

# The Kalasatama Digital Twins Project

The final report of the KIRA-digi pilot project



- 1. THE STARTING POINT OF THE PROJECT .....3
  - Definition of a Digital Twin
- 2. THE OBJECTIVES OF THE PROJECT .....4
  - Intermediate Objective 1: Production of the Twin Models** .....5
    - Reality Mesh Model** .....6
      - Data Collection for the Mesh Model .....7
        - Aerial Photography
      - Preparatory Work for Modelling the Mesh Model .....8
        - Model Coverage and Tiling
        - Ground Control Points
        - Waterbodies
    - Production of the Mesh Model .....10
      - Processing the Source Data
      - Automatic Aerial Triangulation
      - Model Calculation
  - CityGML City Information Model** .....12
    - CityGML Thematic Modules and Classes .....13
    - CityGML v2.0 Levels of Detail .....14
    - Production of the CityGML v2.0 Model .....15
    - CityGML Buildings .....16
      - Modelling the Buildings
      - Maintenance of CityGML Buildings at Helsinki City Survey Services
    - Modelling the Future Buildings and Environment .....19
      - Planned Models
      - Modelling Planned Buildings into the CityGML format
  - CityGML WaterBodies .....23
    - Modelling the CityGML Waterbodies on a Theoretical Level
  - CityGML Bridges .....24
    - Modelling Bridges into the CityGML Format
  - The CityGML Terrain Model .....27
    - Modelling the CityGML Terrain Model on a Theoretical Level
    - Production of the Terrain Model
  - CityGML Land Use Areas .....29
    - Modelling the CityGML Land Use Areas on a Theoretical Level

- Intermediate Objective 2: Sharing the Twin Models as Open Data** .....30
  - Twin Models as Open Data .....30
    - The Kalasatama Mesh Model
    - The Mesh Model in the Unity Streaming Service .....31
    - Information System Infrastructure and Architecture of CityGML Model .....33
      - The Database Dump of Kalasatama’s CityGML Model
- Intermediate Objective 3: Twin Models as a Smart Development Platform** .....35
  - Smart Kalasatama .....35
    - The Smart City Platform
    - Creating the Platform
  - Smart Kalasatama Projects and Innovations .....37
  - Smart Interaction .....38
- Intermediate Objective 4: Twin Models as a Simulation Platform** .....39
  - Wind Simulation .....39
    - Exploring the Speed and Direction of the Wind
    - Implementation of the Wind Simulation
  - Sun Study: Solar Hour Analysis .....49
  - Sun Study: Shadow Analysis .....52
- Intermediate Objective 5: Twin Models in City Processes** .....54
  - The Use of Digital Twins in City Processes and Service Production – Helsinki’s Digitalisation Programme .....54
    - From the BIM Model to a CityGML Model – Overview of Research and Development .....55
      - Summary of the National University of Singapore’s Research and Development Work
      - Summary of Delft University’s Research and Development Work
      - The Simplebim® and CityGML Conversion Add-on
    - Conversion Experiment from an IFC Model to a CityGML Model .....58
    - UAV Drone Experiment .....59

**3. RESULTS: OBSERVED BENEFITS, EFFECTIVENESS AND DEVELOPMENT NEEDS .....60**

# THE STARTING POINT OF THE PROJECT

## Definition of a Digital Twin

“Quite simply, a digital twin is a virtual model of a process, product or service.” This pairing of the virtual and physical worlds allows analysis of data and monitoring of systems to head off problems before they even occur, prevent downtime, develop new opportunities and even plan for the future by using simulations. **Bernard Marr/Forbes 2017**

The digital twin concept is not new. A digital twin refers to the digital representation of a physical object and has been used in computer-aided design for over 30 years. For instance the first 3D city models of Helsinki were created in 1987 representing the Kamppi area.

For three years in a row Gartner have listed digital twins as one of the top 10 strategic technology trends of the year, as shown in Figure 1.

