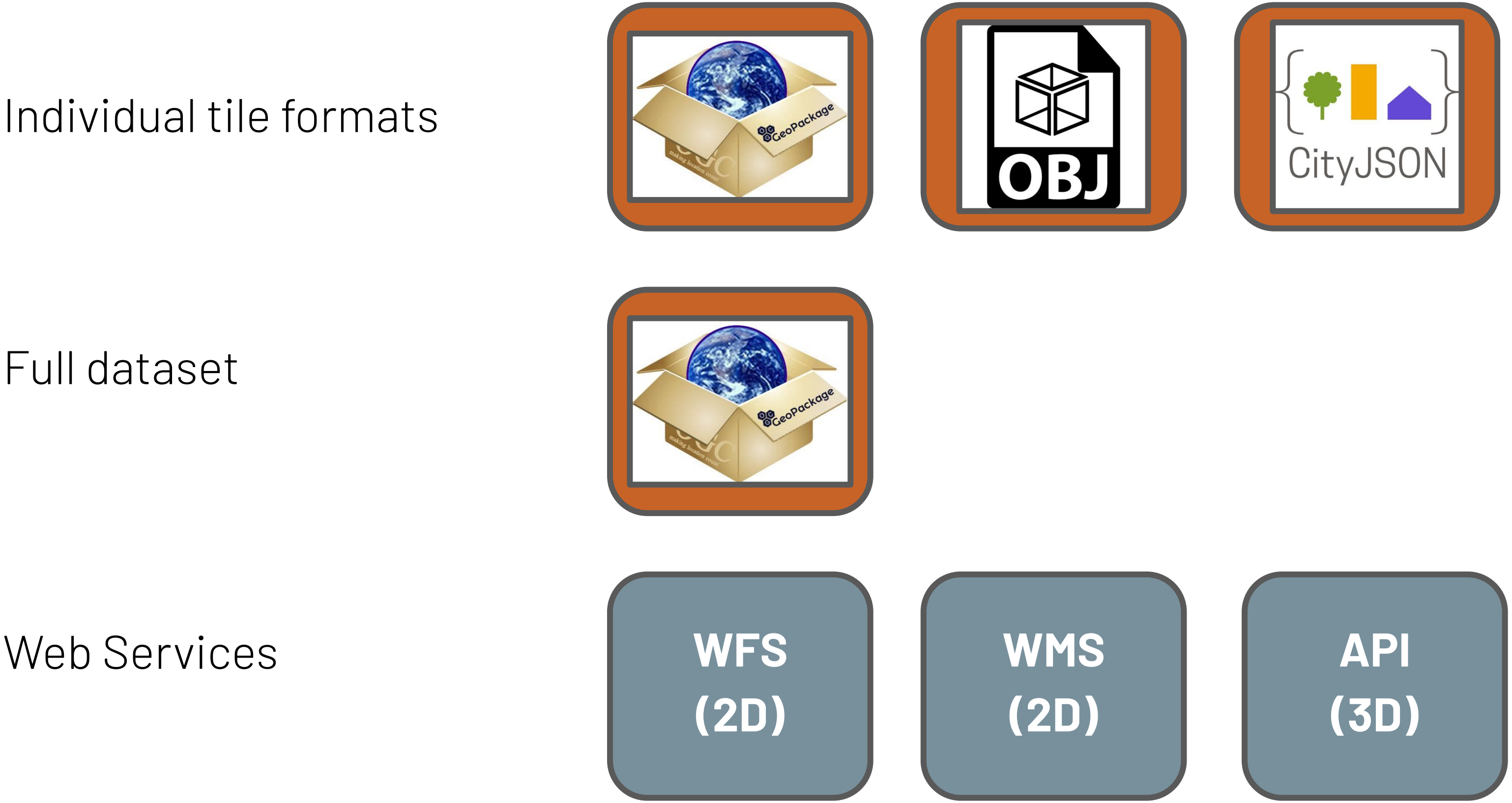
Attributes

b3_azimut
b3_bag_bag_overlap
b3_bouwlagen
b3_dak_type
b3_dd_id
b3_h_dak_50p
b3_h_dak_70p
b3_h_dak_max
b3_h_dak_min
b3_h_maaiveld
b3_hellingshoek
b3_kas_warenhuis
b3_kwaliteitsindicator
b3_mutatie_ahn3_ahn4
b3_nodata_fractie_ahn3
b3_nodata_fractie_ahn4
b3_nodata_radius_ahn3
b3_nodata_radius_ahn4
b3_pand_deel_id

b3_opp_buitenmuur
b3_opp_dak_plat
b3_opp_dak_schuin
b3_opp_grond
b3_opp_scheidingsmuur
b3_puntdichtheid_ahn3
b3_puntdichtheid_ahn4
b3_pw_bron
b3_pw_datum
b3_pw_selectie_reden
b3_reconstructie_onvolledig
b3_rmse_lod12
b3_rmse_lod13
b3_rmse_lod22
b3_val3dity_lod12
b3_val3dity_lod22
b3_volume_lod12
b3_volume_lod13
b3_volume_lod22

begingeldigheid
documentdatum
documentnummer
eindgeldigheid
eindregistratie
geconstateerd
identificatie
labels
oorspronkelijkbouwjaar
status
tijdstipeindregistratielv
tijdstipinactief
tijdstipinactieflv
tijdstipnietbaglv
tijdstipregistratie
tijdstipregistratielv
voorkomenidentificatie

File formats & Services



Individual tile formats

Full dataset

Web Services

WFS (2D)

WMS (2D)

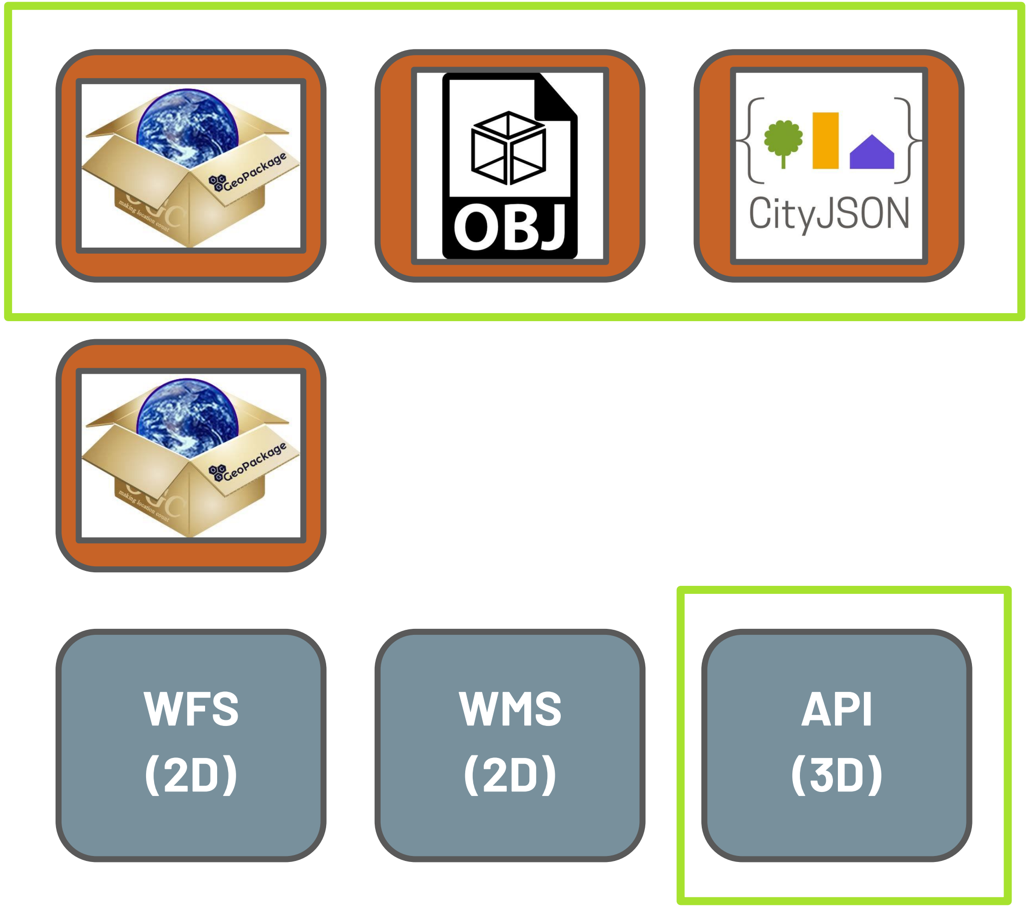
API (3D)

