Deliverable

D4.4
DUET visualization Components v1

Project Acronym:	DUET	
Project title:	Digital Urban European Twins	
Grant Agreement No.	870697	
Website:	www.digitalurbantwins.eu	
Version:	1.0	
Date:	03/06/2021	
Responsible Partner:	VCS	
Contributing Partners:	AIV	
Reviewers:	**Internal** Gert Vervaet (AIV), Philippe Michiels (IMEC), Hans Cornelissen (TNO) **External** Pieter Morlion Michiel Van Peteghem Andrew Stott	
Dissemination Level:	Public Demonstrator	X
	Confidential – only consortium members and European Commission	

This project has received financial support from the European Union's Horizon 2020 Programme under grant agreement no. 870697

Revision History

Revision	Date	Author	Organization	Description
0.1	24.03.2021	Thomas Adolphi	VCS	Initial structure
0.2	17.05.2021	Thomas Adolphi	VCS	Content
0.3	18.05.2021	Thomas Adolphi	VCS	Added VCS tools
0.4	19.05.2021	Thomas Adolphi	VCS	1st full Draft version
0.5	21.05.2021	Thomas Adolphi	VCS	Added intro, summary, conclusion, relation to epics
0.6	24.05.2021	Philiipe Michels	IMEC	Review
0.7	25.05.2021	Andrew Stott		Review
0.8	26.05.2021	Pieter Morlion	MORE LION	Review
0.9	31.05.2021	Thomas Adolphi	VCS	Response to reviewer comments and added content
1.0	4.06.2021	Thomas Adolphi, Jurgen Silence, Lieven Raes	VCS, AIV	Final version

Table of Contents

List of figures

1 Executive Summary

This document gives an overview about the implemented and integrated visualizations and tools for future versions of DUET visualizations.
The described functions, tools and visualizations of this document are integrated here [https://duet.virtualcitymap.de/closedBetaDev].

This document shows the current developments for the DUET solution as well as provided functionalities from VCS for VC MAP users. The visualizations described in the beginning of this document starting from chapter 1 to chapter 6 including chapter 15 are made and implemented for DUET explicitly. The tools and resulting visualizations described in chapter 7 to chapter 14 are made by VCS for all customers of VCS and are generic.

This document describes the components developed for analysis and visualization in the current DUET visualization client. Visualization components for use in reports will be described in D4.5 and for use in dashboards and other UI components will be part of D4.6. Therefore other visualizations beyond 3D will be described in those documents. Of course further improvements to a more generic way of styling and visualization of attributes, properties and features will be done in upcoming versions of DUETs visualization component.

A detailed description - as a kind of how-to / user manual - of the developed / existing tools is done in D5., which is part of the Closed Beta release.

2 Introduction

This Demonstrator report describes the developed and integrated tools of the Closed Beta demonstrator.

Chapters 1 to 3 and chapter 15 relate **to the DUET epics G1 - G4, G6 - G8 and P1**.
These epics mainly deal with simulation scenarios, adding data, inspecting results of simulations, and visualizing them. Even if these visualizations are not part of this document here, the development was done and is part of the closed beta release and is available in the current version of the DUET Digital Twin.

Chapters 4 to 6 **are related to the DUET epic P4.**
Each dataset added to the view from the DUET catalogue can be styled accordingly by its inherited attributes. The described styling options are not developed into full detail and are, for a significant part, the different kinds of geometries and styling cases; nevertheless, they give a good indication of how a user can style the data independently.

Chapter 7 is mainly **related to DUET epics P6 and P9**.
By using the VC Planner tool, users have the possibility to create Planning scenarios as an own future state of the region or city. This tool is provided to DUET from VCS with its own components and databases as a separate extension since it needs persistent storage of models, plannings, an own database, and a back-end server. However, users can now import from several CAD formats into the view and thus create future states of the city.

Chapters 8 to 14 **are not explicitly related to specific DUET epics**, which are very helpful for analysis, inspection and visualizations of cities.

3 Data integration and visualization via Duet Data catalogue

The Duet data catalogue is one of the major components for getting access and finding data available for all regions. The major requirements are:

1. simple presentation of available data
2. simple selection methods
3. simple overview of available date
4. simple search by “value”

As figure 1 shows, the duet catalogue representation is divided into three main parts and based on the mocks presented in D4.1. The parts are as follows:

1. upper left corner - a simple overview map to be used for displaying the extent of your data on the earth surface
2. left panel - providing a simple search/filter function as well as an overview about the dataset types for the current regions and all available datasets in the catalogue
3. main panel - providing a list of data sets per type and some descriptive elements and functions

Main dataset types, sorted into tabs in the catalogue view are:

1. Services - showing all kind of services supported by the visualization component, such as WMS, WMTS and TMS
2. Terrain data - showing Cesium terrain data available for the respective region
3. 3D data - showing available 3d datasets, such as Cesium Buildings (3D Tiles), and Cesium Pointclouds
4. GeoJSON - showing available vector data in GeoJSON format
5. Cases - showing the available cases provided via Duet Cases-API
6. Dynamic types:
    a. *Traffic results* - available and shown once a traffic simulation result is available
    b. *Noise results* - available and shown once a noise simulation result is available
    c. *Air results* - available and shown once an air simulation result is available

CityFlows results - available and shown once a CityFlows simulation result is available.

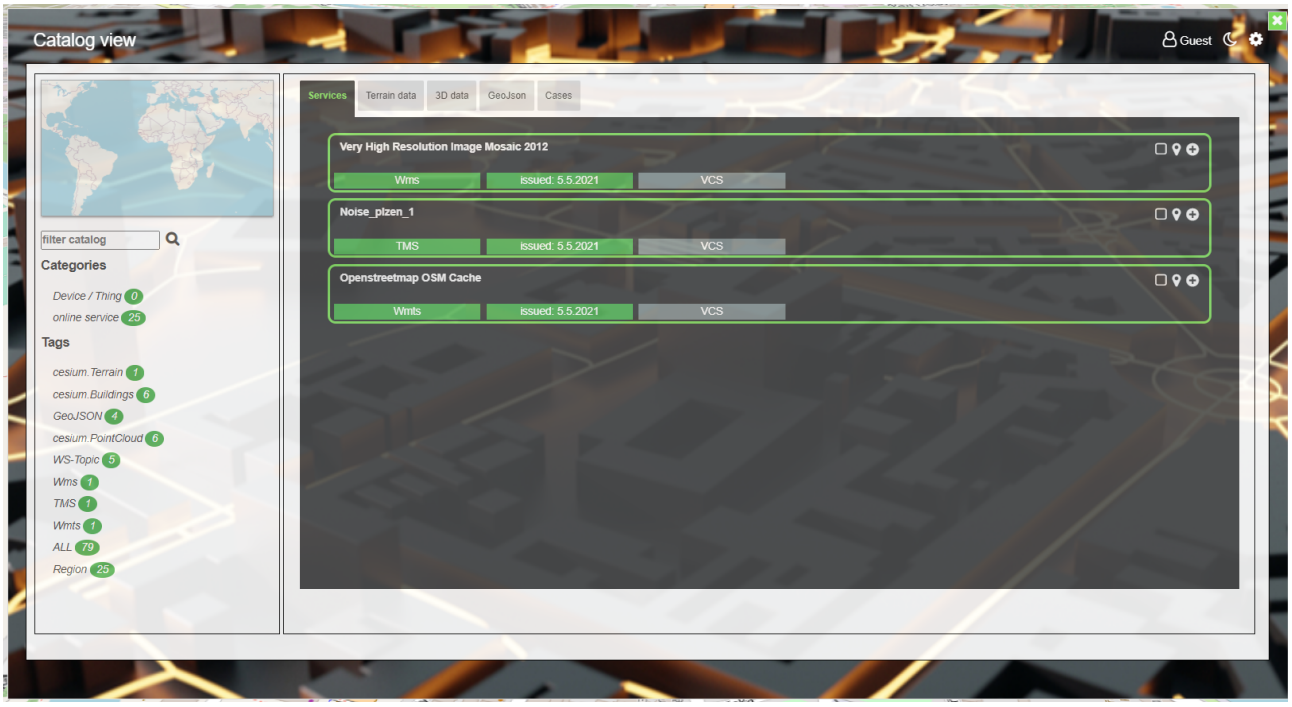


Figure 1: Duet data catalogue representation inside visualization component

As described above and shown in figure 2, the upper left part of the catalogue view is a small map view based on OpenStreetMap that shows the provided extent of datasets. This is useful if the descriptive part of a dataset does not indicate the spatial extent or location of the dataset. When the user clicks on the small rectangle per dataset, the extent of the data will be shown in the map overview.

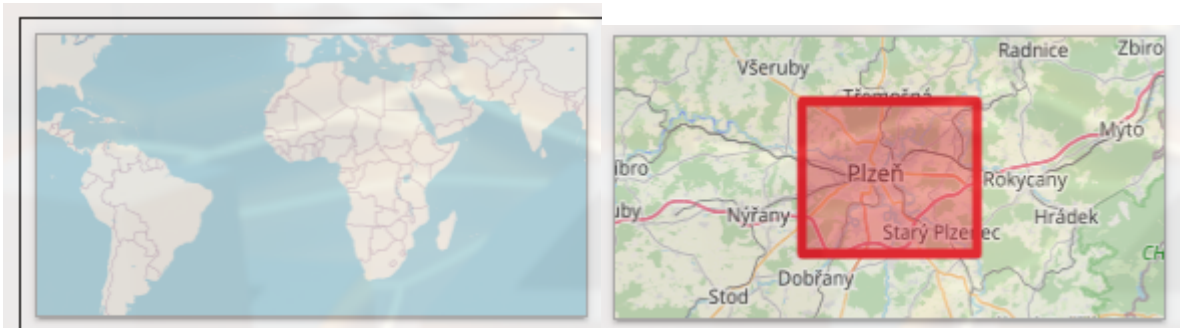


Figure 2: Duet data catalogue map overview (left = initial | right = showing the extent of a dataset in Plzen region)

The rest of the left-hand side panel gives the user an overview of the dataset types in conjunction with some selection methods.
The first option is to use the filter catalogue field to search and filter the available datasets according to a phrase in the name or type, or description of the dataset. Thus typing in **"*cesium*"** will filter the whole catalogue and provide only datasets containing that phrase.

The second option is to use the tags to overview the types for setting a filter to the catalogue and showing only datasets owning that tag/type. The tag is simply the type of the dataset.
Thus clicking on **"WMS"** will only list the WMS datasets inside the service tab.

Figure 3: Duet data catalogue overview panel showing the available datasets for the current region

As figure 4 & figure 5 show, the entire content of the DUET catalogue is presented as a scrollable list of entries. Each entry in the list of datasets owns some descriptive and some functional items. The descriptive items are listed on the left-hand side of each entry. Those are:

1. the title of the dataset (top)
2. the type of the dataset (first left bottom element)
3. issue date (second bottom element)

Figure 4: Duet data catalogue main panel showing the available datasets for the current selection such as type & region

1. change date (if available - third bottom element)
2. data provider (third / fourth bottom element)

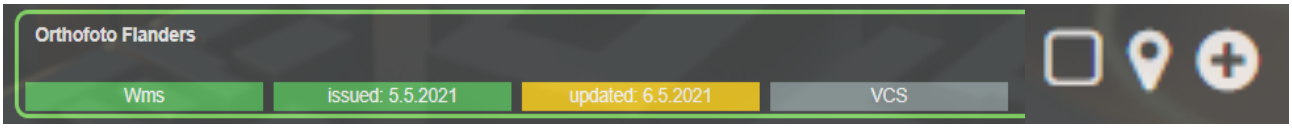


Figure 5: Duet data catalogue entry

On the right-hand side of each data catalogue entry the functional items are organized. Functions are:

1. Show the extent of the dataset - square element. Clicking on that item takes the extent (BBOX) of the dataset and shows it inside the overview map; see also figure 2.
2. Jump to position in map - place marker. Clicking on that item takes the BBOX of that catalogue entry element and calculates a centre position, and jumps to that position in the main view.
3. Add dataset - plus sign. Clicking on that item fetches the relevant fields of that catalogue entry and adds the dataset to the list of map datasets. The id of that catalogue entry will be stored internally for later use in a case section, see chapter Create a new a case and save it via cases-api & 3.2 Extend and save a case via cases-api

4 Case support

Cases can be described as entries into the system, with a view specific for a certain analysis, discussion, policy action/decision. Depending on the selected case, datasets are selected, settings will be filled in, relevant models will be presented, and dashboards will be created in the platform component of the DUET portal. Cases are registered via cases-api and provided to users via the DUET platform or the visualisation client directly. Each case holds a lot of information as intended use, expected outcome and used datasets; thus, each case can be loaded and used inside the DUET environment.
As for the DUET portal, we defined 3 case variants:

- **Prepared cases** where the initial settings are predefined/preset to get an optimal visualisation of a specific case and its accompanying analysis as a starting point. This category includes the showcases, which are exceptional cases fully worked out, being key examples for similar (future) studies.