Bolton, McColl-Kennedy, Cheung, Gallen, Orsingher, Witell and Zaki (2018) define digital twins as “a dynamic virtual representation of a physical object or system across its life-cycle, using real-time data to enable understanding, learning and reasoning”. They see digital twins playing an important role in the Fourth Industrial Revolution.

However, using a digital twin as a city model is an exceedingly new concept. In product-, equipment- and plant design digital twins have been used for decades. According to John Vickers at NASA, the ultimate vision for a digital twin is to create, test and build everything first in a virtual environment. This definition is still relevant.

The use of digital twins in a city concept is a larger vision compared to industrial design. For instance, the Smart City and digital twin concepts of are interlinked on many levels, ranging from the Internet of Things, the interactivity between digital twins and the real world, sensor systems, simulations and analysis of present and planned city environments, maintenance and administration systems, emergency planning and management, and the construction production chain, to mention a few.

The digital twins in the Kalasatama project in Helsinki offer high quality 3D city models as open data to all operators. The project team hopes that these city model platforms will promote diverse product development, research, teaching and innovation.



Figure 1. TOP 10 strategic technology trends for 2019 by Gartner [<https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2019>]

# THE OBJECTIVES OF THE PROJECT

The general objective of the project was to produce high quality digital twin city models of the Kalasatama area and to share the models as open data. The models serve as a platform for designing, testing, applying and servicing the entire lifecycle of the built environment. The progress of the project was divided into five intermediate objectives. The general objective of producing the models was the first.

The sharing of the 3D city models as open data was the second objective in the project. There are hundreds of applications for city models, but the lack of a suitable size and detailed development platform have been a bottleneck in the development. The objective of the second milestone in this project is to help eliminate this bottleneck and allow the development of the platform through open source tools.

The third objective was focused on cooperation with the main partner, the Smart Kalasatama project. The digital twins are part of the Smart Kalasatama project which is a platform for innovation and development. In the project, an online platform for activities in Kalasatama and interaction with the residents was built on the 3D model platform.

The fourth objective was to try out the latest ways to model, test and utilize 3D city models. In particular, simulations and analyses with CityGML models are undergoing a strong phase of development. This intermediate objective applies the basic idea of digital twins: “design, test and build first digitally.”

The fifth objective was to promote the exploitation of digital twins in city processes and service production. An accurate, up-to-date model of an existing city structure and the future plans will enable the development of processes, practices and services based on 3D technology.