File formats & Services

Individual tile formats

Full dataset

Web Services

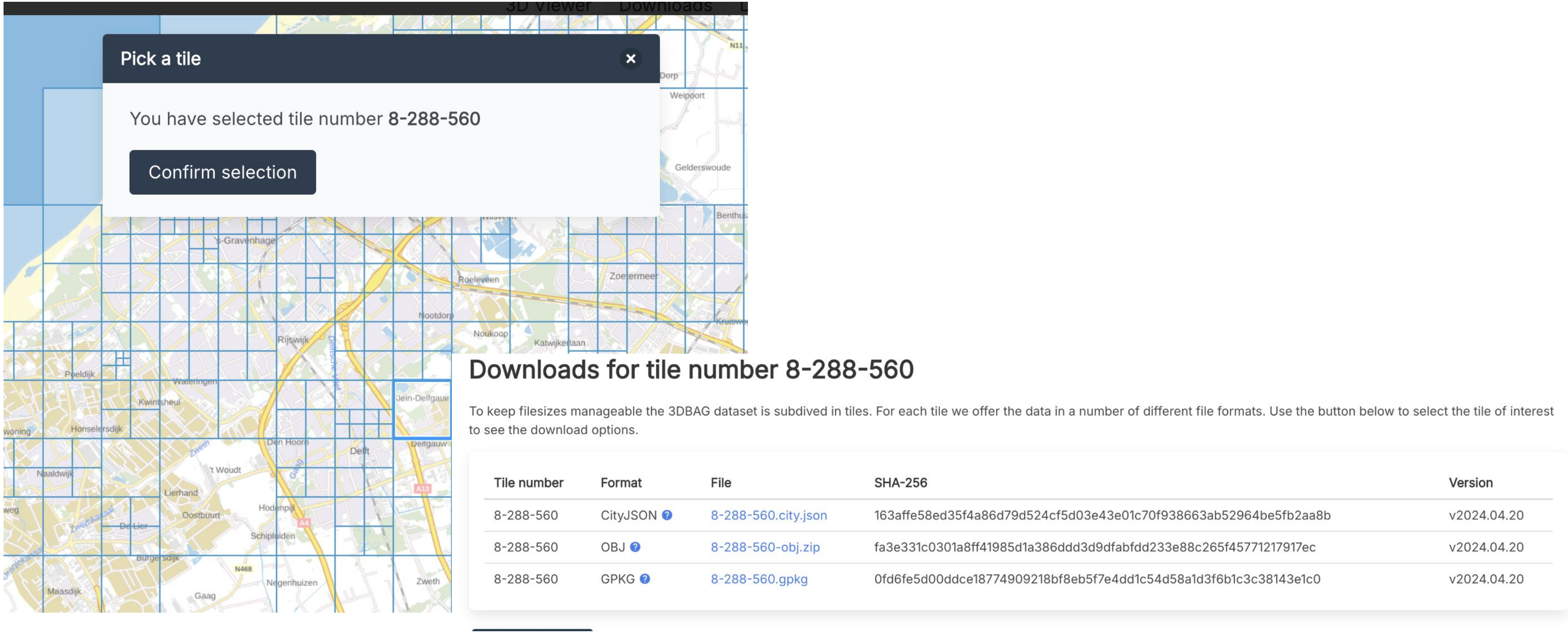


WFS (2D)

WMS (2D)

API (3D)

How-To: Downloading 3DBAG tiles



Pick a tile ×

You have selected tile number **8-288-560**

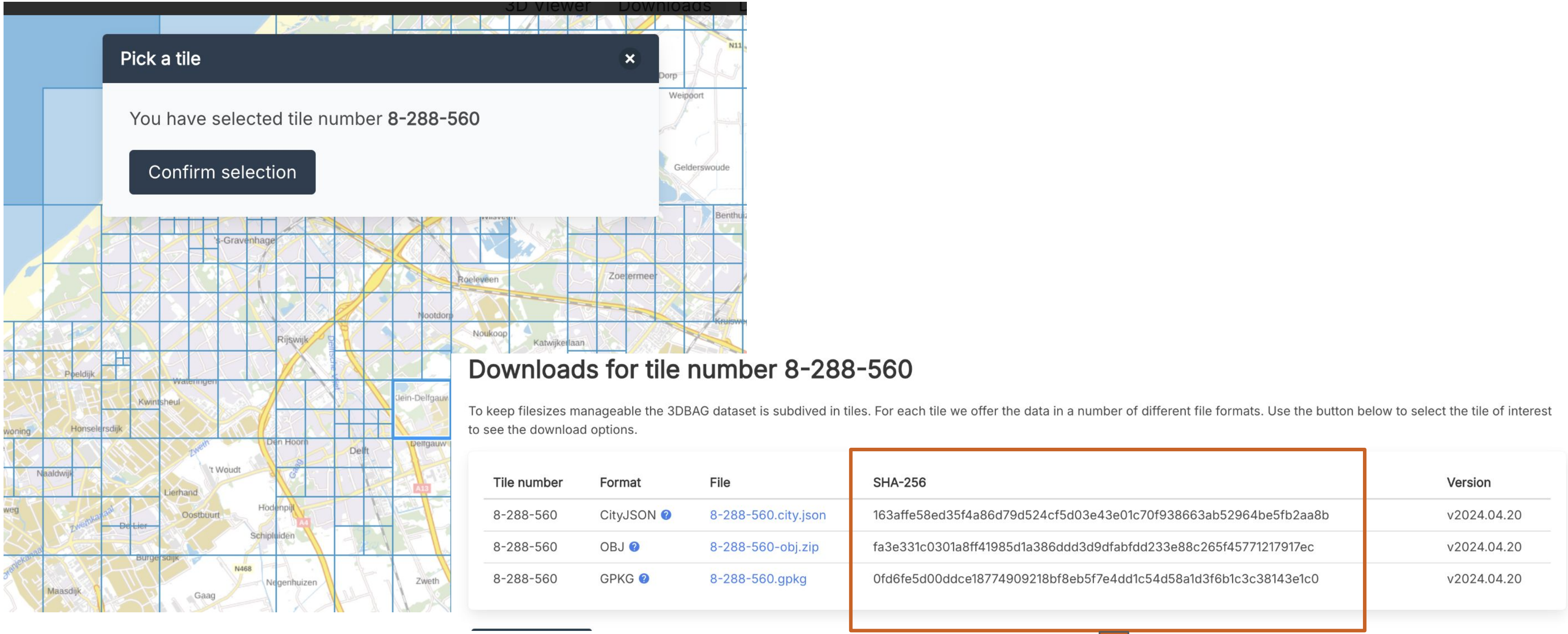
Confirm selection

Downloads for tile number 8-288-560

To keep filesizes manageable the 3DBAG dataset is subdived in tiles. For each tile we offer the data in a number of different file formats. Use the button below to select the tile of interest to see the download options.

Tile number	Format	File	SHA-256	Version
8-288-560	CityJSON ?	8-288-560.city.json	163affe58ed35f4a86d79d524cf5d03e43e01c70f938663ab52964be5fb2aa8b	v2024.04.20
8-288-560	OBJ ?	8-288-560-obj.zip	fa3e331c0301a8ff41985d1a386ddd3d9dfabfdd233e88c265f45771217917ec	v2024.04.20
8-288-560	GPKG ?	8-288-560.gpkg	0fd6fe5d00ddce18774909218bf8eb5f7e4dd1c54d58a1d3f6b1c3c38143e1c0	v2024.04.20

How-To: Downloading 3DBAG tiles



Pick a tile ×

You have selected tile number **8-288-560**

Confirm selection

Downloads for tile number 8-288-560

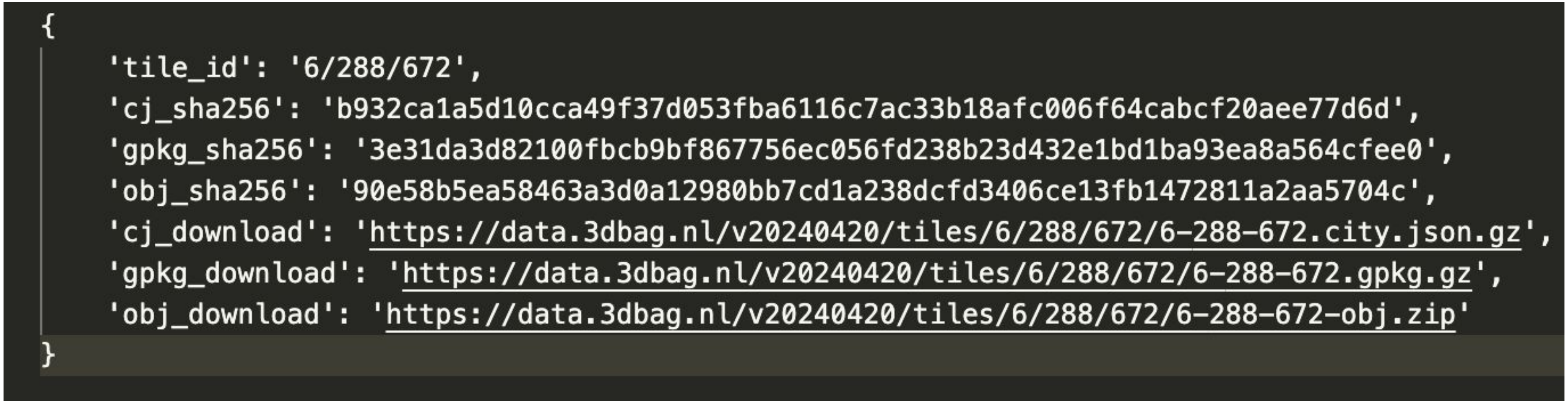
To keep filesizes manageable the 3DBAG dataset is subdived in tiles. For each tile we offer the data in a number of different file formats. Use the button below to select the tile of interest to see the download options.