- **Semi generic cases** offer more exploration freedom to the visitor/user. For these pilot-related and epic-based (see Deliverable 2.3) cases, the initial settings and selected datasets can be easily adapted according to the visitor/user's personal interest or angle of approach of the visitor/user.

- **Generic cases** are defined by the user himself/herself. It is the user who decides which of the available data layers, models or dashboard will be utilised. Supported by the information available on the password-protected pages (the user needs to apply for a password first), datasets, models and dashboards can be freely used and additional datasets can be **uploaded** to create an own - tailor-made - DUET with access to all available datasets and tools.

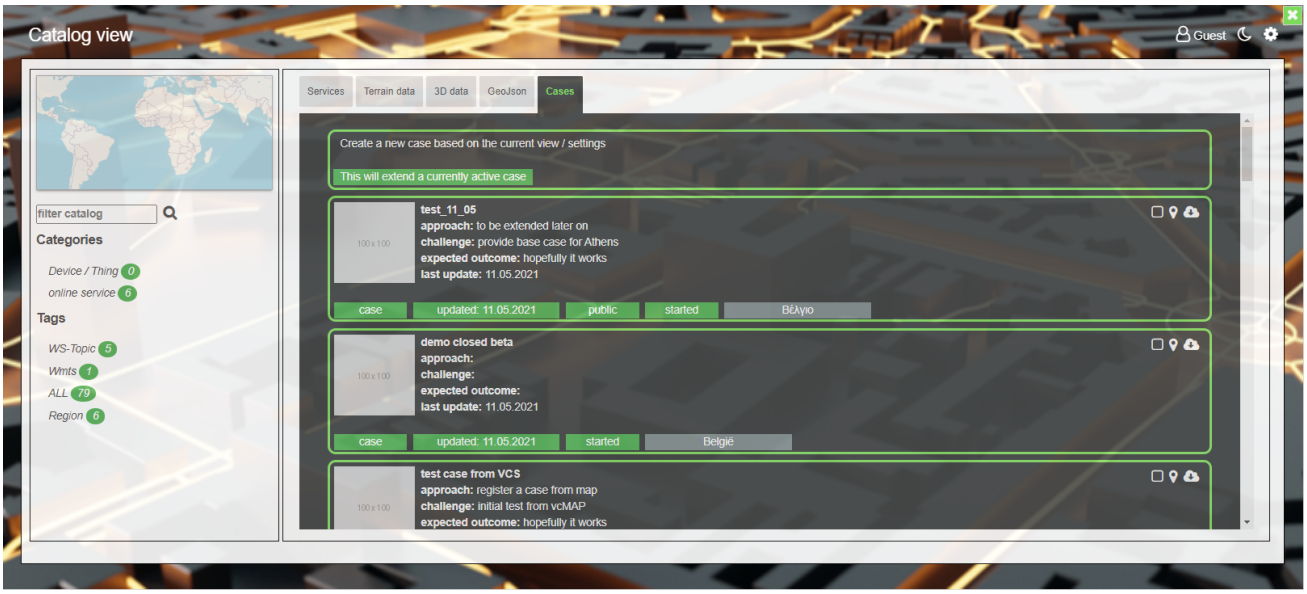


Figure 6: Duet data catalogue view - cases tab

Cases are presented to the user inside the visualization component in the catalogue view's cases tab. Here all listed cases of the cases-api will be shown to the user. Inside the tab, several functions are provided to the user.

1. overview about existing cases in DUET environment, see 3.1 Load a case
2. creation of a new case, see 3.2 Create a new case and save it via cases-api
3. update/extension of an existing case, see 3.3 Extend and save a case via cases-api

As figure 6 & figure 7 show, the whole content of the DUET cases catalogue is presented to the user as a scrollable list of entries. The list is sorted by issue/update date and uses the sortBy and sortOrder functionality of the DUET cases-api. Each entry in the list of cases owns some descriptive and some functional items. The descriptive items are listed on the left-hand side of each entry.

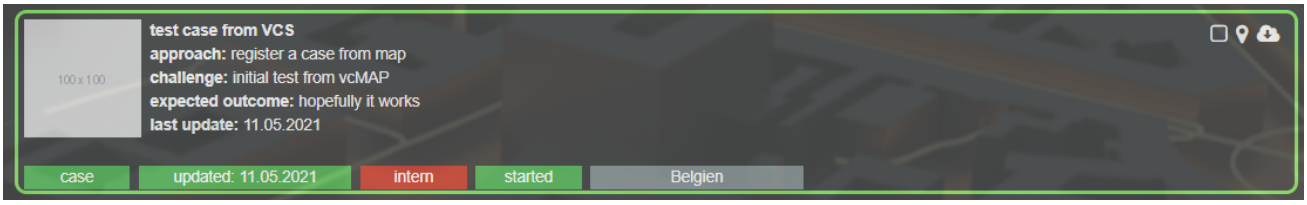


Figure 7: Duet data catalogue view - cases tab - case entry

The descriptive elements are:

1. a thumbnail of the case, if available. if not, a placeholder will be shown, followed by:
2. title of the case
3. approach description of the case
4. challenge description of the case
5. expected outcome description of the case
6. last update date of that case

Below the main descriptive elements of each case, some easy to fetch items are listed. Presented to the user as W3.css tag in a W3.css grid. Thus the most relevant information, except for the descriptive elements such as title etc., are highlighted. The highlighted elements are:

1. case
2. update date
3. target audience - depending on audience the tag is green = public, red = intern, yellow = all others,
4. status of the case - depending on status, the tag is green = started, red = stopped, yellow = all others
5. region or city where the case is located

In the upper right corner, the functional elements of each case entry are located. Here, similar to a catalogue entry, the functional items are:

1. Show the extent of the case- square element. Clicking on that item takes the extent (BBOX) of the case and shows it inside the overview map, see figure 2.
2. Jump to a position in map - place marker. Clicking on that item takes the BBOX of that case and calculates a centre position and jumps to that position in the main view.
3. load case - download sign. Clicking on that item fetches a case and adds the datasets associated with that case to the list of map datasets (see chapter 3.1 Load a case.

4.1 Load a case

Loading a case inside the visualization component works on clicking on the download case sign. By doing so, the whole map configuration will be loaded and the associated data sources will be added to the legend (content) tab of the visualization component. In addition to that, user-defined styles will be fetched from the case definition and will be added to the map as well, to be used in the style section of each layer (see Styling non-time dependent data based on individual attributes & Styling time-dependent data based on individual attributes).

4.2 Create a new case and save it via cases-api

At the top of the cases tab, the uppermost list element (see figure 8) provides users with the option to save a new case or update an existing case. A new dialogue opens by simply clicking on the list element, providing some UI elements relevant for creating or updating a case.

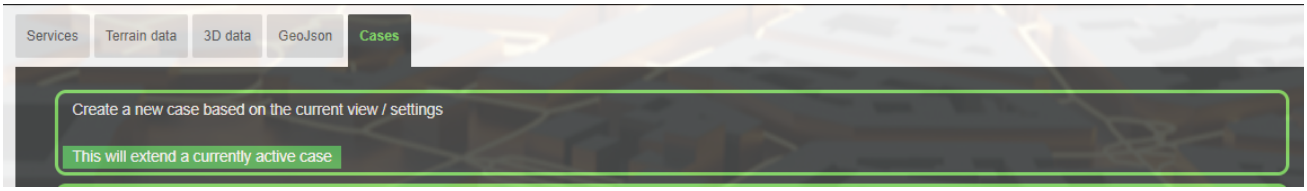


Figure 8: Duet data catalogue view - cases tab - create or update a case

As figure 9 shows, the “save case” dialogue asks for some user input by text fields. The main fields of user required inputs are the same fields as used for describing a case inside the duet case list view, except for the tags fields mainly used in the Duet platform [https://staging.citytwin.eu/] word cloud, filter function for cases, etc. By clicking on the “Save case” Button, some other required fields will be generated by the underlying API. The country, city, position will be created via geocoding using the Nominatim API [3] and fetching the relevant

parameters via posting the BBOX of the current view extent. Thus it is guaranteed that each case has an extent and a location. Afterwards, all relevant parameters will be taken and pushed via HTTP protocol POST to the DUET CASES API.

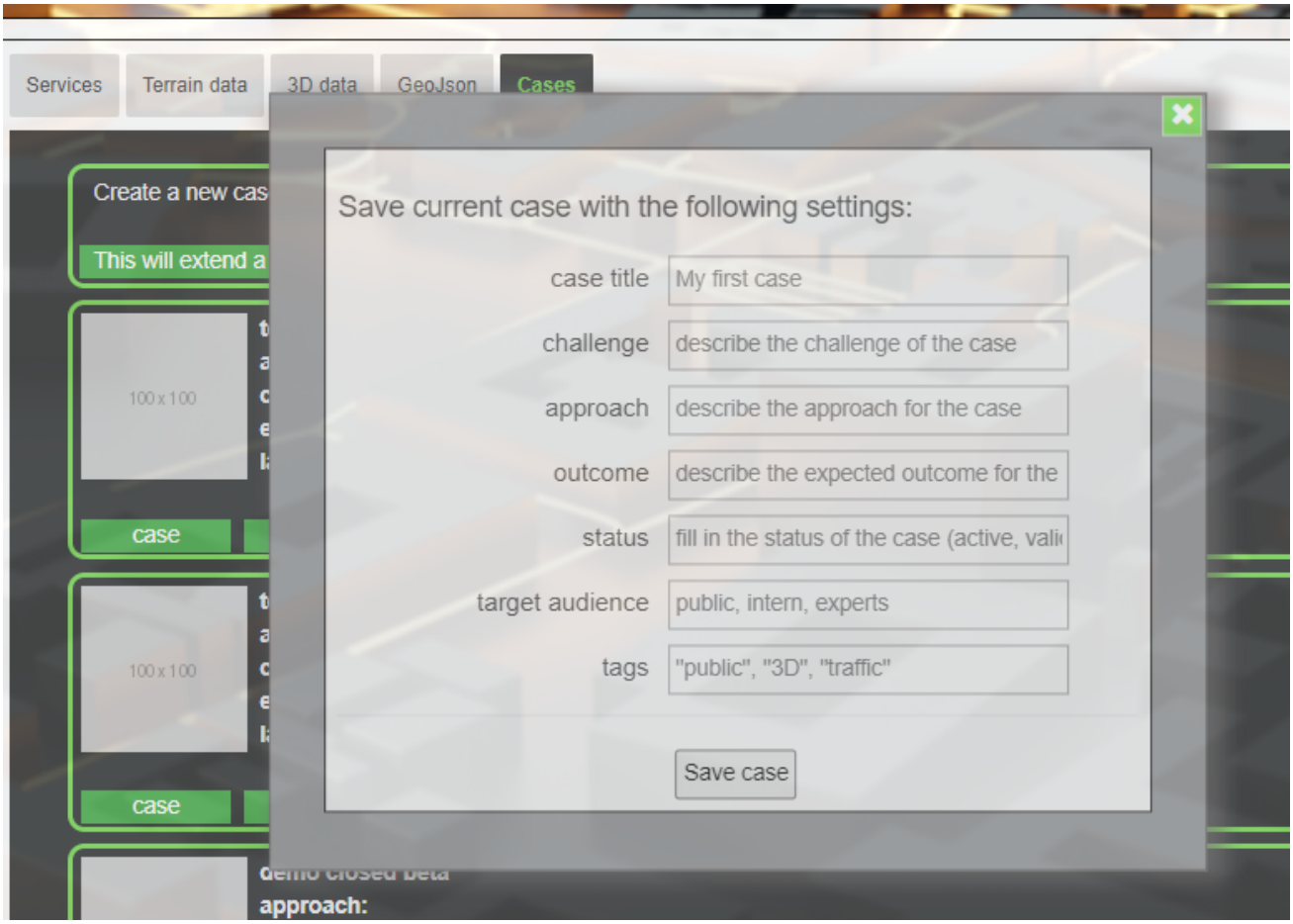


Figure 9: Duet data catalogue view - cases tab - create a new case dialog

The user will be informed about success or failure via notifications. See Notifications.

4.3 Extend and save a case via cases-api

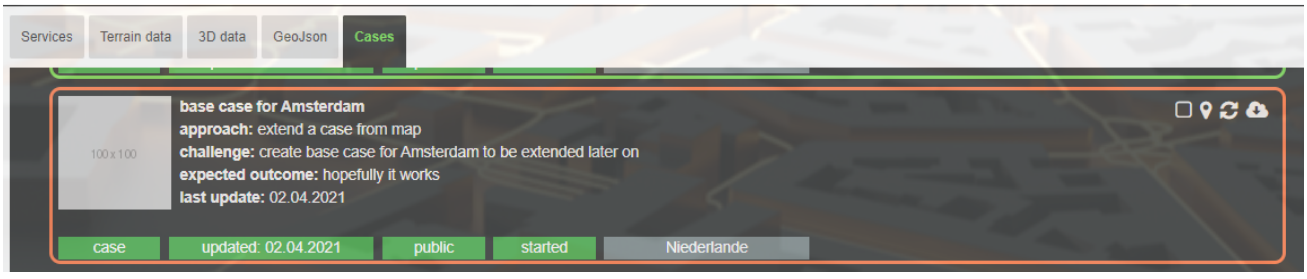


Figure 10: Duet data catalogue view - cases tab - loaded case entry

In addition to figure 7, where a standard case entry is shown, figure 10 shows an already loaded case - highlighted by a “coral” coloured frame. A new symbol is added to the functional section, showing a reload / update sign. The user can now see which case is loaded and is able to extend that case by adding new styles or datasets to that case. The first can be created by adding a new style as described in Styling non-time

dependent data based on individual attributes or Styling time-dependent data based on individual attributes. The latter by choosing dataset entries from the list of datasets and adding it to the visualization component.