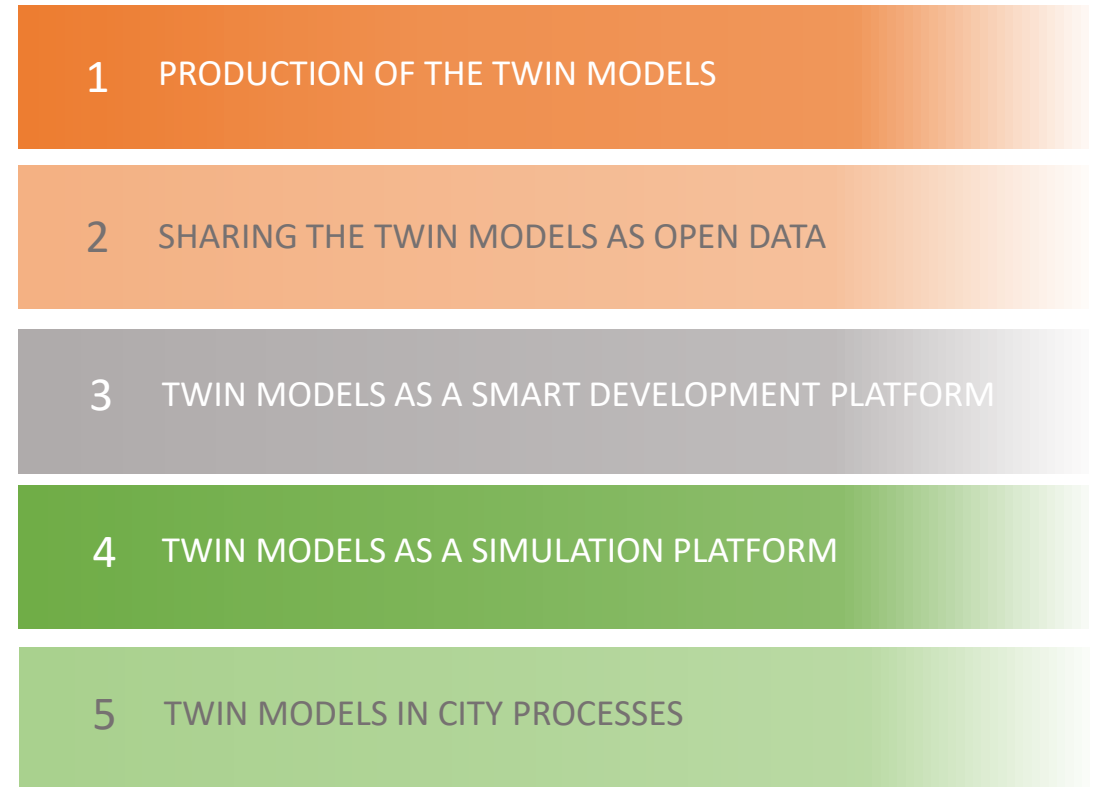


Figure 2. Intermediate objectives of The Kalasatama digital twins project

## Production of the Twin Models

The principal objective of the project was to produce high quality digital twin city models of the Kalasatama area. Digital twins work as a design-, testing-, application- and service platform for the entire lifecycle of the built environment, as well as for smart urban development.

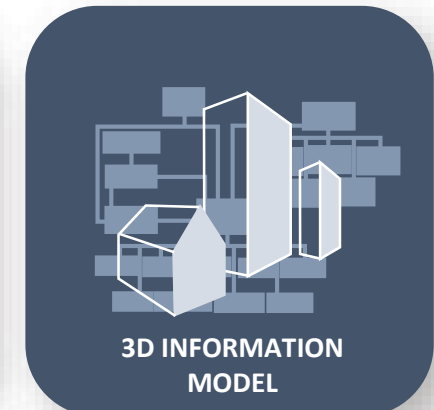
The digital twins were produced with the latest modelling technologies and applications as a **CityGML standard-based semantic city information model** and a **reality mesh model**. The Kalasatama twin models were produced as separate works for this project, and their source data, content, and accuracy are of higher quality than the previously modelled 3D city models of Helsinki.

A reality mesh model is a photorealistic city model that creates a visual geometric model which matches the reality of a city based on aerial photographs. Producing a mesh model even for large cities is cost-effective as the final production process of the model is based on highly automated computing. However, calculating a city-wide model will take several months to run on an efficient computer farm. The strength of the mesh model is its visual information that contains all items that are mostly stationary at the time of shooting. Reality mesh technology can deliver highly detailed 3D models. The accuracy of the source data and availability of the computing power set the most significant constraints on the model production.

The CityGML model is a semantic city information model according to the Global OGC (Open Geospatial Consortium) standard (CityGML v2.0). In continental Europe, CityGML models are produced by several major cities, and Singapore and New York also use CityGML models. The 3D modelling of the national built environment by the National Geographic Survey of Finland is also based on CityGML.

The strengths of the CityGML model are that it is an open global standard and open information model, that it has the Application Domain Extension (ADE), and that numerous and evolving open source tools for creating the information model environment are available. In addition, the standard is supported by a wide range of global developers, including universities, research institutes, states, cities, and a growing range of utility applications.

Prior to the project, the most important source data for both models (reality mesh model and city information model) had already been acquired from aerial images and point cloud data sets from 2017, aerial photography and laser scanning, as well as registries and spatial data. The first objective in this project focused on the model creation process and on the modelling of planned buildings, which is an essential principle behind the use of digital twins.