Tile number	Format	File	SHA-256	Version
8-288-560	CityJSON ?	8-288-560.city.json	163affe58ed35f4a86d79d524cf5d03e43e01c70f938663ab52964be5fb2aa8b	v2024.04.20
8-288-560	OBJ ?	8-288-560-obj.zip	fa3e331c0301a8ff41985d1a386ddd3d9dfabfdd233e88c265f45771217917ec	v2024.04.20
8-288-560	GPKG ?	8-288-560.gpkg	0fd6fe5d00ddce18774909218bf8eb5f7e4dd1c54d58a1d3f6b1c3c38143e1c0	v2024.04.20

Integrity Verification

How-To: Downloading 3DBAG tiles

All tiles within a bbox: Use the complete tile index (FlatGeoBuf)

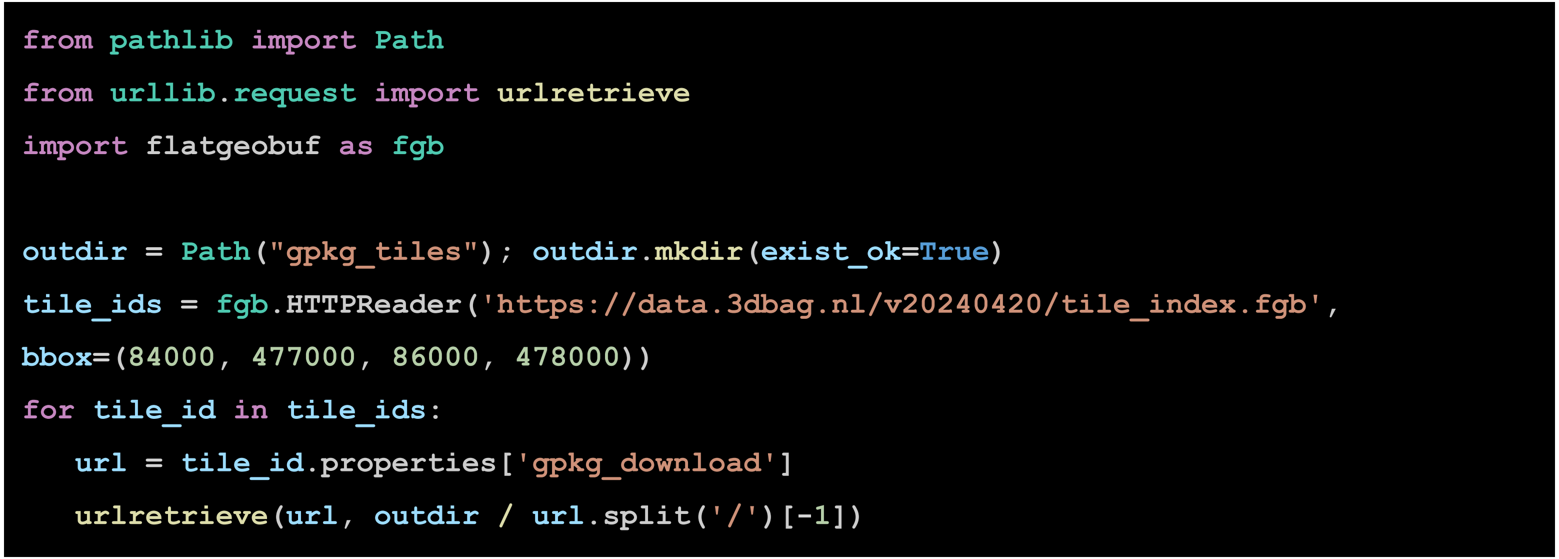


```
{
    'tile_id': '6/288/672',
    'cj_sha256': 'b932ca1a5d10cca49f37d053fba6116c7ac33b18afc006f64cabcf20aee77d6d',
    'gpkg_sha256': '3e31da3d82100fbcb9bf867756ec056fd238b23d432e1bd1ba93ea8a564cfee0',
    'obj_sha256': '90e58b5ea58463a3d0a12980bb7cd1a238dcfd3406ce13fb1472811a2aa5704c',
    'cj_download': 'https://data.3dbag.nl/v20240420/tiles/6/288/672/6-288-672.city.json.gz',
    'gpkg_download': 'https://data.3dbag.nl/v20240420/tiles/6/288/672/6-288-672.gpkg.gz',
    'obj_download': 'https://data.3dbag.nl/v20240420/tiles/6/288/672/6-288-672-obj.zip'
}
```

How-To: Downloading 3DBAG tiles

All tiles within a bbox: Use the complete tile index (FlatGeoBuf)

Python snippet:



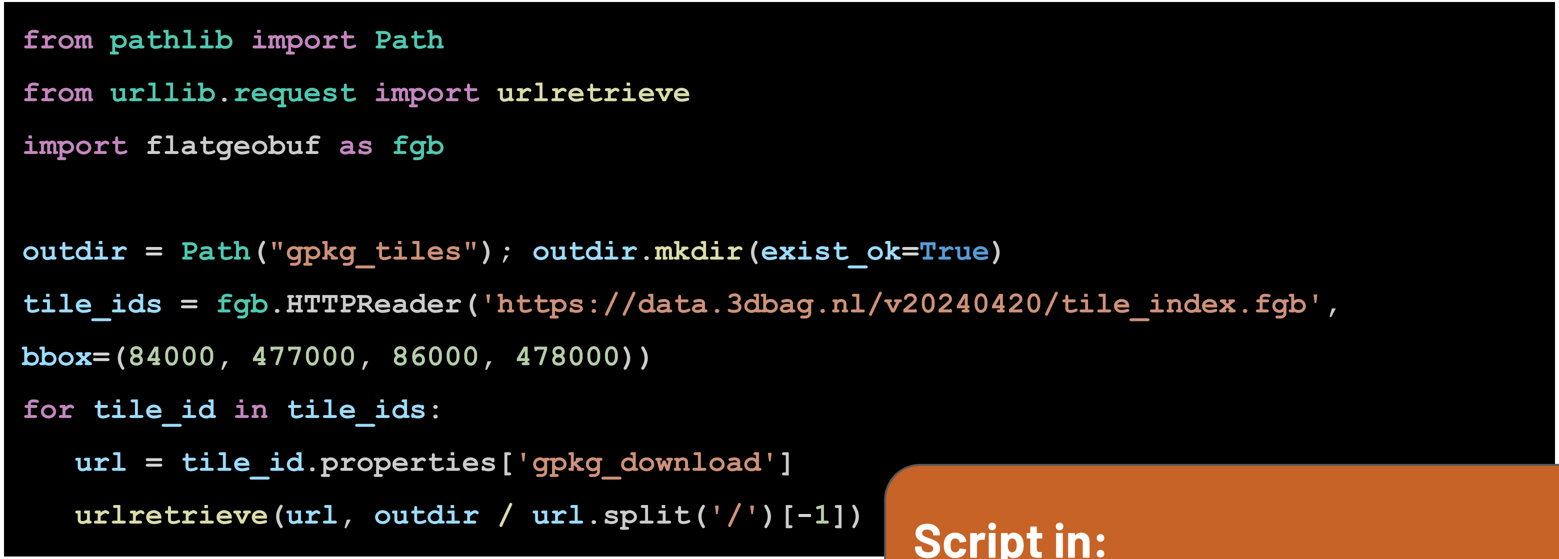
```python
from pathlib import Path
from urllib.request import urlretrieve
import flatgeobuf as fgb

outdir = Path("gpkg_tiles"); outdir.mkdir(exist_ok=True)
tile_ids = fgb.HTTPReader('https://data.3dbag.nl/v20240420/tile_index.fgb',
bbox=(84000, 477000, 86000, 478000))
for tile_id in tile_ids:
   url = tile_id.properties['gpkg_download']
   urlretrieve(url, outdir / url.split('/')[-1])
```

How-To: Downloading 3DBAG tiles

All tiles within a bbox: Use the complete tile index (FlatGeoBuf)

Python snippet:



```python
from pathlib import Path
from urllib.request import urlretrieve
import flatgeobuf as fgb

outdir = Path("gpkg_tiles"); outdir.mkdir(exist_ok=True)
tile_ids = fgb.HTTPReader('https://data.3dbag.nl/v20240420/tile_index.fgb',
bbox=(84000, 477000, 86000, 478000))
for tile_id in tile_ids:
   url = tile_id.properties['gpkg_download']
   urlretrieve(url, outdir / url.split('/')[-1])
```

Script in:

github.com/3DBAG/3dbag-scripts

How-To: 3DBAG-API

- Programmatic access to the 3DBAG data
- Returns cityJSONSeq
- Allows the retrieval of specific buildings and bbox operations
- Not yet OGC-compliant
- Only supported CRS: Amersfoort / RD New + NAP height (EPSG:7415)

https://api.3dbag.nl/

How-To: 3DBAG-API

Request Single Building:

BAG identifier

https://api.3dbag.nl/collections/pand/items/NL.IMBAG.Pand.0503100000032914

Request All Buildings in a BBox:

xmin ymin xmax ymax

https://api.3dbag.nl/collections/pand/items?bbox=85000.00,446700.03,85300.00,447000.00

Instructions in:

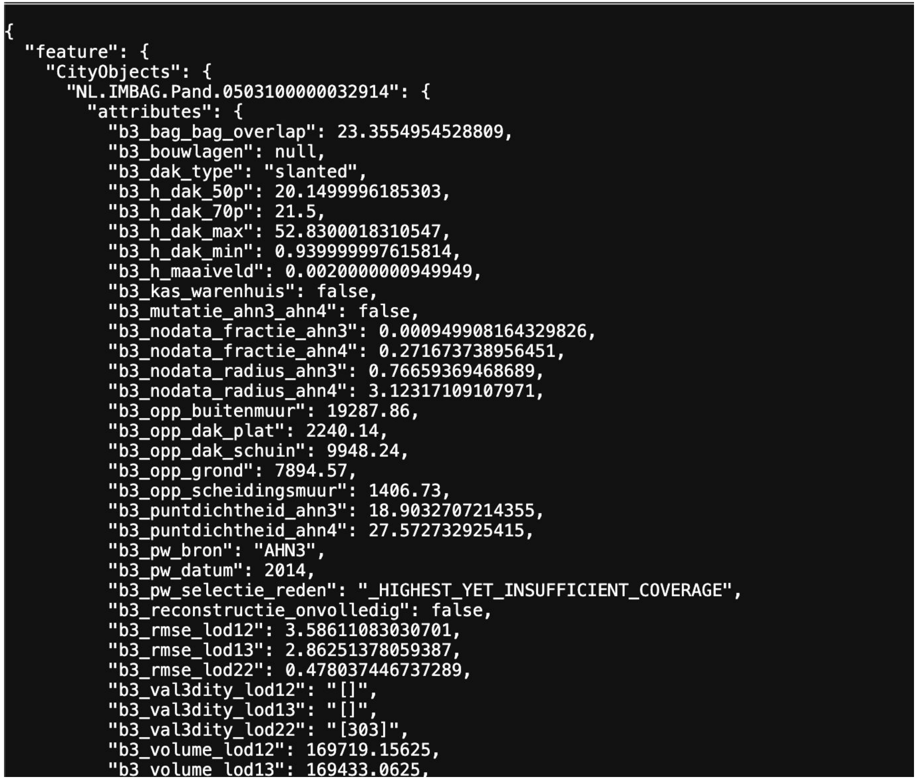
api.3dbag.nl/api.html

How-To: 3DBAG-API

CityJSONSeq

Request Single Building:

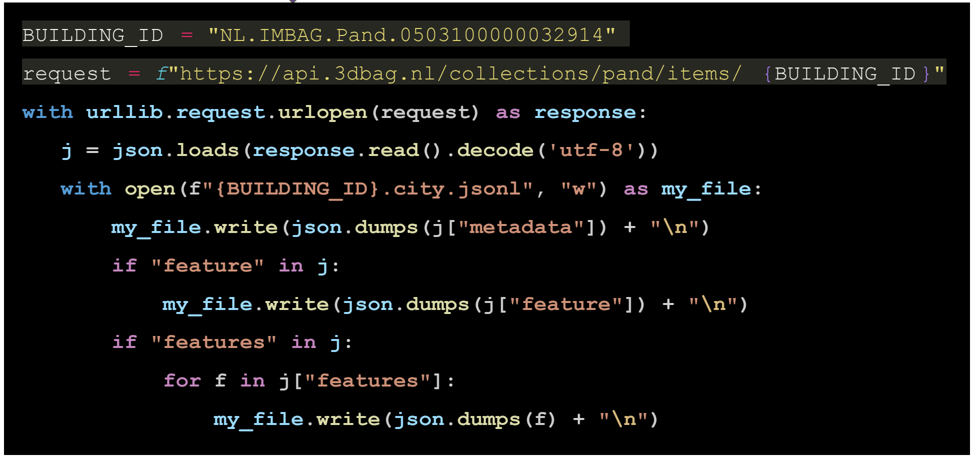
https://api.3dbag.nl/collections/pand/items/NL.IMBAG.Pand.0503100000032914



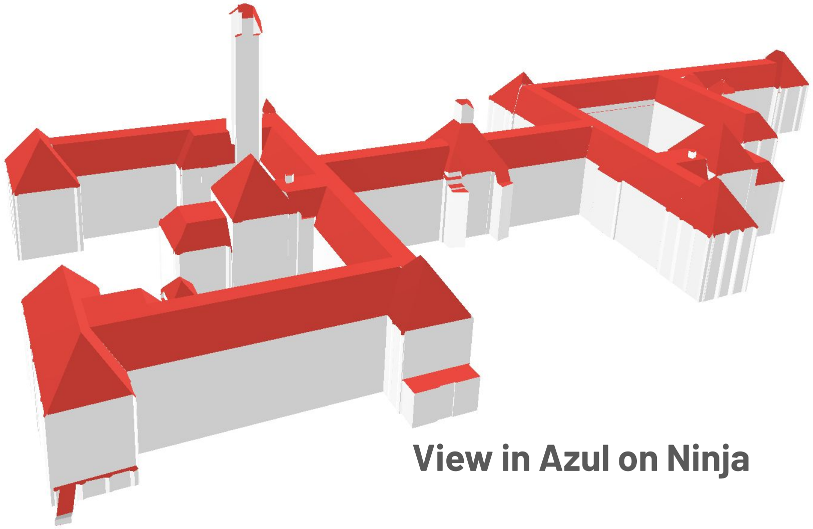
```
{
  "feature": {
    "CityObjects": {
      "NL.IMBAG.Pand.0503100000032914": {
        "attributes": {
          "b3_bag_bag_overlap": 23.3554954528809,
          "b3_bouwlagen": null,
          "b3_dak_type": "slanted",
          "b3_h_dak_50p": 20.1499996185303,
          "b3_h_dak_70p": 21.5,
          "b3_h_dak_max": 52.8300018310547,
          "b3_h_dak_min": 0.939999997615814,
          "b3_h_maaiveld": 0.0020000000949949,
          "b3_kas_warenhuis": false,
          "b3_mutatie_ahn3_ahn4": false,
          "b3_nodata_fractie_ahn3": 0.000949908164329826,
          "b3_nodata_fractie_ahn4": 0.271673738956451,
          "b3_nodata_radius_ahn3": 0.76659369468689,
          "b3_nodata_radius_ahn4": 3.12317109107971,
          "b3_opp_buitenmuur": 19287.86,
          "b3_opp_dak_plat": 2240.14,
          "b3_opp_dak_schuin": 9948.24,
          "b3_opp_grond": 7894.57,
          "b3_opp_scheidingsmuur": 1406.73,
          "b3_puntdichtheid_ahn3": 18.9032707214355,
          "b3_puntdichtheid_ahn4": 27.572732925415,
          "b3_pw_bron": "AHN3",
          "b3_pw_datum": 2014,
          "b3_pw_selectie_reden": "_HIGHEST_YET_INSUFFICIENT_COVERAGE",
          "b3_reconstructie_onvolledig": false,
          "b3_rmse_lod12": 3.58611083030701,
          "b3_rmse_lod13": 2.86251378059387,
          "b3_rmse_lod22": 0.478037446737289,
          "b3_val3dity_lod12": "[]",
          "b3_val3dity_lod13": "[]",
          "b3_val3dity_lod22": "[303]",
          "b3_volume_lod12": 169719.15625,
          "b3_volume_lod13": 169433.0625,
```

How-To: 3DBAG-API

Request Single Building:
https://api.3dbag.nl/collections/pand/items/NL.IMBAG.Pand.0503100000032914



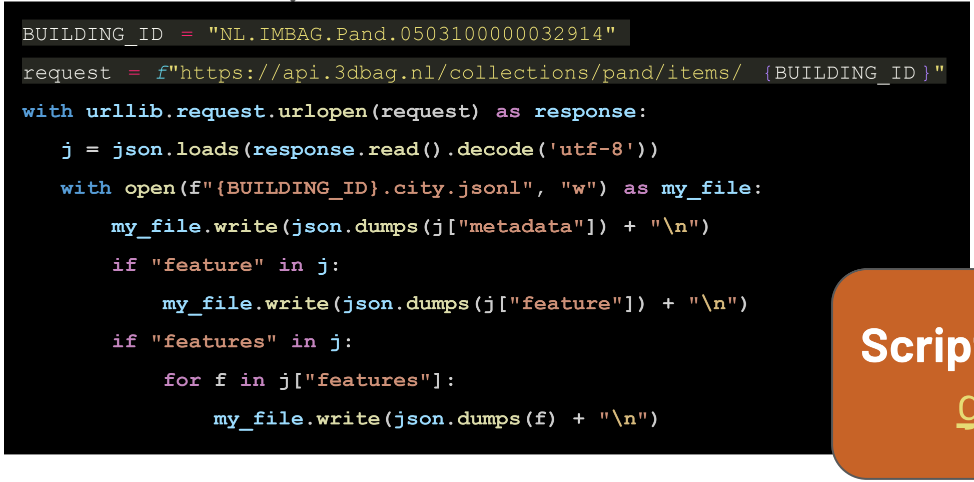
```
BUILDING_ID = "NL.IMBAG.Pand.0503100000032914"
request = f"https://api.3dbag.nl/collections/pand/items/{BUILDING_ID}"
with urllib.request.urlopen(request) as response:
   j = json.loads(response.read().decode('utf-8'))
   with open(f"{BUILDING_ID}.city.jsonl", "w") as my_file:
       my_file.write(json.dumps(j["metadata"]) + "\n")
       if "feature" in j:
           my_file.write(json.dumps(j["feature"]) + "\n")
       if "features" in j:
           for f in j["features"]:
               my_file.write(json.dumps(f) + "\n")
```



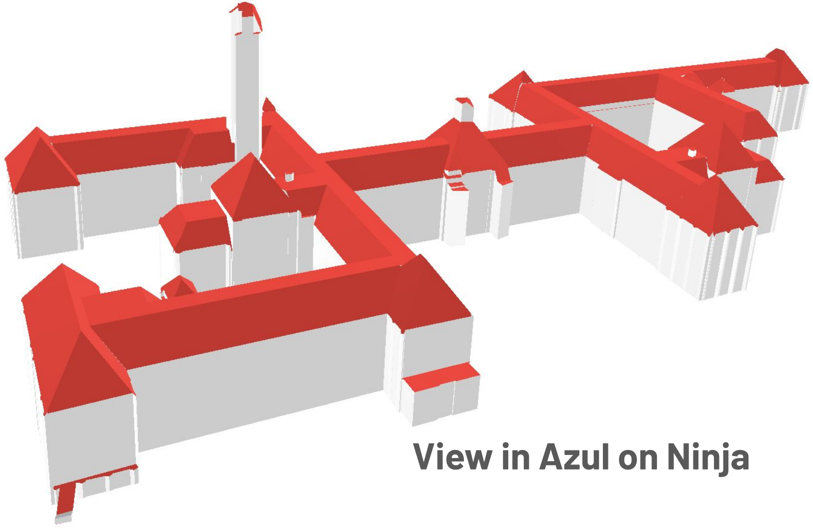
View in Azul on Ninja

How-To: 3DBAG-API

Request Single Building:
https://api.3dbag.nl/collections/pand/items/NL.IMBAG.Pand.0503100000032914



```
BUILDING_ID = "NL.IMBAG.Pand.0503100000032914"
request = f"https://api.3dbag.nl/collections/pand/items/{BUILDING_ID}"
with urllib.request.urlopen(request) as response:
    j = json.loads(response.read().decode('utf-8'))
    with open(f"{BUILDING_ID}.city.jsonl", "w") as my_file:
        my_file.write(json.dumps(j["metadata"]) + "\n")
        if "feature" in j:
            my_file.write(json.dumps(j["feature"]) + "\n")
        if "features" in j:
            for f in j["features"]:
                my_file.write(json.dumps(f) + "\n")
```