Suppose a user clicks on that symbol or the main entry point for saving a case displayed in figure 8. In that case, the “Save Case” dialogue opens (see figure 11), showing now instead of default value-filled fields - the current loaded case information. The user can now keep the fields or change them and can then decide whether to save that case as a new case or simply update the existing case. By doing the latter, the existing case will be overwritten with the settings and properties of the new one.

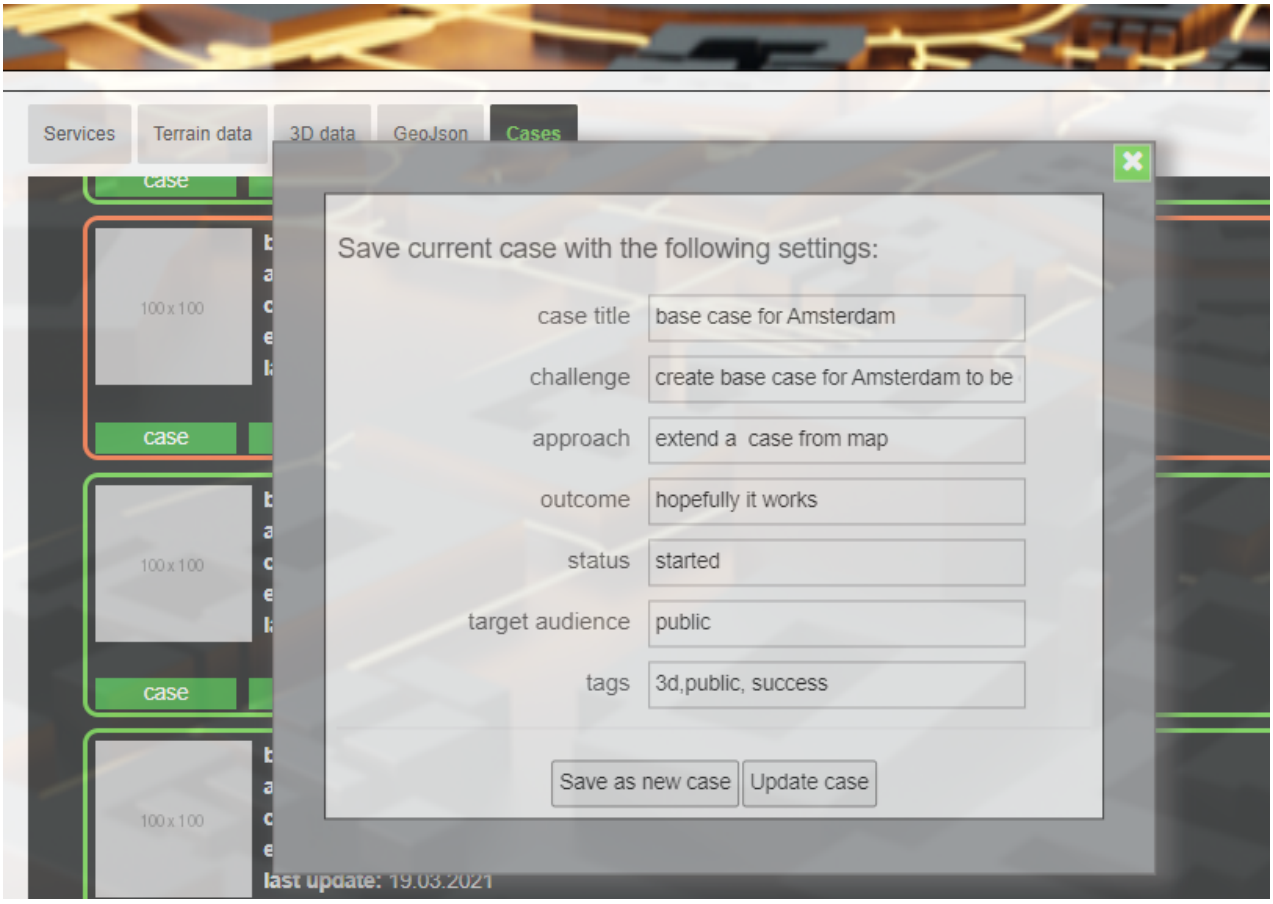


Figure 11: Duet data catalogue view - cases tab - update case dialog

5 Styling non-time dependent data based on individual attributes

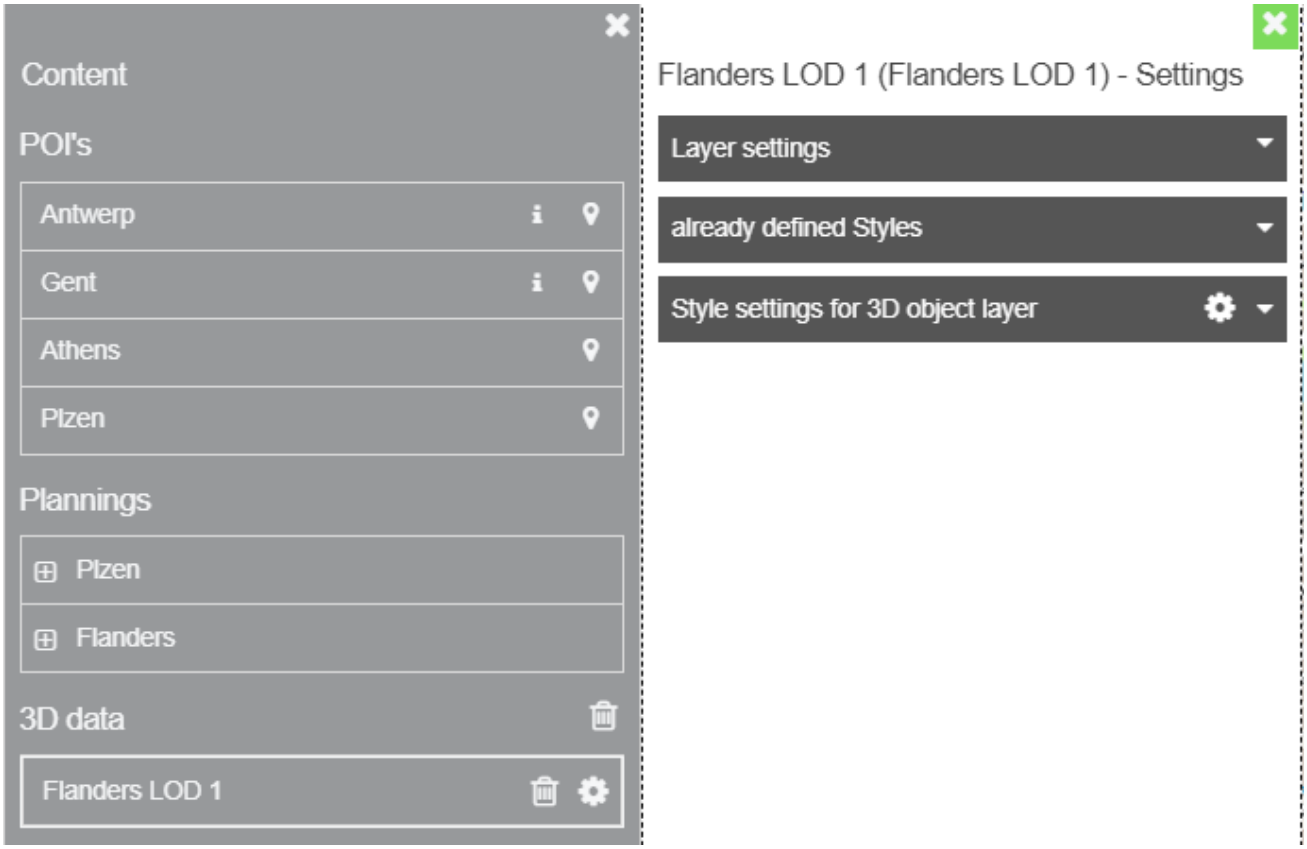


Figure 12: Added 3D Data to visualization component (left), layer settings panel (right)

After successfully adding a dataset to the visualization component, the dataset will be listed in the content panel on the left-hand side of the view, as presented in D4.1.
If a user wants to visualize the data, a simple click on the legend entry will enable that dataset and visualize it. A click on the trash icon will remove that dataset from view and legend.
A click onto the cog sign (settings) will open a new panel for layer and style settings. Here, a user can manually change the dataset title (in **Layer settings**) to be displayed with that in the legend.

In "**already defined Styles**", the user will find a list of styles that can be applied to the dataset. The list of available Styles is already filtered against attributes/properties of the dataset in comparison with the properties used in the style. Only if at least one property matches, the style will be available to be chosen by the user. The style definition is based on Cesium 3D-Tiles Styling language (https://github.com/CesiumGS/3d-tiles/tree/master/specification/Styling).
Some styles are already known to the system and defined by VCS. The following 2 examples show how to define a style for all kinds of features (2D / 3D), holding the attributes used in the style definition.
Thus a chosen style from the menu "**already defined Styles**" must be applied to the dataset by clicking on the ***"Apply Style"*** button.

A simple style can be defined as follows:

```
    {
      "name": "Intensity",
      "title": "CityFlows intensity",
      "declarativeStyle": {
        "show": "Boolean(${attributes.olcs_extrudedHeight}) === true",
        "color": {
          "conditions": [
            [
              "true",
              "color('grey',.9)"
            ]
          ]
        }
      }
    }
```

This example style checks if an attribute "***olcs_extrudedHeight***" exists and shows at first all features containing that attribute. Afterwards, a colour condition will be applied to it, setting the colour of each element - holding the attribute "***olcs_extrudedHeight***" - to grey with an opacity value of 0.9.

A bit more complex style can be defined as follows:

```
    {
      "name": "traffic3D",
      "declarativeStyle": {
        "defines": {
           "FLOW":       "Number(${attributes.flow})       >       0       ?
Number(${attributes.flow})    :    (Number(${attributes.traffic})    >    0    ?
Number(${attributes.traffic}) : 1.0)",
           "CAPACITY":     "Number(${attributes.capacity})     >     0     ?
Number(${attributes.capacity}) : 1.0"
        },
        "show": "true",
        "color": {
          "conditions": [
            [
              "(${FLOW}-${CAPACITY}) >= 100",
              "color('red', 0.5)"
            ],
            [
              "(${FLOW}-${CAPACITY}) > 1",
              "color('yellow', 0.5)"
            ],
            [
              "true",
              "color('green', 0.5)"
            ]
          ]
        }
      }
    }
```

Here the style at first defines some general attribute names ("***FLOW***" & "***CAPACITY***") based on the existence of either the attribute ***flow/traffic*** or ***capacity.*** Afterwards, 3 color conditions will be applied to the features.

1. If ***flow - capacity*** is greater than 100, a red colour with 0.5 opacity will be applied
2. if ***flow - capacity*** is greater than 1 and smaller than 100 a yellow colour with 0.5 opacity will be applied
3. in all other cases, a green colour with 0.5 opacity will be applied

Of course, a user can define his style by opening the menu ***"Style settings for XXX layer"***. The XXX means that this option is available for Vector, 3D Objects and time-dependent datasets.