## Reality Mesh Model

The reality mesh model, is a model made up of a uniform surface of triangles. In itself, the mesh refers to the presentation of a geometric model that can be implemented by means of a uniform network of triangles, squares, or polygons. In this project, a triangular mesh model was created. The triangular mesh model consists of plane surfaces of triangles, which in turn consist of faces and vertices. The vertex (later referred to as a tie point) of each plane surface also belongs to the adjacent plane surface, whereby the entire triangular mesh creates one uniform surface. The triangular mesh structure is shown above. The mesh model is an approximation of the actual object shape with a certain level of accuracy. The accuracy of the shape of the reality mesh depends on the density of the triangulated network and thus on the size of the triangles. The smaller the triangles in the model, the more specific the model is.

Typically, mesh models are used in different modelling programs to create and represent the shape of the object to be modelled. Commonly, mesh models are also used when there is a need to visualize an object or produce a base model for the object's design.

In calculating the mesh model, the source data consists of several partially overlapping photographs of the object to be modelled, and the resulting model looks realistic, which makes it possible to speak of both a photogrammetric and a photorealistic model. The production of the reality mesh model can be divided into three phases: data collection, preparatory work and production.

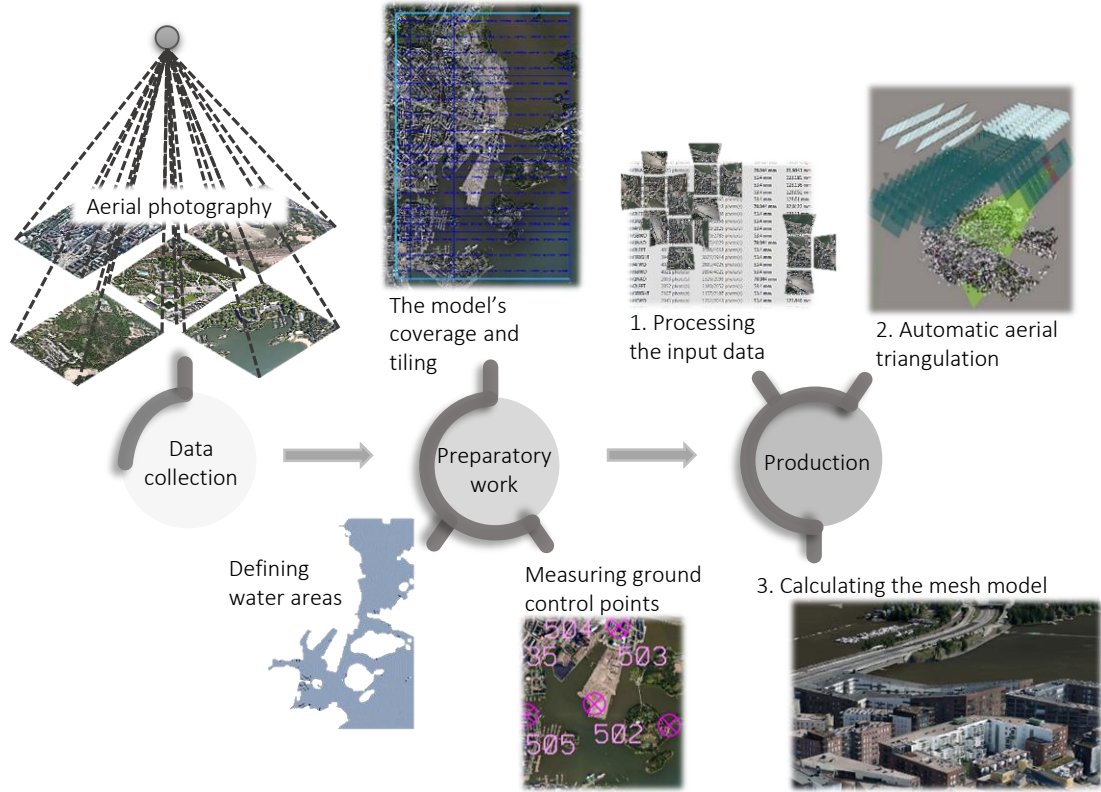


Figure 3. The production work flow of the mesh model

## Data Collection for the Mesh Model

### Aerial Photography

Data collection for the mesh model begins with aerial photography. The source data for the Kalasatama mesh model were aerial photographs prepared in the summer of 2017. The aerial photography was performed by Blom Map Oy using an [UltraCam Osprey Prime II](#) five-camera imaging system consisting of five calibrated metering cameras, one of which was a nadir camera with four other oblique cameras. The oblique aerial images were taken at an angle of 45 degrees, allowing precise