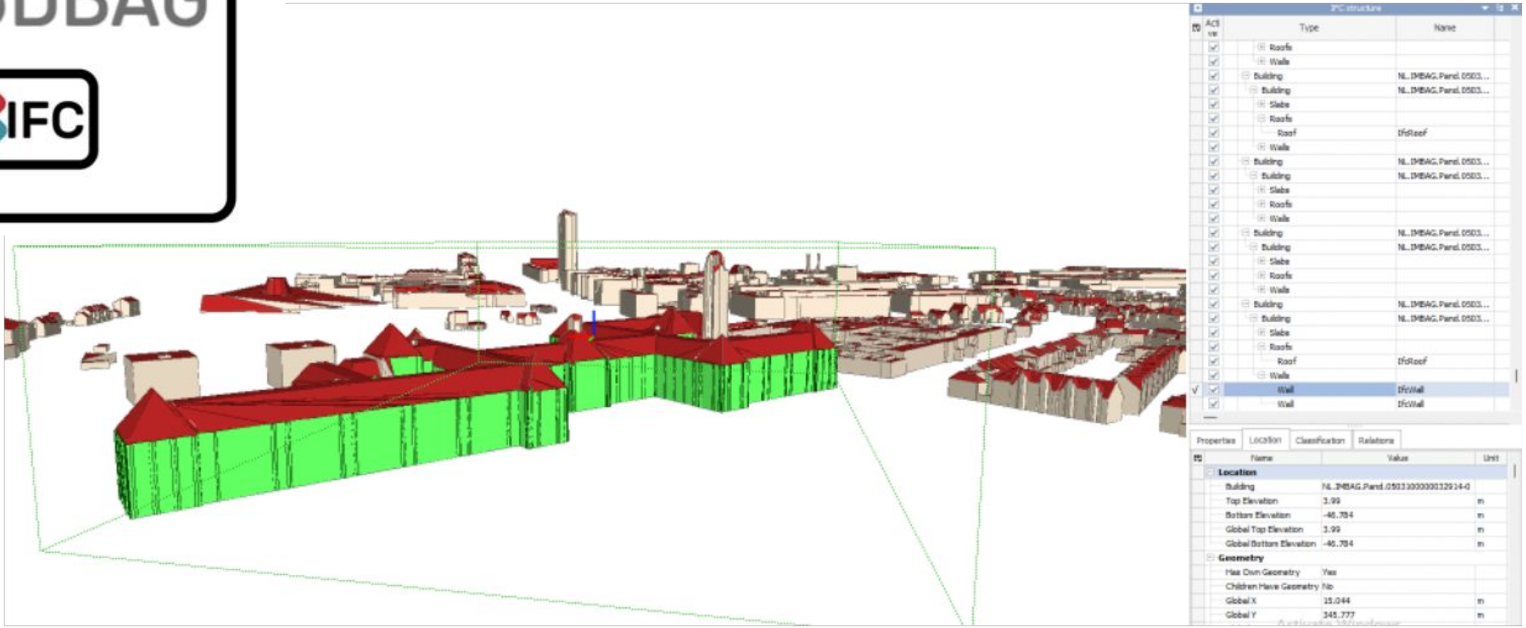


View in Azul on Ninja

Script in:
github.com/3DBAG/3dbag-scripts

New file format: IFC (soon)

Converter: cityjson2ifc



Acti ve	Type	Name
☑	Roofs	
☑	Walls	
☑	Building	NL.IMBAG.Pand.0503...
☑	Building	NL.IMBAG.Pand.0503...
☑	Slabs	
☑	Roofs	
☑	Roof	IfcRoof
☑	Walls	
☑	Building	NL.IMBAG.Pand.0503...
☑	Building	NL.IMBAG.Pand.0503...
☑	Slabs	
☑	Roofs	
☑	Walls	
☑	Building	NL.IMBAG.Pand.0503...
☑	Building	NL.IMBAG.Pand.0503...
☑	Slabs	
☑	Roofs	
☑	Walls	
☑	Building	NL.IMBAG.Pand.0503...
☑	Building	NL.IMBAG.Pand.0503...
☑	Slabs	
☑	Roofs	
☑	Roof	IfcRoof
☑	Walls	
✓ ☑	Wall	IfcWall
☑	Wall	IfcWall

Name	Value	Unit
Location		
Building	NL.IMBAG.Pand.0503100000032914-0	
Top Elevation	3.99	m
Bottom Elevation	-46.784	m
Global Top Elevation	3.99	m
Global Bottom Elevation	-46.784	m
Geometry		
Has Own Geometry	Yes	
Children Have Geometry	No	
Global X	15.044	m
Global Y	345.777	m

Part III – Roadmap & community

3DGI

TUDelft 3Dgeoinfo

The next version of 3DBAG

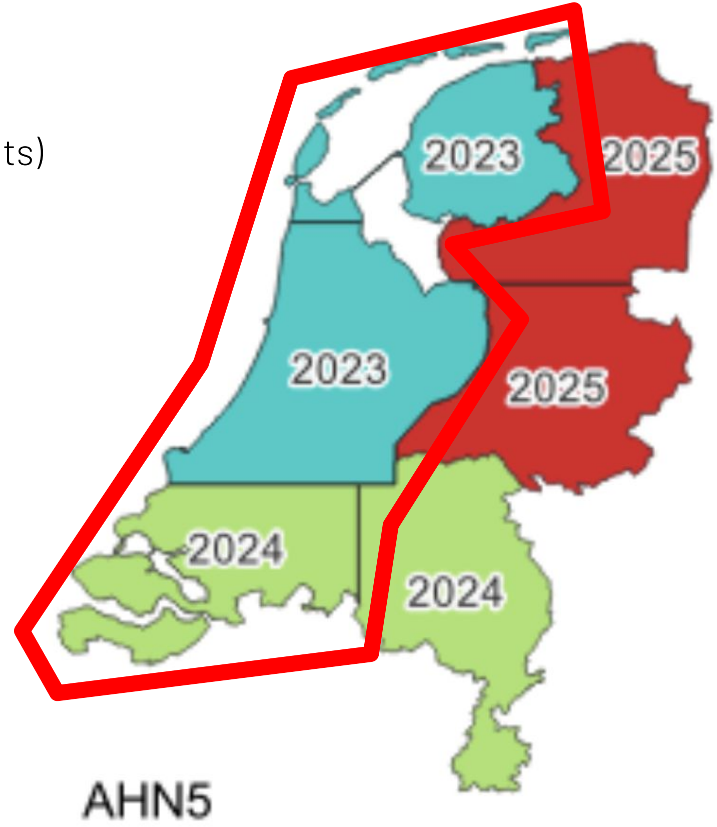
Next steps?

Currently working on a new 3DBAG version:

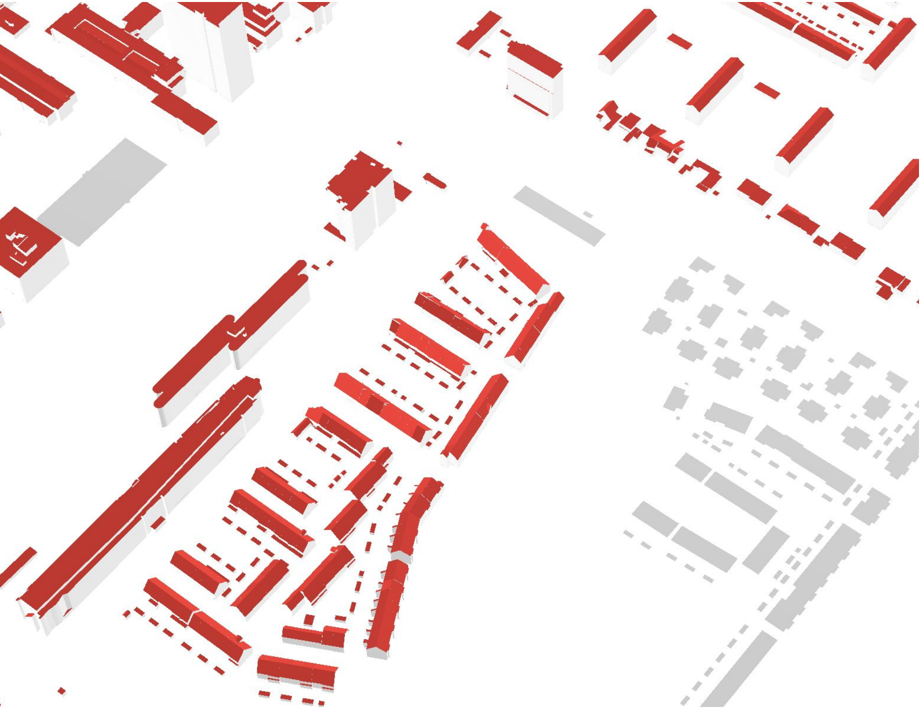
- using our improved software pipeline
    - outputs also object without 3D geometry (eg. due to no points)
    - Improved detection glass roofs
- AHN5 pointcloud Update
    - update buildings west NL