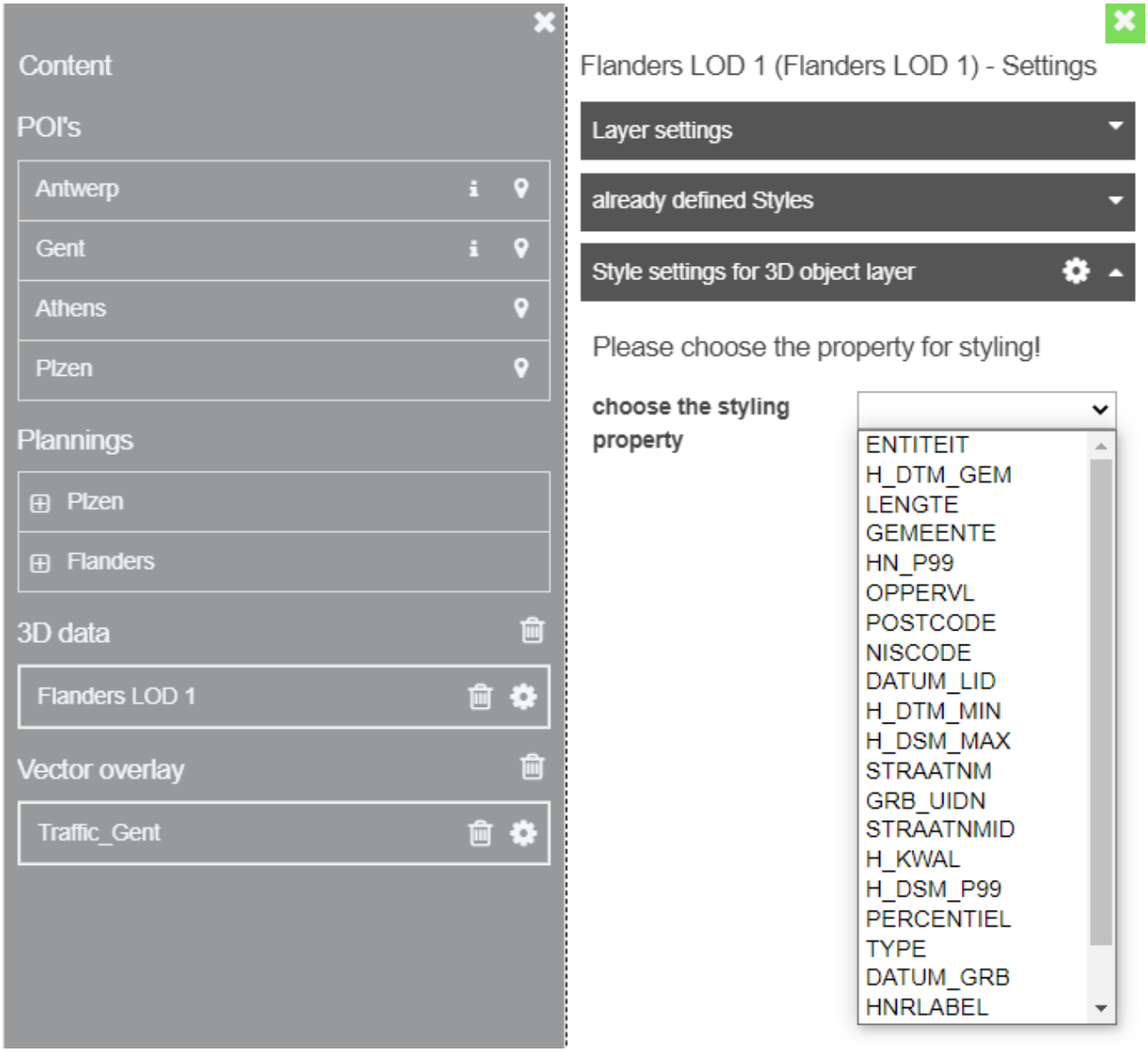


Figure 13: Added 3D Data to visualization component (left), style settings menu with available attributes (right)

As figure 13 shows, the user can choose from the list of available attributes of the features for styling the features. Typical attributes for styling are height attributes or any other number attributes.
NOTE: the styling functionality only works for Line / Polygon / Building features with NUMBER like attributes. A style using string values will not work!

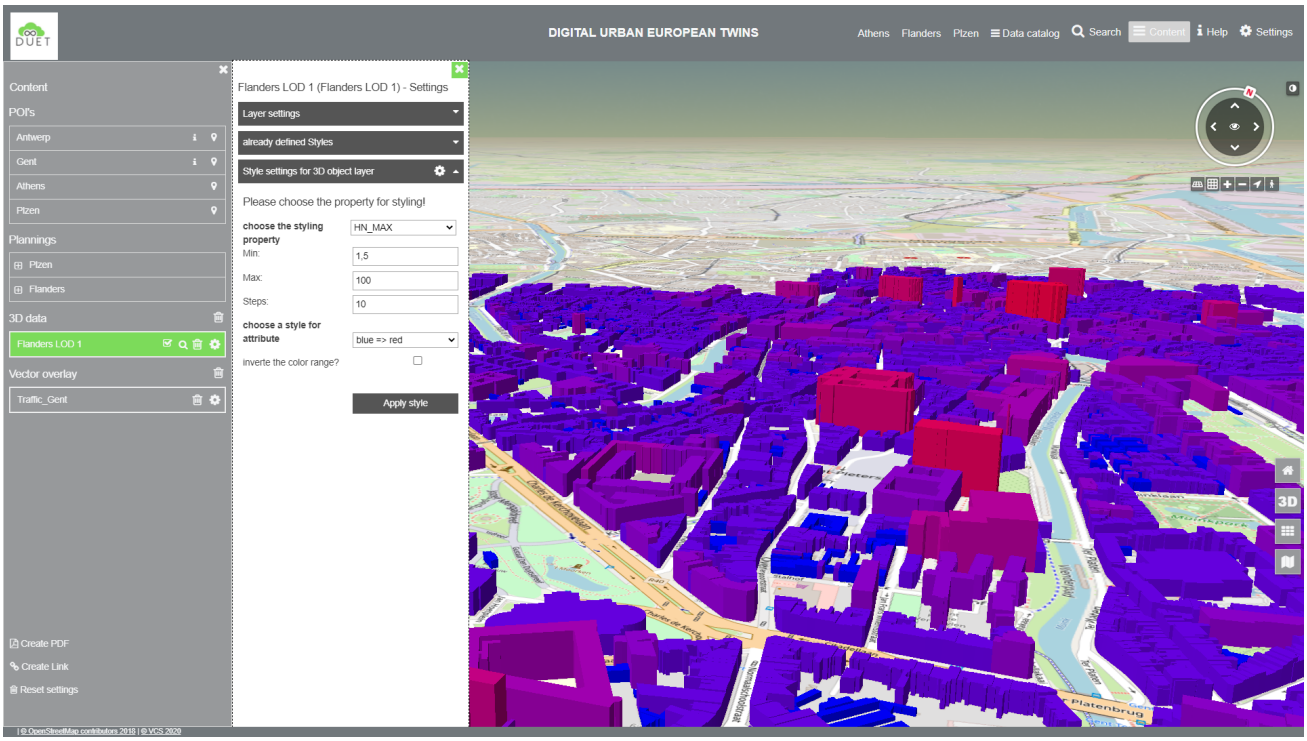


Figure 14: Applied user style to added 3D Data

Figure 14 shows a user-defined style for attribute ***"HN_MAX"*** of the dataset ***"Flanders LOD 1"***. It applies colours from blue to red for buildings having a height in the range of [1.5; 100] m in 10 colour steps. Thus, the highest buildings will be visualized in red and the lowest buildings in blue.

The user-defined style will be integrated into the system and will be available for other datasets using the same attribute as defined in the style.

6 Styling time dependent data based on individual attributes

Styling features individually by time is generally relying on the same styling method as without time values. However, some differences are worth pointing out. At first, the API scans all features for array values, as shown below. Thus a time-dependent dataset should at least hold the information about simulation time interval as well as some attributes represented in an array of values, such as:

```
"result": [
        {
           "id": 125,
           "flow": [1256, 1500],
           "cost": [0.08, 0.05]
        },
       {.... }
     ]
```

If such attributes are found, the component will ask the user if time-dependent features should be styled. If so, he has to choose the time array attribute first and afterwards, he can choose the attribute property for colour and stroke style. The styling in general works as described in Styling non-time dependent data based on individual attributes.

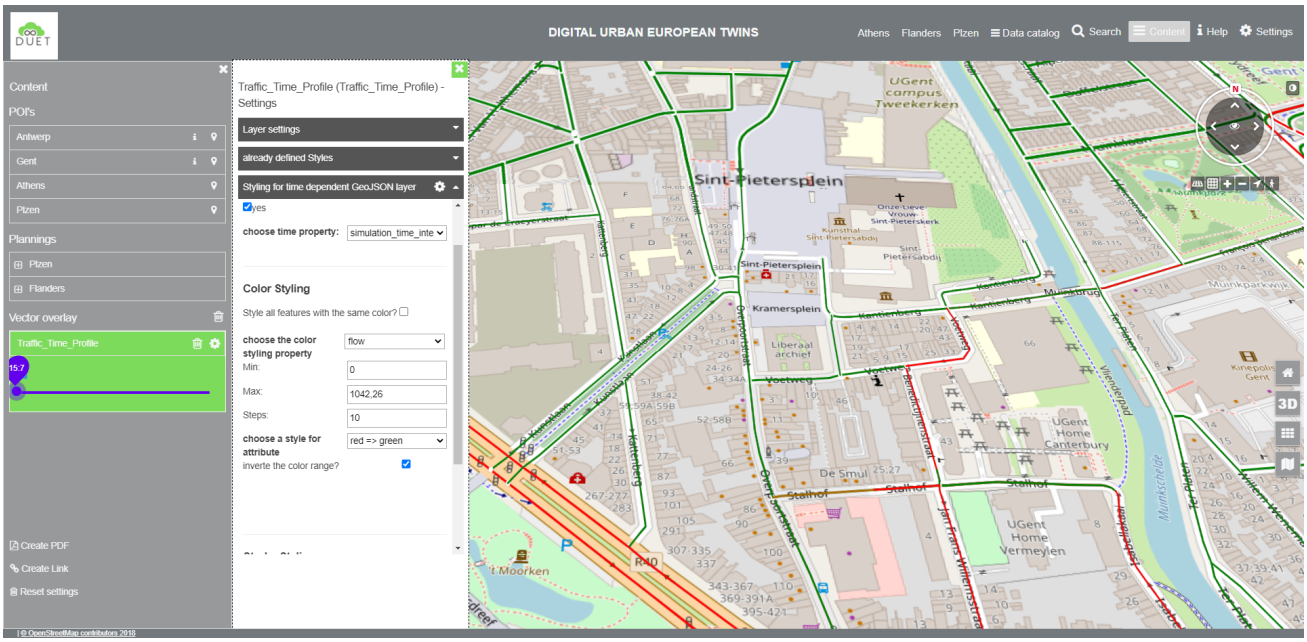


Figure 15: Applied user style to added time dependent traffic data

By clicking the button ***"Apply Style"***, the respective style will be created, and furthermore, since the user declared a time-dependent dataset, a time slider will appear below the dataset entry in the legend. Here the user can slide through the simulation time, and the style will be automatically applied to the features. This means the style selects the corresponding attribute array entry and will use that entry for styling. That works for colour styles and stroke styles, keeping in mind that the restrictions from Styling non-time dependent data based on individual attributes are still valid.

7 Visualization and inspection of individual features

Figure 15 & 16 show 2 different visualizations and two individual inspections of the respective features. Figure 15 inspects a traffic element, and the attributes are displayed on the left-hand side of the visualization

component inside a panel. The inspected feature is highlighted in green in the visualization component, and the marker points to the clicked point in 3D / 2D. Means the marker is placed in 3D on top of the feature. The proposed Balloon information of D4.1 is replaced by the left-hand side panel and the position indicator by the red marker. The individual content per feature is displayed in the information panel. Based on the feature to be inspected, the information representation can vary from a pure display of attributes (compare figure 17) up to integrated graph visualizations, as shown in figure 16 & 18. The developed API scans for specific attributes, such as speed, intensity, traffic, flow and integrates the graph visualizations using c3js [6], a javascript library based on d3js [7].