resolution of vertical surfaces such as the facades of buildings. The coverage ratios of the aerial photography block were as follows: the side coverage of the flights was 60 % and the length coverage 80 %. The aerial photography was carried out from an airplane with an altitude of about 1.2 km. The accuracy of the images, i.e., the ground sampling distance (GSD), was about 7.5 cm. Three channel RGB (red, green and blue) images were recorded. The whole city of Helsinki was filmed in five flights during the period 27.7. - 14.8.2017 and 42,810 images were produced. The shooting area covered not only Helsinki but also areas in Espoo, Vantaa and Sipoo. In addition to creating a mesh model, the aerial photographs were used to texturize the city information model and to produce both stereo and ortho images.

In order to create a mesh model for the Kalasatama area, the aerial photographs taken were limited to about 12 square kilometres. In the Kalasatama modelled area, 2,083 aerial photographs were used. Each image also had location and position information using the ETRS-GK25 / N2000 coordinate system.



Figure 4. UltraCam Osprey Prime II five-camera system used for vertical and oblique aerial photography [Vexcel Imaging, [https://ultracam.files.wordpress.com/2014/04/ucop\\_2.png](https://ultracam.files.wordpress.com/2014/04/ucop_2.png)]

## Preparatory Work for Modelling the Mesh Model

### Model Coverage and Tiling

The mesh model was produced using a 3D modelling application that is capable of producing 3D models almost automatically covering entire cities. However, computing large 3D models requires a lot of graphics processing computing power as well as computer memory. Therefore, the computing is done in smaller areas to lighten the process. To do so the 3D modelling area should be tiled, i.e., divided into smaller areas that are generated one at a time.

Part of the source data for the mesh model was the area boundary of the project area and its tiling, which is shown in Figure 5. The project area was tiled and mapped into 250 x 250 meter sections according to the City of Helsinki's map sheet division. One good aspect of using this resolution is its compatibility with existing maps and spatial data sets.