Versions after:

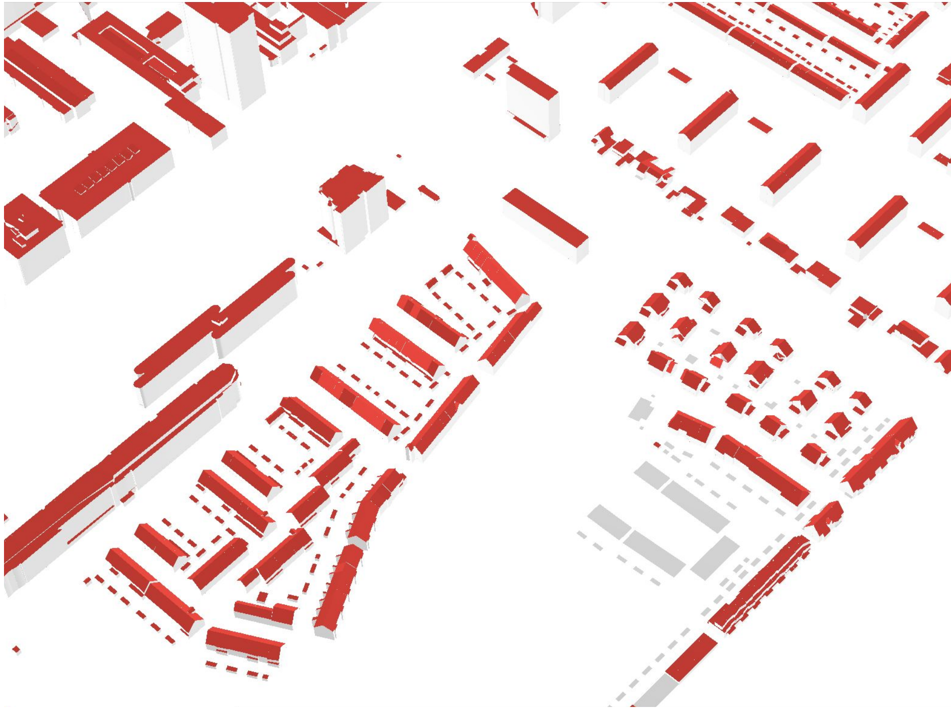
- Add IFC as tile format
- Update 3DBAG webviewer + 3D Tiles
- Improvements reconstruction algorithm
    - Improved plane detection
    - Better geometry regularisation
    - Improved dormer detection
- ...