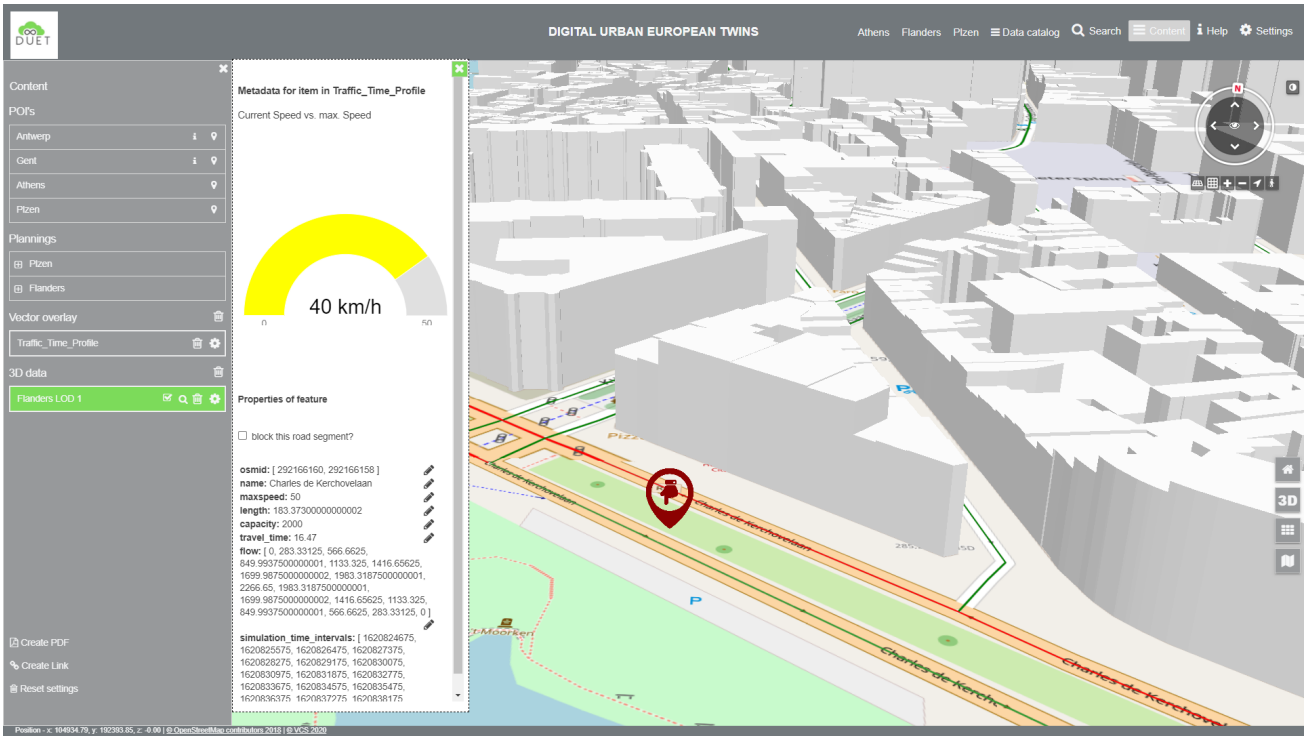


Figure 16: Individual inspection of traffic data

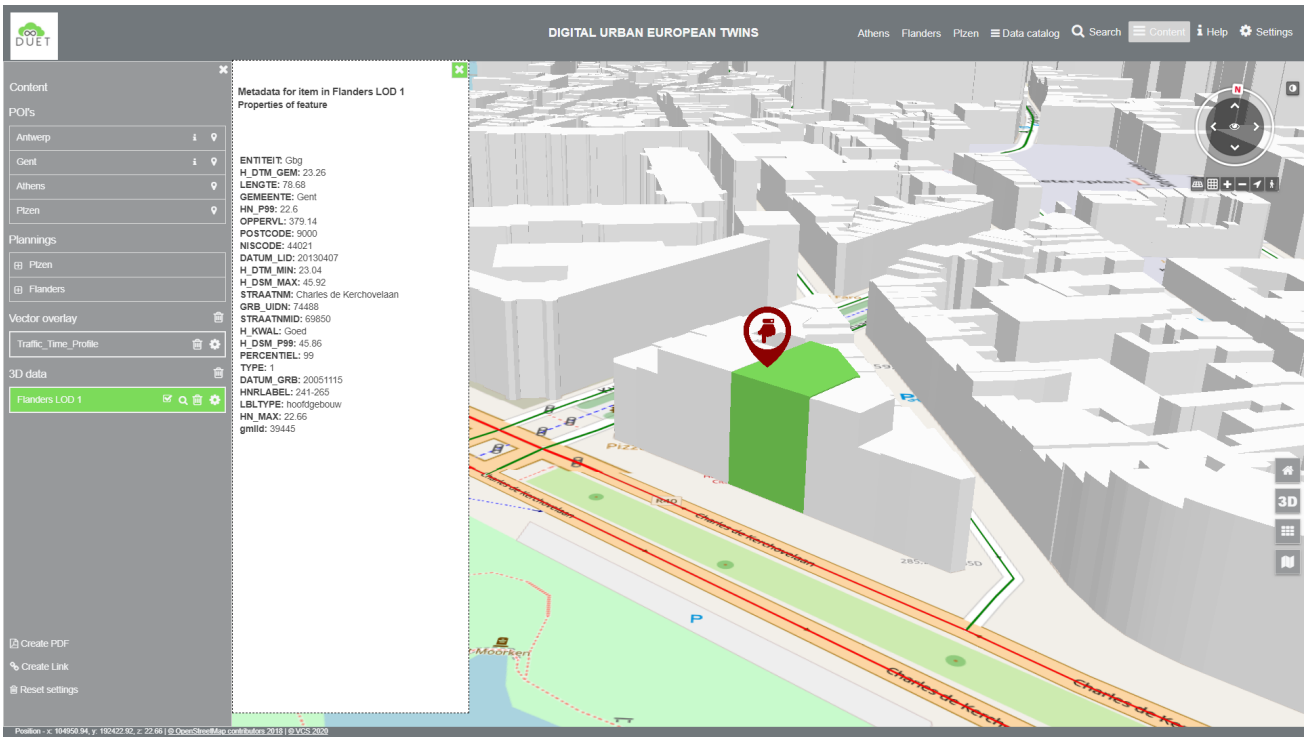


Figure 17: Individual inspection of 3D Building data

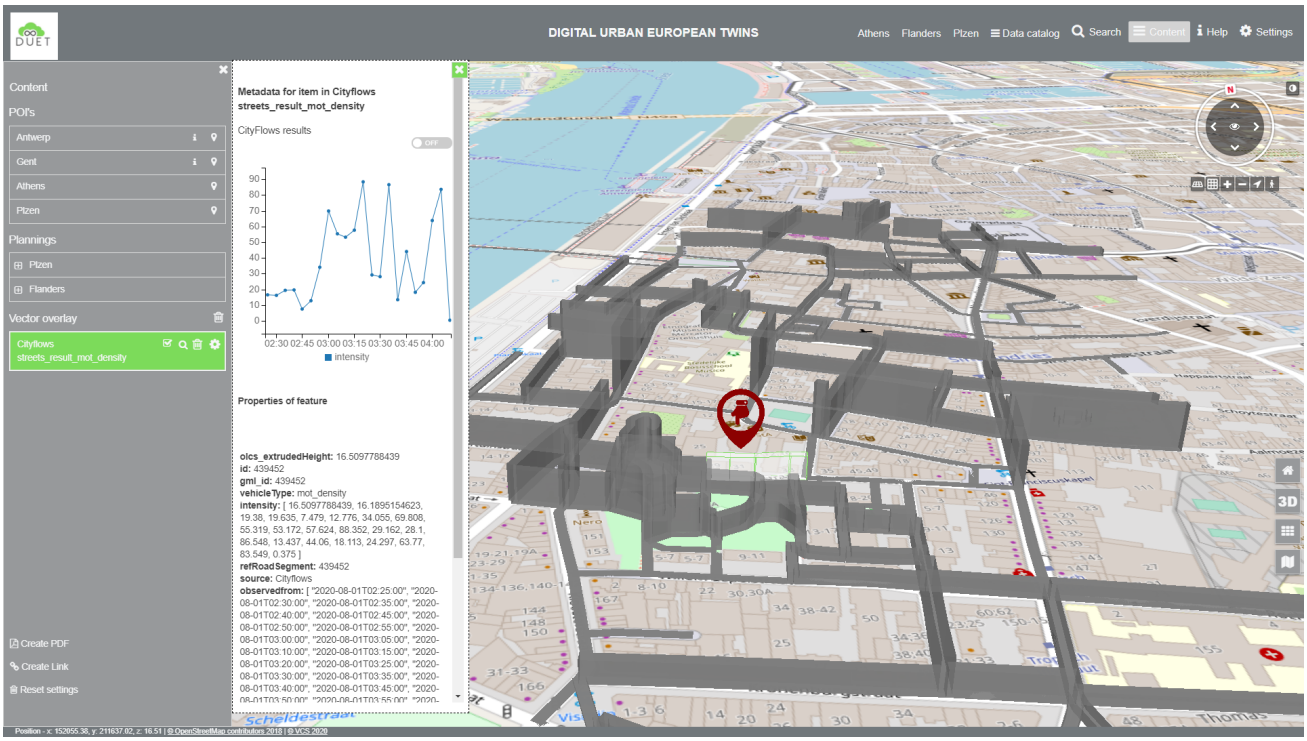


Figure 18: Individual inspection of cityflow data

8 Visualization of future states using the VC Planner

With VC Planner, urban planners and architects can integrate their designs into a 3D city model in the context of the surrounding development in just a few steps and make them available to other users. This makes the VC Planner suitable for various applications. For example, it can be used to present and compare urban planning and architectural competitions, urban planning designs can be created easily, and collaborative work with several participants in a project is possible. The VC Planner stores changes, objects, drawings etc persistent per project. Thus alternative or future states of a region, neighbourhood, city can be easily modelled and presented to the audience.

The main functions at a glance:

- **Projects and usage rights** - Creation of multiple projects with the assignment of different usage rights
- **Viewpoints and camera flights** - Create viewpoints and camera flights in a few clicks
- **Drawing** - Drawing tools for points, lines, polygons, circles, bounding boxes and text labels
- **hide/unhide** - Show and hide existing objects with a mouse click
- **Import per Drag-and-Drop** - Import urban designs, individual buildings and street furniture via drag-and-drop
- **Share and publish** - Share plans and 3D scenes with other users via a web link and/or publish them via a web browser.

Figures 19 & 20 show a bit of the potential using the VC Planner. In figure 18, the replacement of the church is done exemplarily to showcase the rebuild of a building or replacement by a high-res architectural building. Thus a reconstruction of a building, neighbourhood or district can be realized by the VC Planner. The original objects will be kept, just hidden from view once the planning is enabled. If the planning is disabled, the original state of the 3D scene is shown.

Figure 19: Replaced church in Plzen using the VC Planner (left), original LOD 2 data (right)

Figure 20 below shows the result of adding the Plzen Zoo by importing separate obj-files in the visualization client. This shows that even a complete neighbourhood recreation can be represented by using the VC Planner.

Figure 20: Added Plzen Zoo to visualization component via VC Planner

9 Performing Visibility analysis using Viewshed tool

The viewshed analysis allows the user to visualize object visibility from a specific location in 3D space. Objects within the observer's field of vision are illuminated, and objects that cannot be seen are shaded. Two different modes are offered. One is the directional mode, where the user determines the viewer's point of view and direction of vision, and the other is the 360° mode. In this mode, the user selects the point of view, and the analysis is performed 360° around this point.

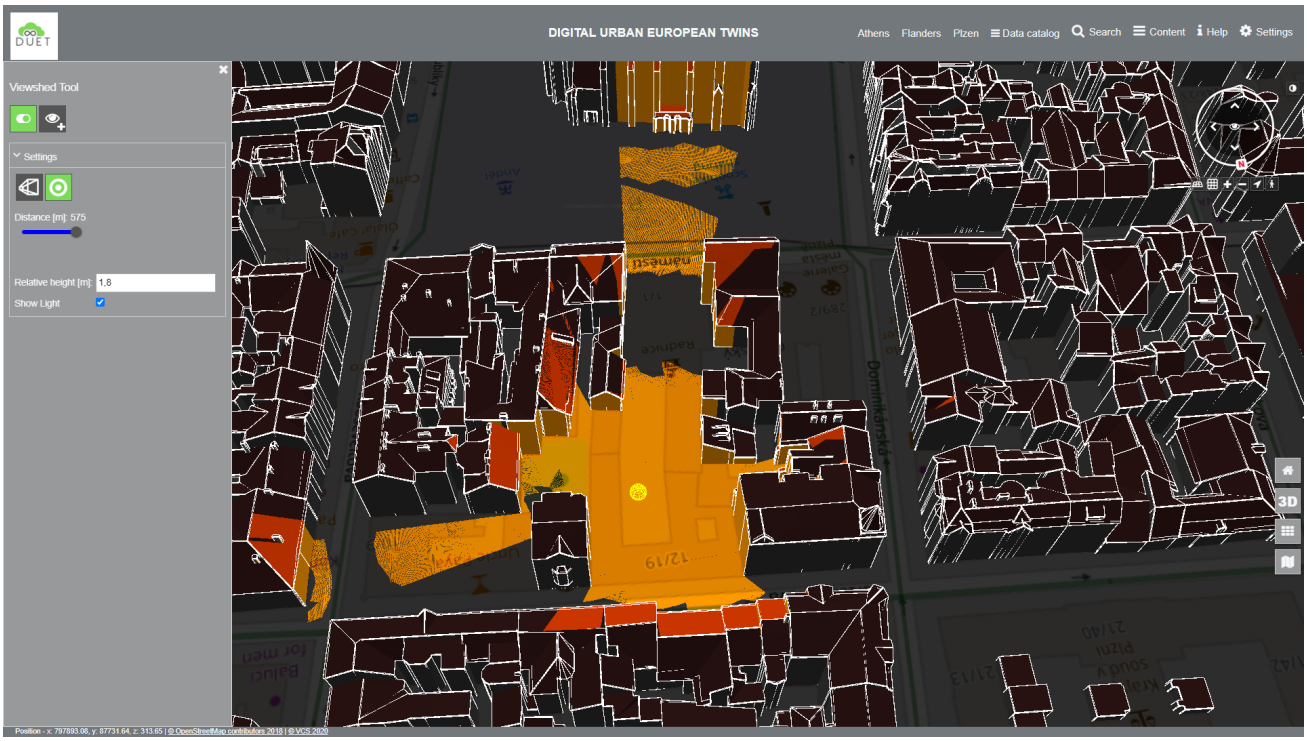


Figure 21: Visibility analysis in Plzen

10 Performing elevation analysis using elevation profile

The height profile tool allows the user to create a cross-section including height information between two points in the 3D map, compare figure 23.