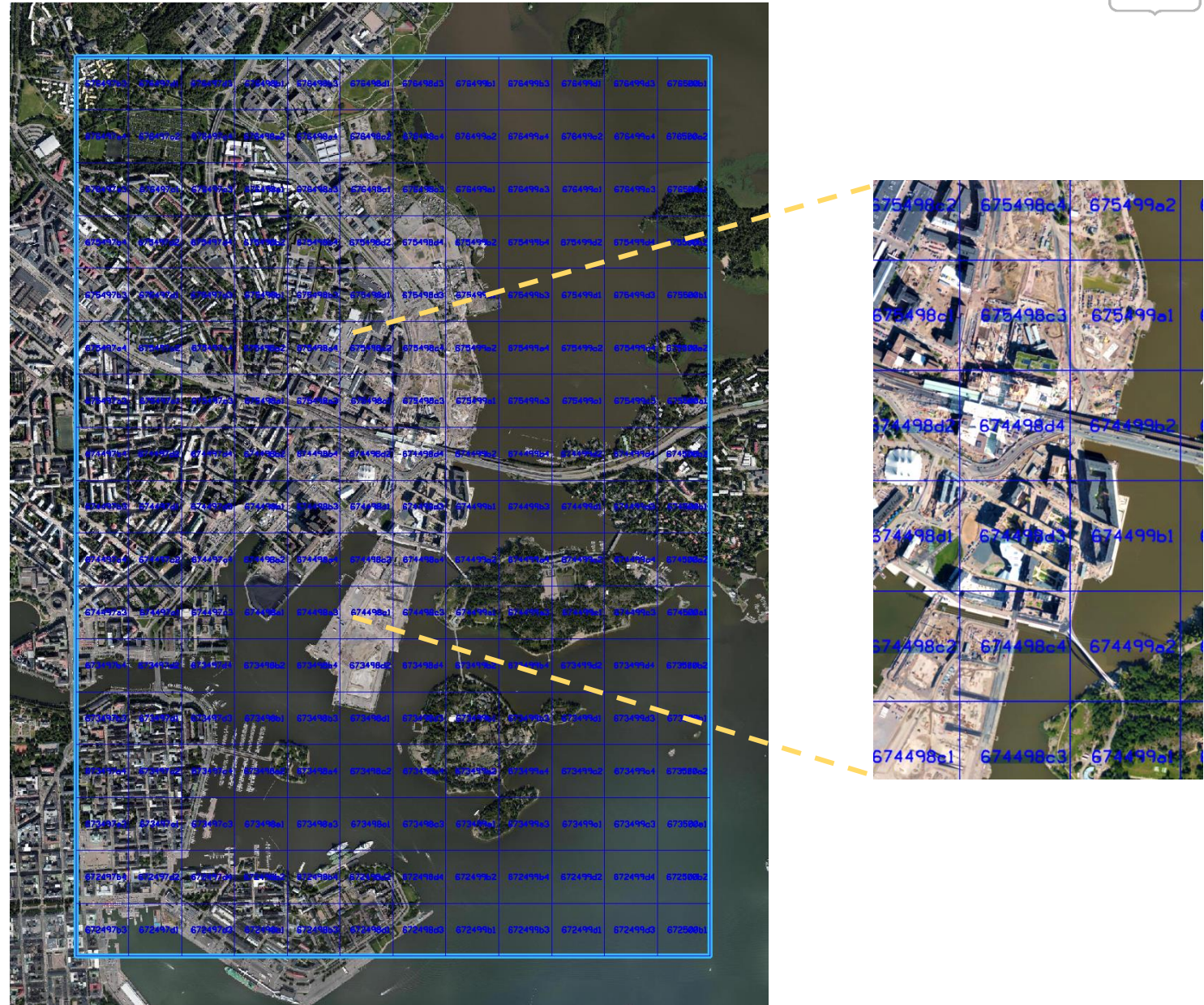


Figure 5. The boundaries of the Kalasatama area and its division into 250 x 250 m sections (close-up on the right). The mesh model was produced inside the blue cropping



## Ground Control Points

Measuring the ground control points is an important step, as the accuracy of the reference points correlates with the accuracy of the 3D model produced. The location of the ground control points must be determined manually so that the mesh model formed by the ground control points covers the entire area as well as possible and completely encloses the modelled area.

The ground control point measurement process was done with the [MicroStation](#) and [TerraScan](#) software programs and proceeded as follows: First, MicroStation searched for a suitable location as the X and Y coordinates were measured using an ortho-image and a base map. The selection of the location of the ground control points utilised well-distinguished and perceptible objects, such as road and other markings, from the orthophoto. Secondly, TerraScan measured the Z coordinates of the ground control points using a classified point cloud. With MicroStation, TerraScan was able to seamlessly look at the desired point cloud classes, such as only ground points or building points.



Figure 6. 15 ground control points and waterbodies in Kalasatama

## Waterbodies

The automated calculation of the mesh model is based on identifying tie points from the images, so 3D modelling of transparent, reflective or moving objects is not possible without specifying separate modelling criteria. For example, the photogrammetric 3D modelling of the surface of large waterbodies generally fails because the surface texture of the water cannot be reliably determined and tie points are impossible to define.

The waterbody to be modelled in the Kalasatama mesh model project area was digitized with a base map and an ortho-image and the final product, a solid vector, was set to zero level. With the water vector data, the geometry of the surface of the mesh model could be modelled correctly. A 3D model can be created according to vector data if no tie points can be found in the images.