AHN5 sneak preview

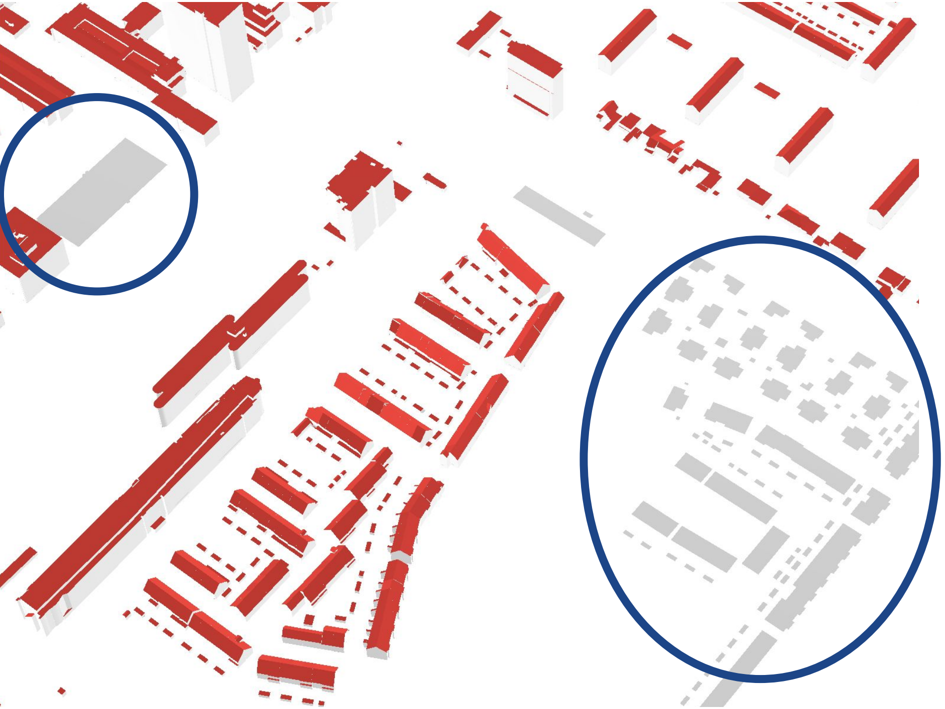


AHN4

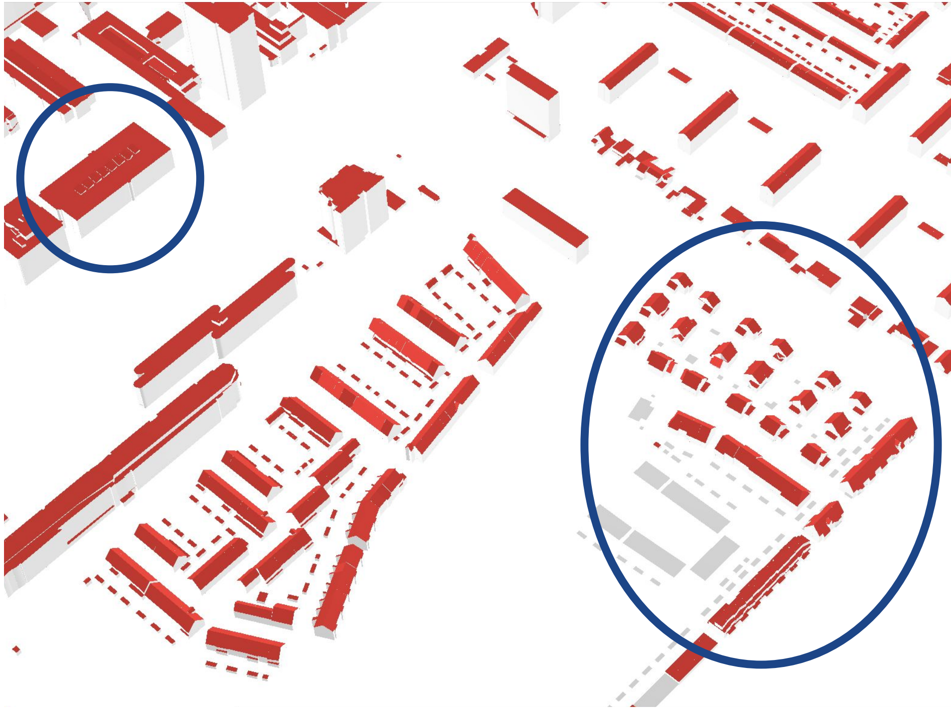


AHN5

AHN5 sneak preview

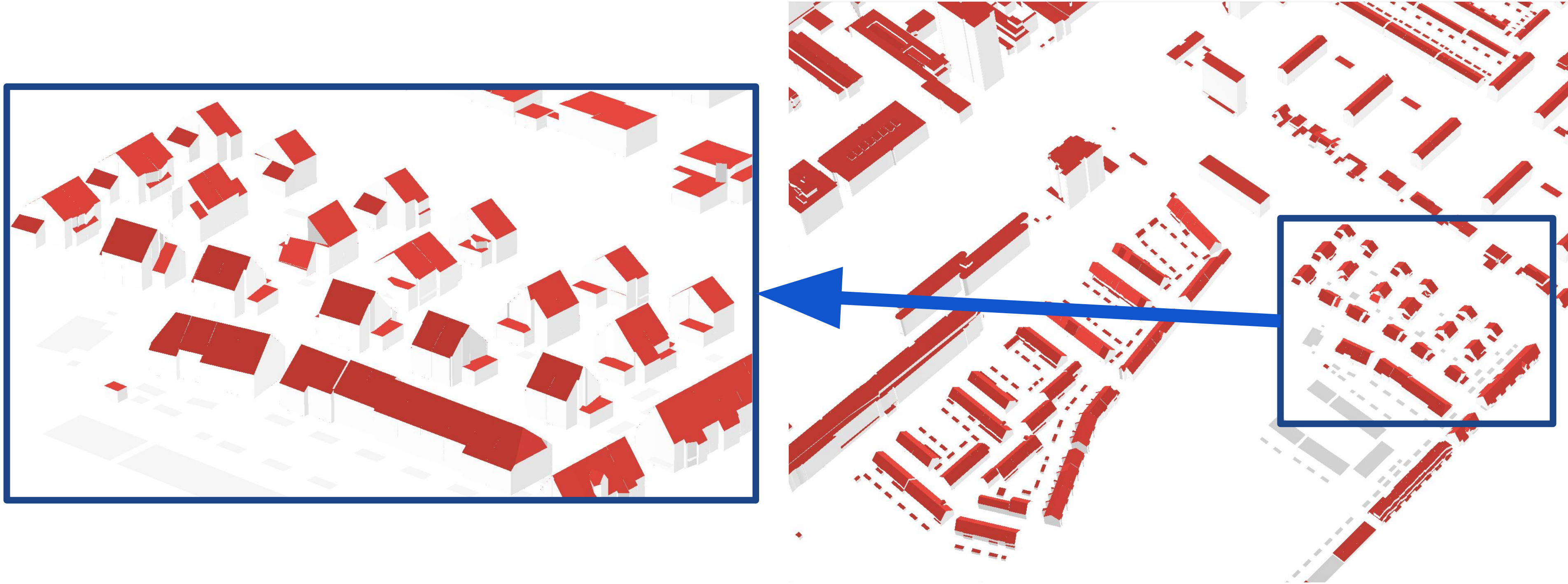


AHN4



AHN5

AHN5 sneak preview

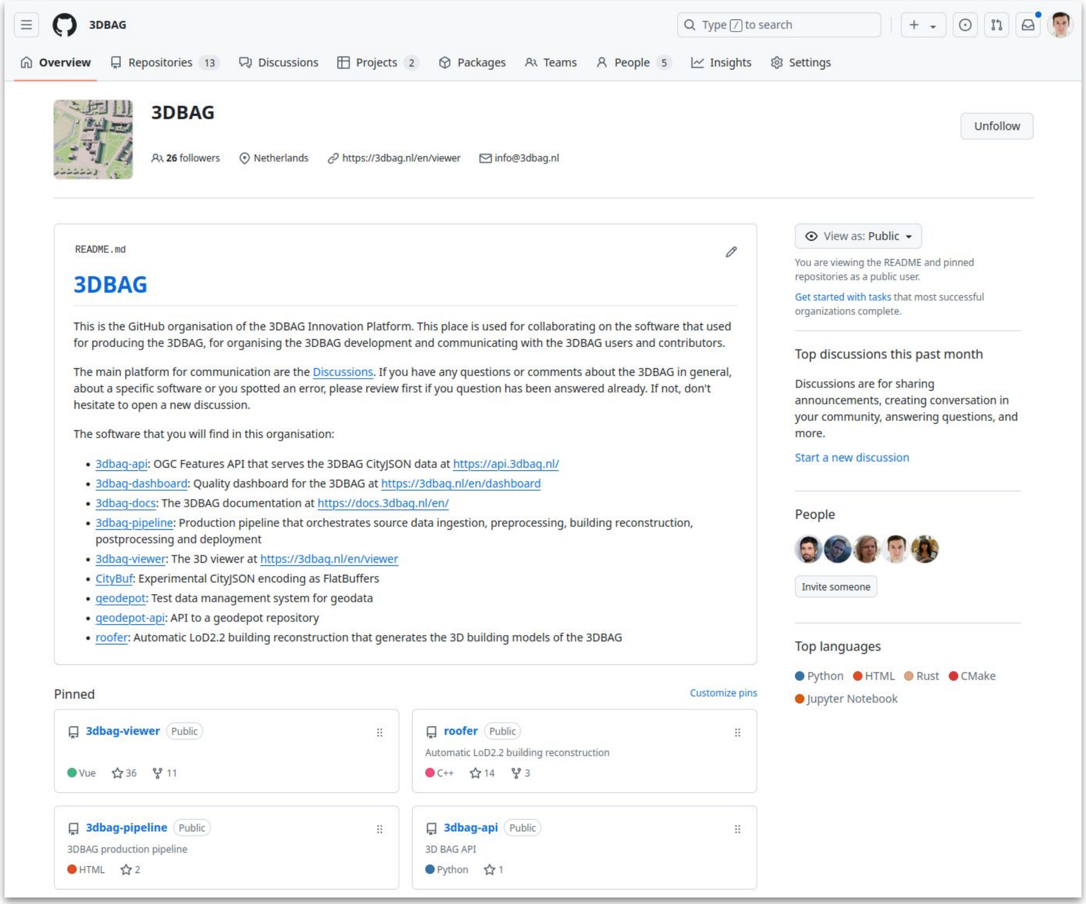


AHN5

3DBAG organisation and contributors

3DBAG on GitHub

https://github.com/3DBAG



3DBAG

26 followers · Netherlands · https://3dbag.nl/en/viewer · info@3dbag.nl

README.md

3DBAG

This is the GitHub organisation of the 3DBAG Innovation Platform. This place is used for collaborating on the software that used for producing the 3DBAG, for organising the 3DBAG development and communicating with the 3DBAG users and contributors.

The main platform for communication are the Discussions. If you have any questions or comments about the 3DBAG in general, about a specific software or you spotted an error, please review first if you question has been answered already. If not, don't hesitate to open a new discussion.

The software that you will find in this organisation:

- 3dbag-api: OGC Features API that serves the 3DBAG CityJSON data at https://api.3dbag.nl/
- 3dbag-dashboard: Quality dashboard for the 3DBAG at https://3dbag.nl/en/dashboard
- 3dbag-docs: The 3DBAG documentation at https://docs.3dbag.nl/en/
- 3dbag-pipeline: Production pipeline that orchestrates source data ingestion, preprocessing, building reconstruction, postprocessing and deployment
- 3dbag-viewer: The 3D viewer at https://3dbag.nl/en/viewer
- CityBuf: Experimental CityJSON encoding as FlatBuffers
- geodepot: Test data management system for geodata
- geodepot-api: API to a geodepot repository
- roofer: Automatic LoD2.2 building reconstruction that generates the 3D building models of the 3DBAG

You are viewing the README and pinned repositories as a public user.

Get started with tasks that most successful organizations complete.

Top discussions this past month

Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

Start a new discussion

People

Top languages

● Python ● HTML ● Rust ● CMake ● Jupyter Notebook

Pinned

Repository	Description	Language	Stars	Forks
3dbag-viewer (Public)		Vue	36	11
roofer (Public)	Automatic LoD2.2 building reconstruction	C++	14	3
3dbag-pipeline (Public)	3DBAG production pipeline	HTML	2	
3dbag-api (Public)	3D BAG API	Python	1	

We are ready for contributors

- all software Open Source
- permissive licenses (MIT, Apache-2.0, GPLv3)
- independent governance and coordination by Geonovum (3DBAG Innovation Platform)
- public communication (Zulip, GitHub)
- contributor guidelines (in progress)
- public roadmap (in progress)

Ways to contribute

- Use the 3DBAG data and report how it goes
- Use the software and report how it goes
- Improve the 3DBAG data documentation
- Improve the software user and developer documentation
- Fix an issue and open a PR
- Spread the word about the 3DBAG 😀

- We are looking for sponsors (contact us)

Ways to contribute

- Use the 3DBAG data and report how it goes
- Use the software and report how it goes
- Improve the 3DBAG data documentation
- Improve the software user and developer documentation
- Fix an issue and open a PR
- Spread the word about the 3DBAG 😀

- We are looking for sponsors (contact us)