The height profile is created with the button **New Profile**. The resolution and the precision of the height profile can be adjusted. Please note that the lower the resolution, the longer the calculation time of the profile. If the resolution is too low and the extension is too large, an error message will appear.

Under the selection item "Reference surface" you can select whether the 3D elements are to be included in the height profile or only the terrain is to be taken into account. The setting options can also be adjusted after the profile has been created.

The start and endpoints of the height profile can be defined and then calculated with two separate clicks. The profile can be moved later by selecting the green line in the map and clicking on the round symbol again.

Attention: Although the green line is set exactly to the 3D content, overhangs cannot be taken into account in the height profile.

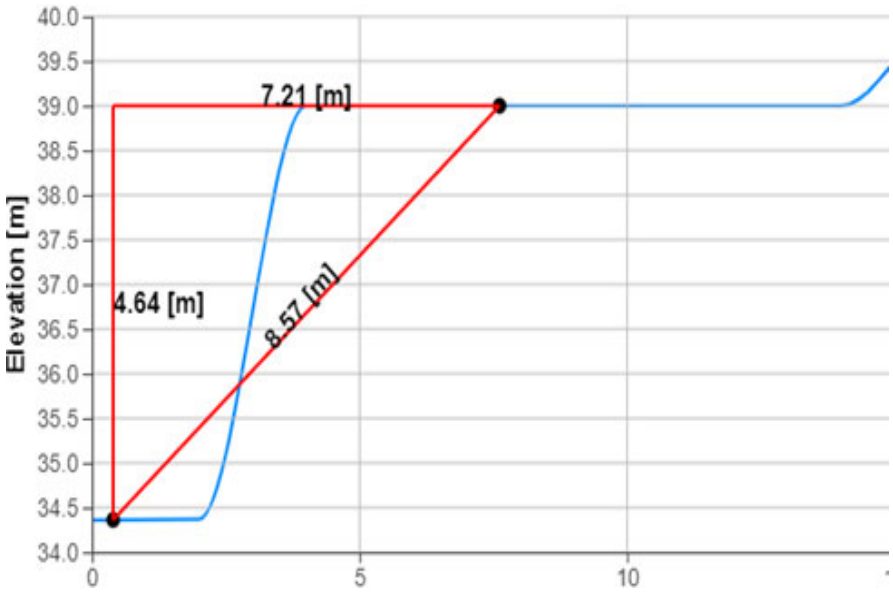


Figure 22: Title

Detailed measurements can be made in the output height profile by selecting two points between which the measurement is to be made:

Measurements taken can be replaced by selecting two other points or removed by pressing the **Delete Measurement** Button.

In addition, the output height profile, including the measurements performed, can be saved as an image by pressing the **Download** Button.

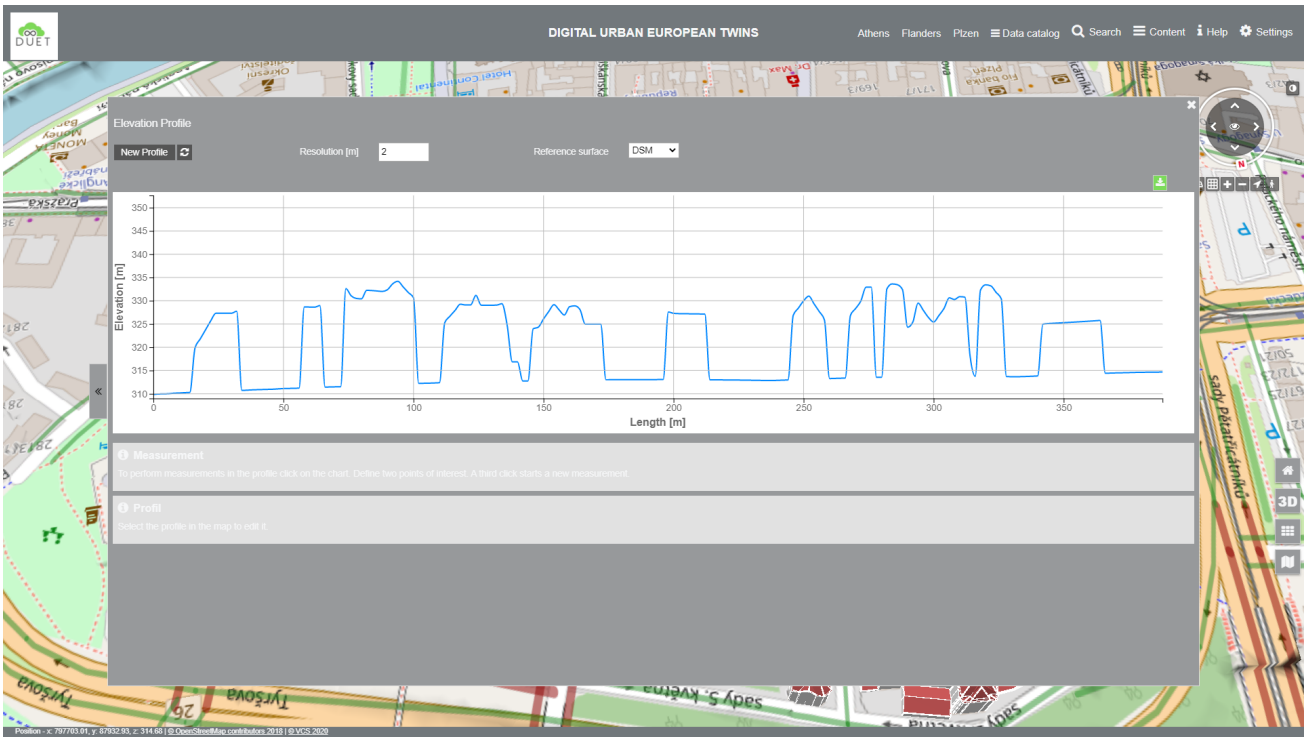


Figure 23: Elevation profile analysis in Plzen

11 Inspecting subsurface structures using Terrain cut

The Terrain cut mode allows the user to look in 3D view under the terrain, for example, to make elements visible under the surface. Activate the tool and position the box by clicking on the desired location. Click again to move the box.

12 2D / 3D measurements

With the Measuring tool, it is possible to measure directly in the VC Map. It can be activated by clicking the shown symbol.

All tools of the measurement tool deliver as a result the measured value related to the local coordinate system of the map, independent of the global coordinate system used in the viewer. The measurement tool can be provided in the Oblique, 2D and 3D maps. The tools vary depending on the map in which the measurement tool is used.

Measurement objects or the coordinates of the measurement objects (point, line, polygon) are projected into the local coordinate system of the application according to the map used (Oblique, 2D, 3D), and the measured value is output according to this coordinate system.

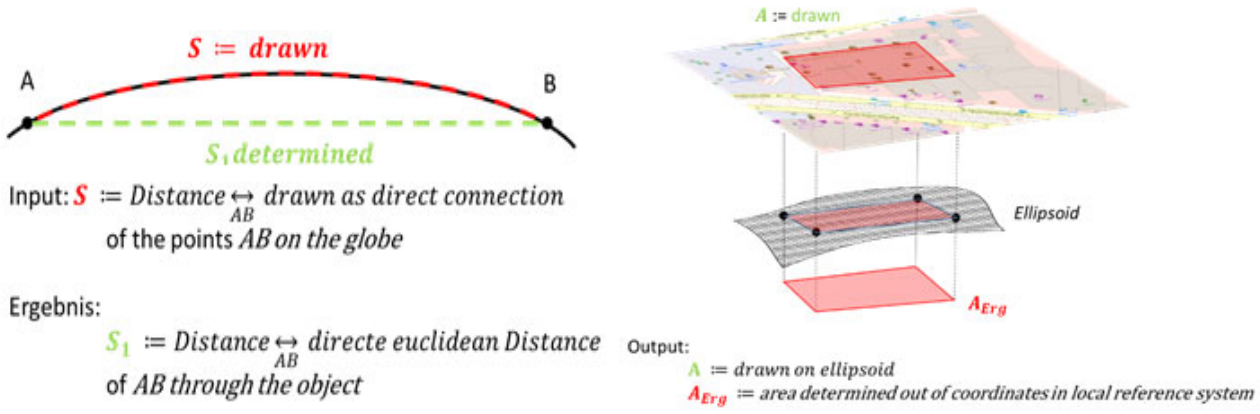


Figure 24: Measuring distance between objects

Measurements can be completed by double-clicking the endpoint or by clicking the "End measurement" button.

In all measurement modes, the measurements performed and the associated geometry objects are retained, even if a new measurement mode is selected.

Measurements and the associated measurement geometries can all be removed or deleted from the view by clicking on the trash. Individual objects can be selected by mouse click on the measurement geometry and deleted individually from the 'Delete' view. Deleted objects cannot be restored!

Selection of a measurement geometry: Left mouse click on the corresponding geometry/measurement

Deselection of a measurement geometry: Left mouse click somewhere in the map

Edit a measurement geometry: Select the measurement geometry and then:

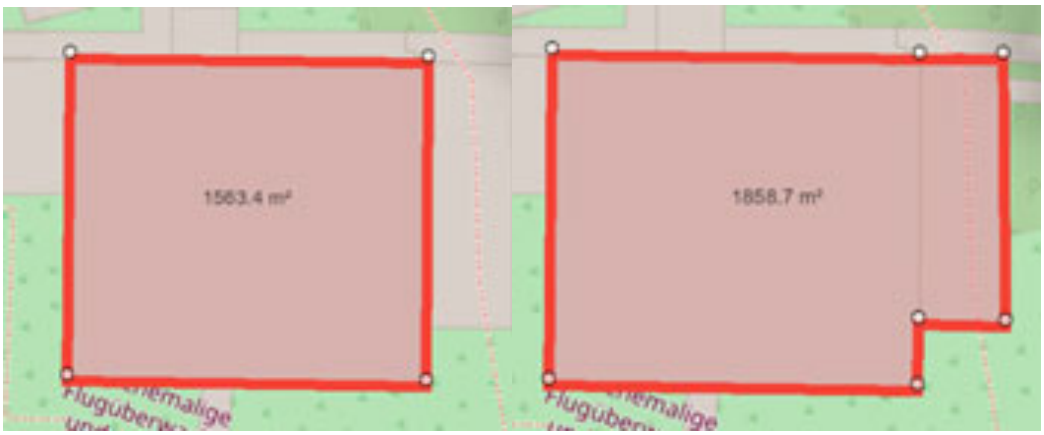


Figure 25: Measurement geometry before editing, Changed measuring geometry by adding 3 additional calibration points.

Add a new vertex:
Click on the corresponding geometry to create a new vertex on the geometry.

Move a vertex in any direction:
In 2D and 3D surface measurement mode - 'CTRL + ALT' + dragging with left mouse button at the desired position.
In 3D surface measuring mode under Linux - 'SUPER + ALT' + dragging with left mouse button at the desired position.

Vertical Shift of a vertex:
In 3D surface measurement mode - Left-click in the measuring field and press 'CTRL' + drag the measuring point with the left mouse button to the desired height.

Delete a vertex:
In 2D and 3D surface measurement mode - 'Shift' + click on vertex to be deleted.
(Important note: The measurement has to be finished before the delete operation!)

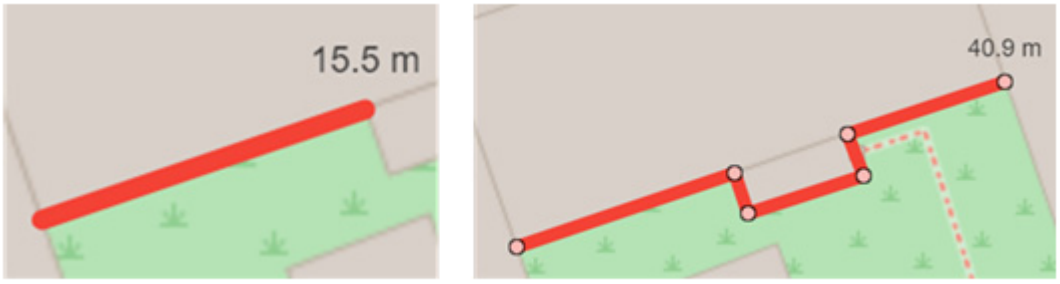
12.1 Measuring tool in the 2D map

The 2D map contains the measuring tools:

- Horizontal distance measurement
- Surface measurement

Multiple measurements are possible, and the measured values together with the measurement geometry are retained.

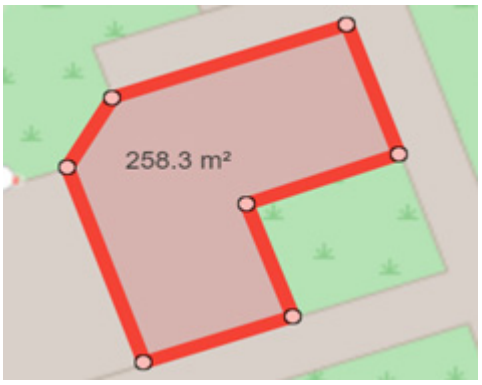
The following measurement modes are available for horizontal distance measurement:



Direct distance between two points *Line measurement as polyline*

Figure 26: Distance and line measurement

The distance measurement follows the principle of the Euclidean distance between two points in the plane or space and is, for example, the length of a distance measured with a ruler that connects these two points. On a two-dimensional plane or in three-dimensional Euclidean space, the distance between two points is the length of the shortest connection between these points. In Euclidean space, this is the length of the straight line between the two points. Therefore, the distance between two geometric objects is the length of the shortest connecting line between the two objects, i.e. the distance between the two nearest points.