## Production of the Mesh Model

Kalasatama's mesh model was produced with the City of Helsinki's own resources and a computer with good computing power. The production of the Kalasatama mesh model included all the steps to produce the actual 3D model and were performed using the [ContextCapture](#) software application. The production can be divided into three phases: processing the input data, automatic aerial triangulation and model calculation.

### Processing the Source Data

The actual production of the mesh model began with the creation of the Kalasatama project in ContextCapture and bringing in more than 2,000 photos with position and orientation information. Next, previously defined ground control points were imported into the project, which were measured using a sufficient number of aerial photographs. After that, all the necessary input data was ready for aerial triangulation.

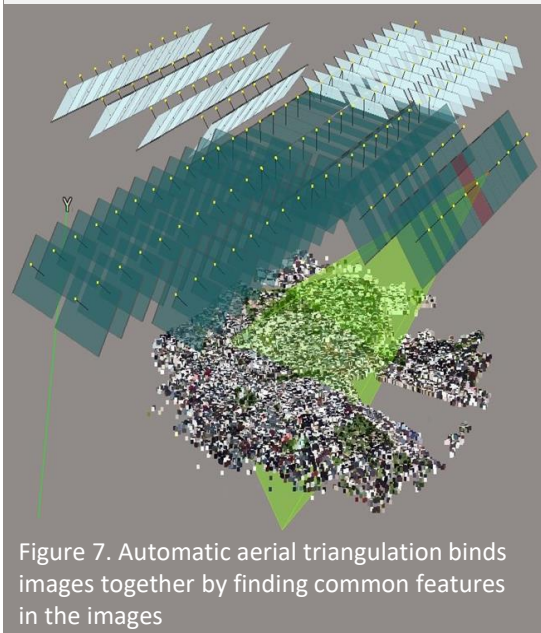


Figure 7. Automatic aerial triangulation binds images together by finding common features in the images

### Automatic Aerial Triangulation

The aerial triangulation was performed using all 2,083 aerial photos. The aerial triangulation attempts to connect all the images together by finding common features in the images. These common features (tie points) constitute a point cloud of millions of points as a result of the triangulation. In ContextCapture, aerial triangulation is not dependent on the user of the application, but different stages of automatic aerial triangulation are solved as closed events in the ContextCapture Engine module.

Automatic aerial triangulation, i.e. bundle block adjustment, is one of the most critical steps in photogrammetric modelling. The purpose of aerial triangulation is to solve the exterior orientation of images using automatic methods and multiple algorithms and to transform the images into the chosen coordinate system. Aerial triangulation includes a production chain involving: image block formation, the selection of tie points, the determination of input values for the tie points, tie point matching, bundle block adjustment and dense image matching.

In aerial triangulation, it is important that the entire image block is processed simultaneously to provide a homogenous quality and uniform pattern. The position and orientation information of the aerial photographs may not be accurate enough to solve the absolute orientation of the image block, which is why ground control points are used to connect the image block to the target coordinate system. Tie points found in the images using automatic methods act as starting points for the triangulation along with the ground control points.