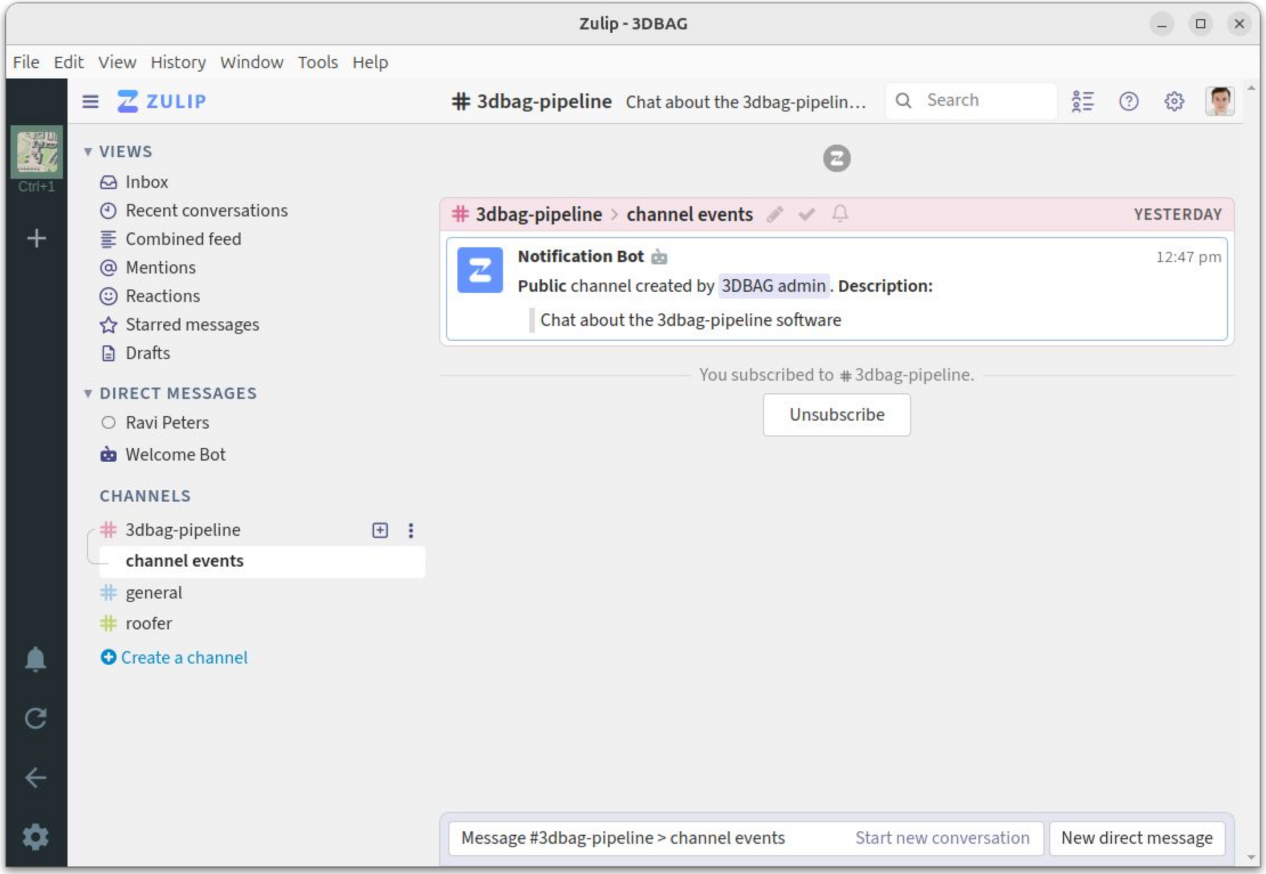
Guidelines per repository

Communication for developers

Newsletter

https://tinyurl.com/52dnazfs

Chat – https://3dbag.zulipchat.com



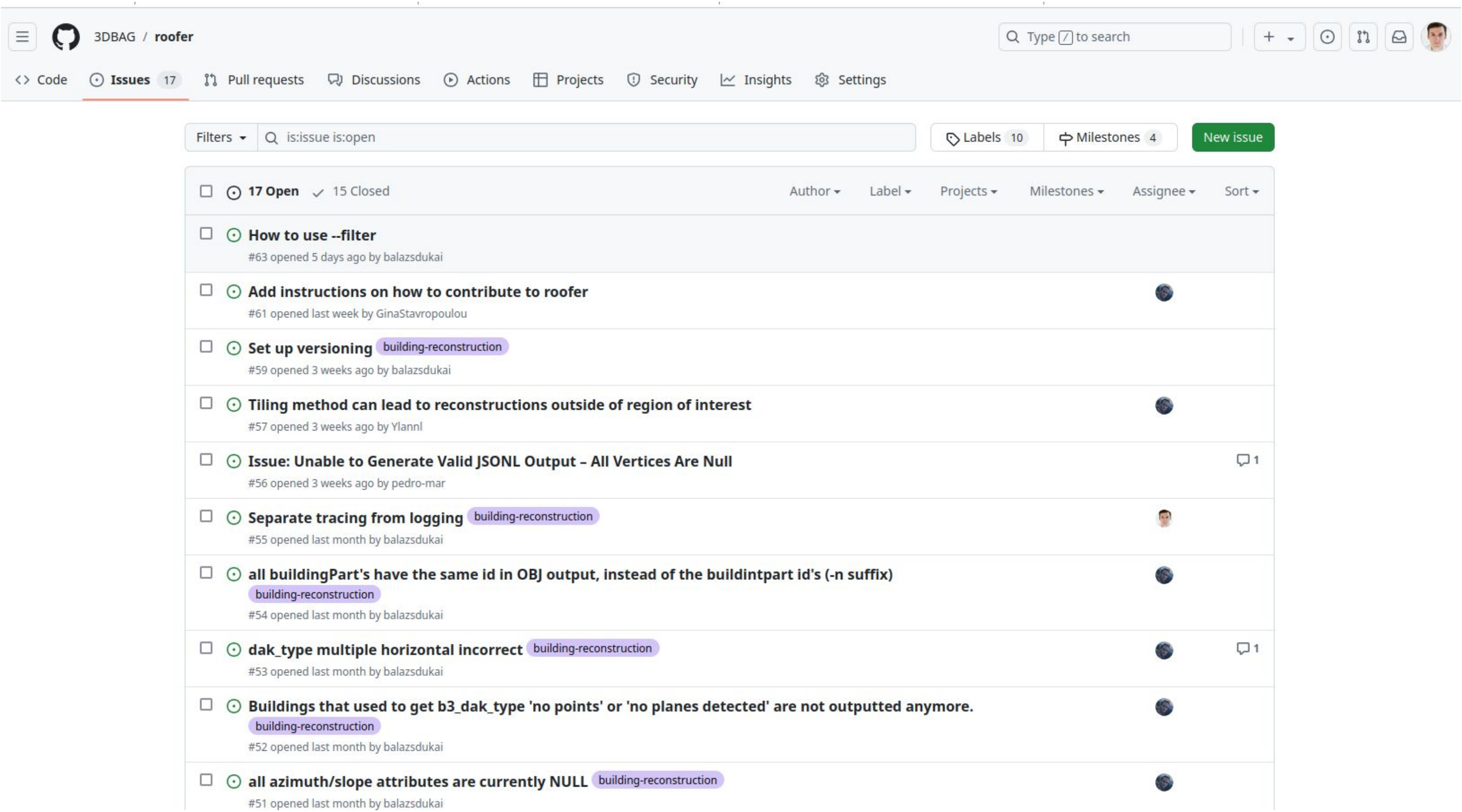
Notification Bot 12:47 pm

Public channel created by 3DBAG admin. **Description:**

> Chat about the 3dbag-pipeline software

You subscribed to #3dbag-pipeline.

Issues per repository



is:issue is:open

17 Open ✓ 15 Closed

□ How to use --filter
#63 opened 5 days ago by balazsdukai

□ Add instructions on how to contribute to roofer
#61 opened last week by GinaStavropoulou

□ Set up versioning building-reconstruction
#59 opened 3 weeks ago by balazsdukai

□ Tiling method can lead to reconstructions outside of region of interest
#57 opened 3 weeks ago by Ylannl

□ Issue: Unable to Generate Valid JSONL Output – All Vertices Are Null (1 comment)
#56 opened 3 weeks ago by pedro-mar

□ Separate tracing from logging building-reconstruction
#55 opened last month by balazsdukai

□ all buildingPart's have the same id in OBJ output, instead of the buildintpart id's (-n suffix) building-reconstruction
#54 opened last month by balazsdukai

□ dak_type multiple horizontal incorrect building-reconstruction (1 comment)
#53 opened last month by balazsdukai

□ Buildings that used to get b3_dak_type 'no points' or 'no planes detected' are not outputted anymore. building-reconstruction
#52 opened last month by balazsdukai

□ all azimuth/slope attributes are currently NULL building-reconstruction
#51 opened last month by balazsdukai

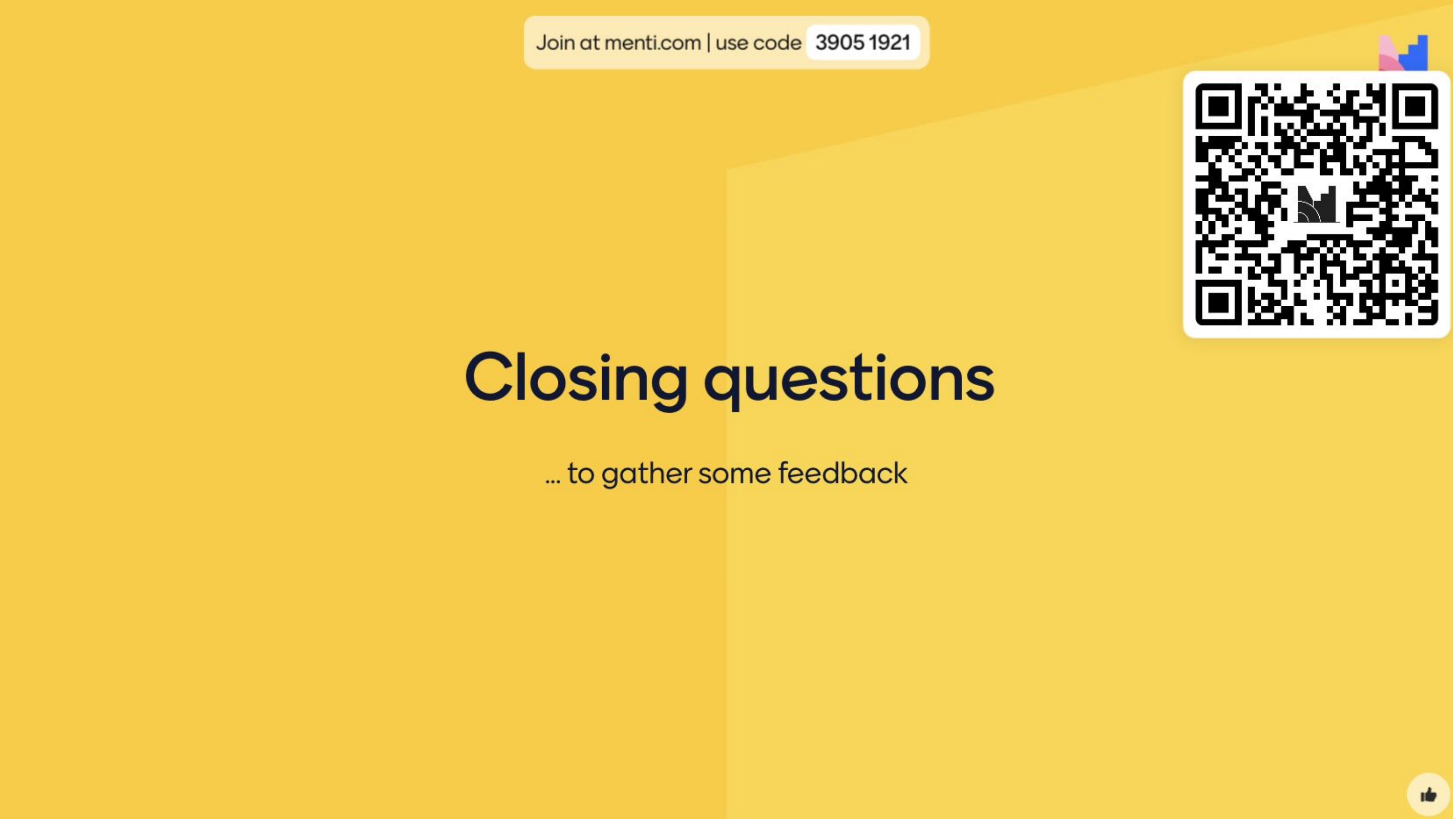
Thank you!

3DGI

TUDelft
3Dgeoinfo

Register for 3DBAG newsletter:

Time for some feedback...



Join at menti.com | use code 3905 1921

Closing questions

... to gather some feedback

QA session