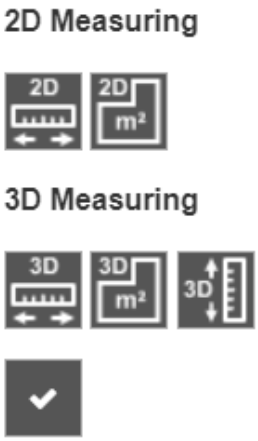
The following measurement modes are available for 2D surface measurement:

The result of the area determination in 2D returns the area and the circumference of the drawn object related to the local coordinate system of the map (here: EPSG:25833).

Area:	258.3m²
circumference:	72.4m

Figure 27: Surface measurement

12.2 Measuring tool in the 3D map



The measuring tool of the 3D Map includes the measuring tools of the 2D Map and extends them by 3D measuring tools, which make it possible to measure 3D objects.

2D Measuring Tools in the 3D Map

The description of 'Measuring tool in the 2D map' applies, whereby it should be noted that in the 3D Map, the 2D measurements always follow the terrain, but the result ALWAYS refers to the local coordinate system of the application.

2D distance measurement in the 3D map:

Direct euclidean distance of two points in 3D considering the terrain

Area measurement in 2D under consideration of the terrain

Figure 28: 3D Distance measurement

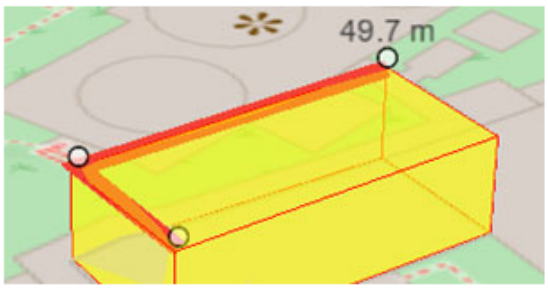
3D measuring tools in the 3D map

The measuring tools in 3D are extended by the height measurement and allow to directly include 3D objects in the measurement or to measure them directly.

The following measuring modes are possible with the 3D measuring tools:

Distance measurement in 3D:

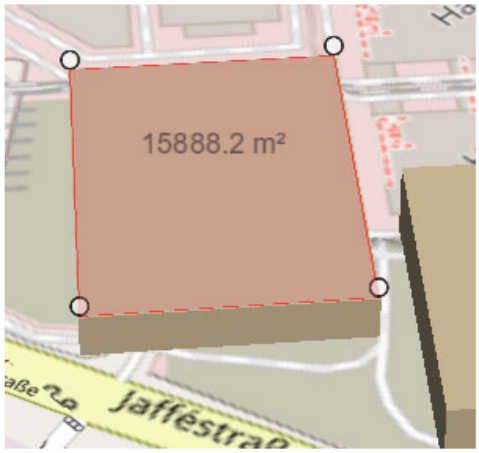
Direct euclidean distance between two points in 3D considering the terrain and the 3D objects



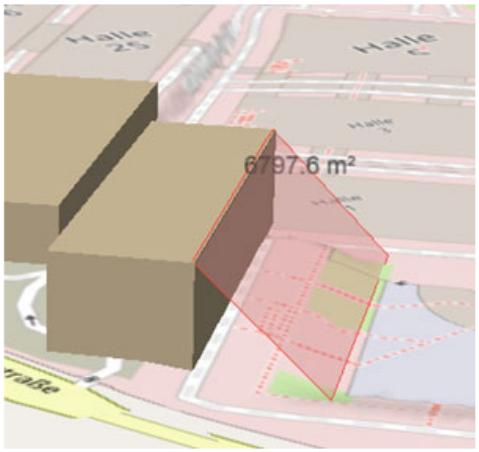
Euclidean distance measurement on the 3D object

Figure 29: Euclidean distance measurement

Area measurement in 3D:



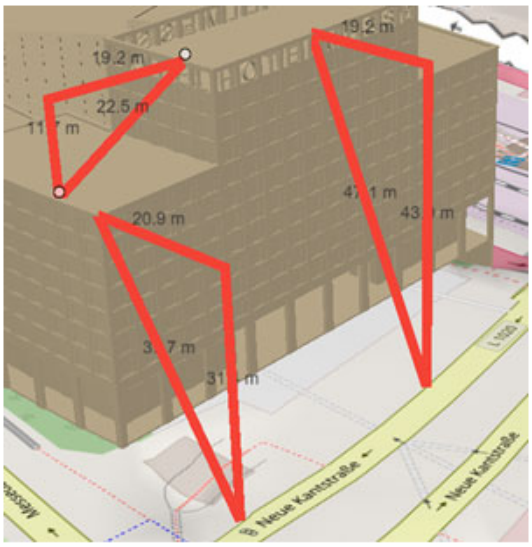
Area measurement on the 3D object



Area measurement on the 3D object up to the terrain

Figure 30: 3D Area measurement

Height measurement in 3D:



Height measurement on / with the 3D object up to the terrain

Figure 31: 3D Height measurement

13 Line-of-sight analysis using extended pedestrian mode

The extended pedestrian mode allows the user to define a visual analysis between two freely selectable points (Line-of-sight). After activating this mode, the first click on the map defines the start position, and the second click defines the view direction. This function can be used to find out which objects a user is able to see from a certain point of view.

This mode can be left by deactivating the symbol in Control-Elements or underneath the Navigation-Control.

14 Drawing tool

The drawing tool allows the user to draw simple as well as complex objects in the context of the environment model. It can be activated by clicking on the icon shown in the tool selection.
The drawing tool is divided into three areas.

In the upper part, the different drawing modes can be selected via the corresponding buttons.
The buttons for object modification, transformation or import/export interface are located directly below.
Below the buttons are the tabs where the settings for the different styles and manifestations can be set.

Figure 32: Drawing tool interface

Drawing objects

Objects can be drawn in different ways. Besides the 'classic' geometry types like point, line and polygon, the drawing of bounding boxes, circles and rectangles is supported. The drawing mode for the desired geometry type is set via the corresponding selection in the upper part of the drawing tool.

- **Points** are set by simply clicking on the desired position on the map.
- **A line** is drawn by setting its supporting points. To close the line, the last fulcrum is set with a double-click. Alternatively, the drawing can be stopped by clicking the hook button.
- **A polygon** is drawn similar to a line. However, if the last polygon point is set, the polygon is automatically closed by a line to the first polygon point.

- **Circles** are drawn by setting a circle centre point in a first click and an additional second point indicating the radius.
- **Bounding boxes** are created by setting the upper left and lower right corners. They are always oriented to the north.
- **Rectangles** with any orientation can be drawn by three points. The first two points indicate the direction. The third point can be used to define the width and the height of the rectangle based on this 'baseline'.
- **Text labels** are positioned by setting a point in the map.

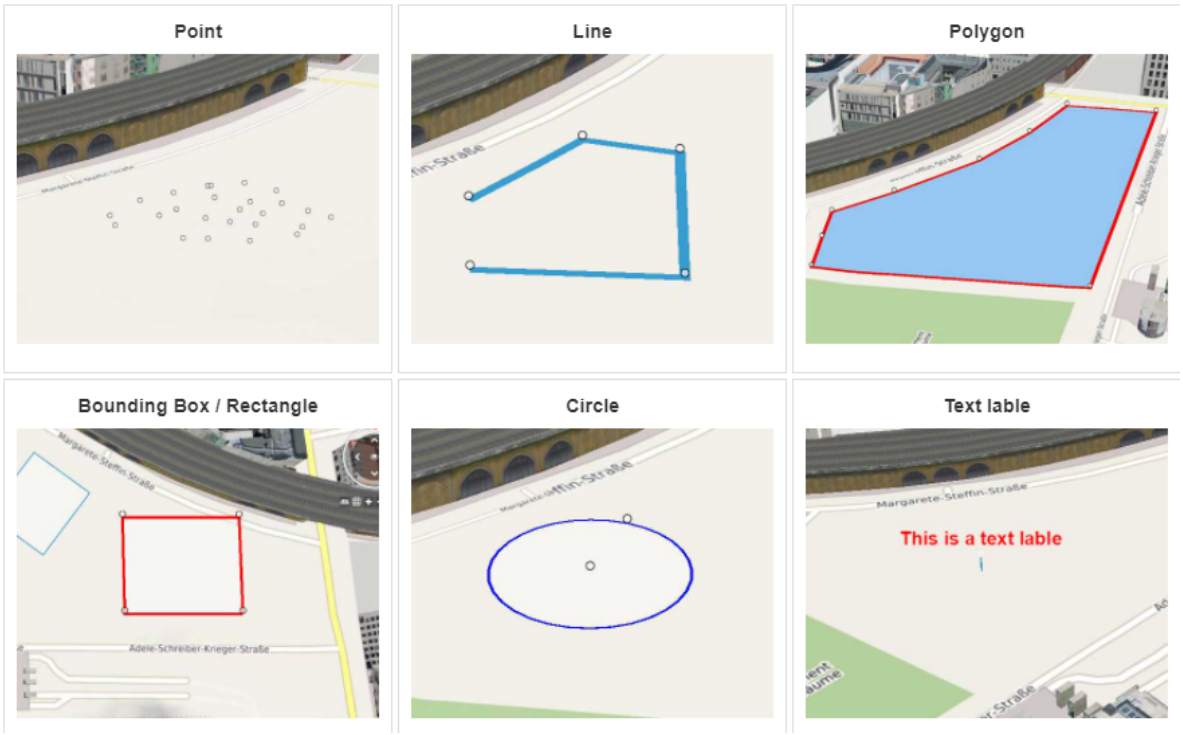


Figure 33: Drawing visualisations

Altitude mode

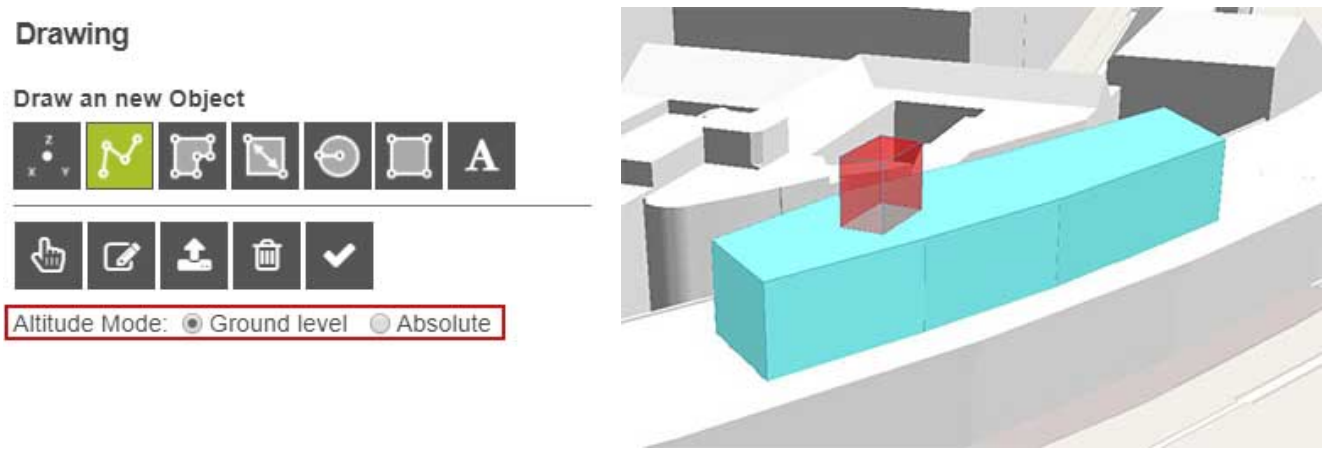


Figure 34: Object drawing

Two height modes are available for drawing objects, which can be selected via the radio buttons. In the 'Ground level' mode, the objects are drawn directly on the terrain. In 'Absolute' mode, objects can be drawn on existing objects.

Edit and Transform Objects

Two modes are available for editing objects. The 'Hand button' takes you to the mode for moving and transforming objects. The second button takes you to the mode for editing objects. After a mode has been selected, objects can be selected in the map and edited according to the functions of the mode.

The individual functions of the modes are described below:

Figure 35: Individual modes functions

Convert to Polygon

When drawing rectangles or circles, the user has the option to convert these element types into a polygon. This can be done by clicking on the object with the editing tool enabled and selecting the 'Convert to Polygon' option.

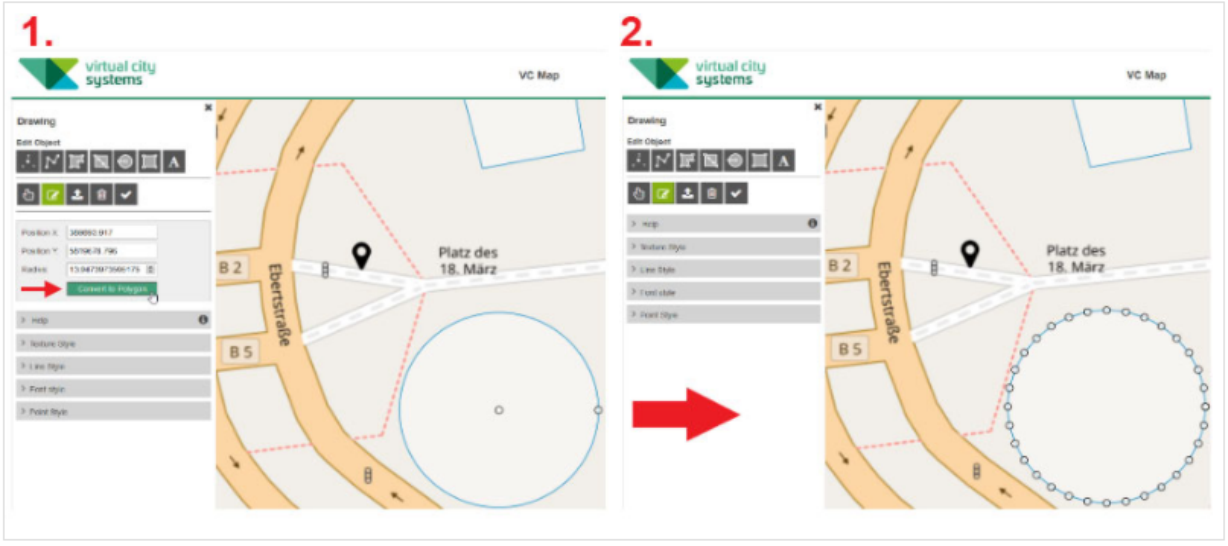


Figure 36: Polygon convertion

Extrusion

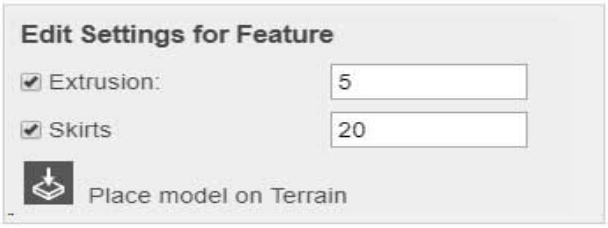


Figure 37: Extrusion settings

In order to generate solid bodies from flat objects, these can be extruded at any height. To do this, an object is selected in edit mode. The button for specifying the extrusion height appears.
In some cases it can be useful for visualization purposes not only to extrude objects in height, but also to extend them in depth. Especially on steep slopes 'floating' buildings can be avoided. The extrusion depth can be edited by altering the height of the 'skirt'.
All objects can also be placed on the terrain.

Import / Export

Figure 38: Import/Export functionality

Drawn objects are only valid for the current browser session. To persist these, objects can be exported in GeoJSON format and then re-imported. It is possible to export only the currently selected objects or all drawn objects.

Styles (texture, line, font, point)

The appearance of the geometries can be set before drawing. Geometries can subsequently be changed in the editing mode.

Texture Style

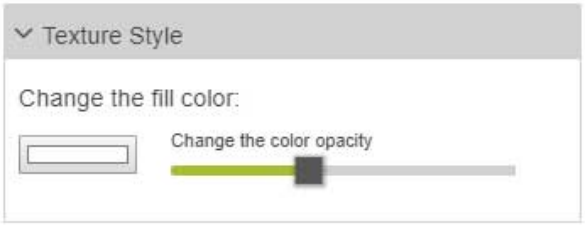
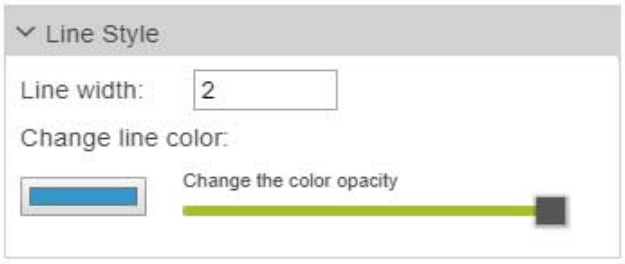


Figure 39: Texture style functionality

In the 'Texture Style' tab you can determine the opacity color and the opacity of the areas. In addition to predefined basic maps, any color tone can be selected via RGB values.

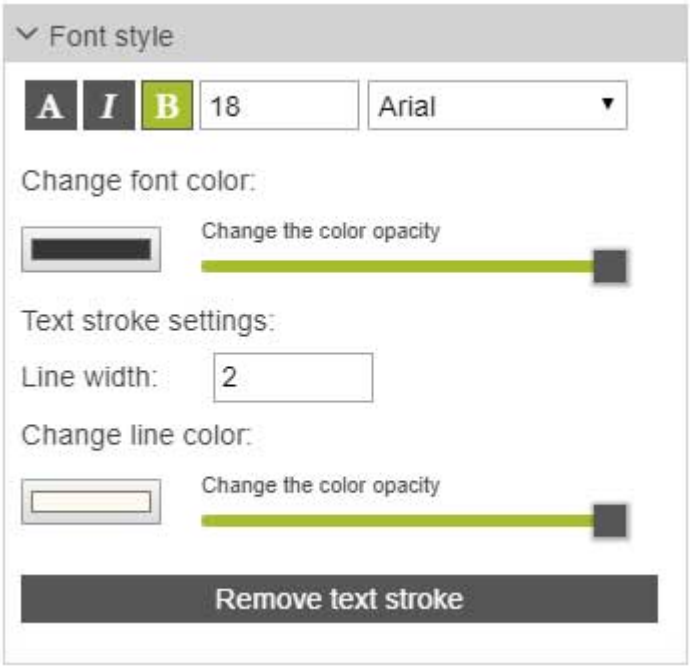
Line Style



For line geometries, line thickness can be specified in addition to line color and opacity.

Figure 40: Line style functionality

Text Formatting

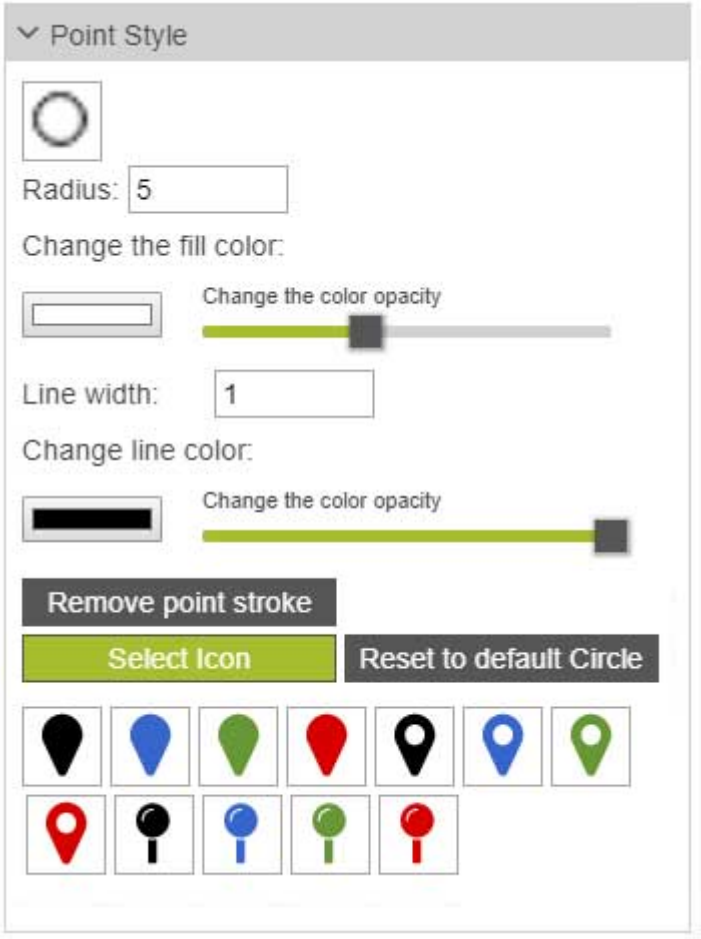


Text labels can be created in different fonts and sizes. The text can be displayed in bold, italic or normal. The text color and its opacity can also be defined individually.

It is possible to surround the text with a border. This can be activated or deactivated by pressing the "Remove text border" button. A choice of colours is also possible here.

Figure 41: Text formatting functionality

Point Style



For points, the desired radius, fill color and line border color can be determined. Optionally the line border can be set. The "Select Icon" button can be used to use predefined symbols as point markers.

Figure 42: Point style functionality

15 PDF Export

A PDF of the application can be created in the tool menu or under Contents, which contains the current view and, depending on the configuration, can be enriched with additional information in the backend.

Please note that the section is created independently of the screen/window size. Instead, it is created in 16:9 format starting from the center of the view. Therefore, it is possible that the view of the resulting PDF differs from the displayed view.

Example of PDF Export Settings
For the **Paper Size**, the DIN sizes A5 to A2 can be selected.

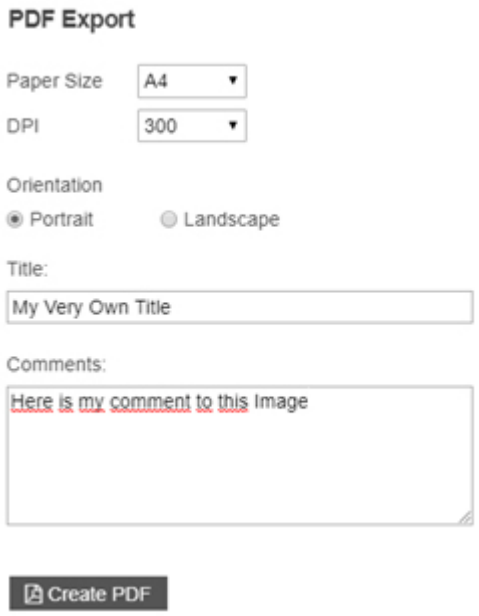


Figure 43: PDF Export functionality

- The **DPI (Dots Per Inch)** number determines the resolution of the map section. Please note that a value of 4096 x 4096 pixels (Internet Explorer 2048 x 2048 pixels) cannot be exceeded. Means, for example, that a set page size of A2 landscape format in combination with 600 dpi cannot be output. Instead, the highest possible resolution level is used.

Both the format and the DPI number have an influence on the resulting size of the PDF and on the creation time.

The **Orientation** can be chosen between portrait and landscape format. The size of the image is correspondingly smaller when the portrait format is used.
The pixel size resulting from the default setting (portrait, A4, 300 dpi) is 2055 x 933 pixel.
The combined setting landscape, A4 and 300 dpi results in a size of 3402 x 1370 pixels.

If a **Title Input Field** has been configured in the backend, any title can be entered here. If this is left empty, the title of the application is used.

If an **Annotation Field** has been configured in the backend, annotations can be entered in this field.

By clicking the **Create PDF** button, a PDF is directly created and downloaded and named map.pdf.
By clicking the **Create Screenshot** button, a screenshot is directly created, downloaded and named screenshot.png

Notifications
The DUET visualization component makes use of so-called TOAST Notifications to inform a user about events happening within the environment. A **Toast** is a non-modal, unobtrusive window element used to display brief, auto-expiring windows of information to a user. A Toast is displayed on top of the main content of an activity and only remains visible for a short time period.

The Duet visualization component uses the "***Toastr***" library [https://codeseven.github.io/toastr/] to display the relevant notifications.

Duet visualization component will inform a user via Toastr of the following events:

1. failure or success by saving or updating a case
2. failure or success by posting a change interaction to the interaction API
3. when a new simulation result is available
4. failure or success by retrieving data from external sources

In general a SUCCESS Notification will be displayed in green,

whereas a failure message will be displayed in red.

Figure 44: Notification messages

16 Conclusion

In this report, an overview of the results of the DUET Digital City Twin frontend components is presented. This report reflects the as-is developments and visualizations in DUET for the Closed Beta version released at the end of May 2021.

Of course, further improvements and integration of functionalities will be made in the next public-oriented open beta release, even if they are not directly epic related.

Future steps will be definitely in the direction of creating a report of the current state of the visualization. The first step in this direction is the PDF creator that creates a visual report of the current scene. However, some more information should be integrated into it, such as used datasets/models, timestamps, used simulation results etc.

17 References

1. OGC 3D Tiles Standard - https://www.ogc.org/standards/3DTiles
2. 3D Tiles overview - https://github.com/CesiumGS/3d-tiles/blob/master/3d-tiles-overview.pdf
3. Nominatim Open-source geocoding https://nominatim.org/
4. w3.CSS - CSS framework - https://www.w3schools.com/w3css/defaulT.asp
5. Toastr - https://codeseven.github.io/toastr/
6. C3.js | D3-based reusable chart library - https://c3js.org/
7. D3.js is a JavaScript library for manipulating documents - https://d3js.org/