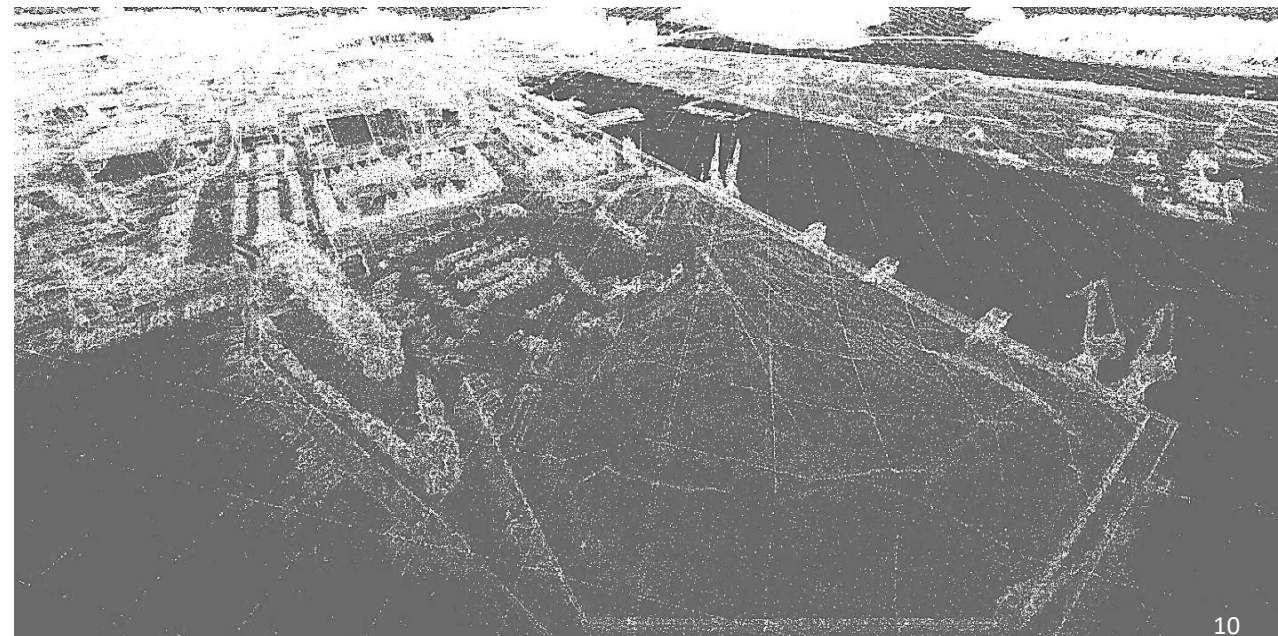


Figure 8. A point cloud formed from the tie points found in the aerial images



Figure 9. Mesh model without visible triangle faces



Figure 10. The final product of the model calculation is a photorealistic mesh model. The shadows captured in the aerial images are one part of the final mesh model

## Model Calculation

Producing the final mesh model product requires a lot of computing power. For this reason, the mesh model should be calculated in parts to lighten the process. The area to be modelled was divided into smaller areas by tiling, which can be modelled one tile at a time. In the calculation of the Kalasatama model, also the vector data of the water area was used to define the water area to match the modelled reality.

The calculation of the mesh model was performed using the ContextCapture software using automated methods in several processing steps. However, for the model's accuracy and quality, the operator was still able to influence the model's computation phase by selecting the desired settings for calculation. Once the mesh model is completed, a quality check can be performed by visual inspection, and by measuring the coordinates of the model, and by comparing the model to the laser scanning data.

**Observations:** The modelling was quite straightforward because the user of the modelling application was only responsible for defining various settings, various preparatory actions, and quality assessment. The success of modelling depends largely on the quality, accuracy and quantity of the source data.

The photorealistic mesh model from Kalasatama was a particularly accurate replica of the real world, because all the stationary objects at the time of photography can be found in the model. The optimum viewing distance of the Kalasatama model is about 100 meters, based on the accuracy of the aerial images used in the modelling. In addition, the model contains almost all the objects on the ground such as trees, shrubs or structures, which makes the model well suited for a variety of analyses and simulations using information on the appearance, geometry, dimensions, or volumes of objects.

The effective use of a model can be based on sharing of other 2D or 3D data, resulting in a well-functioning hybrid model in the best possible case. Planned construction projects have been added to the Kalasatama mesh model, which is illustrated in the following video: [https://youtu.be/c4SoR2SME\\_4](https://youtu.be/c4SoR2SME_4).



## CityGML City Information Model

CityGML is an open, international standard set by the OGC (Open Geospatial Consortium) for 3D city information modelling, storage and transfer. For details see: <https://www.opengeospatial.org/standards/citygml>.

A semantic 3D city information model provides an extra dimension to traditional two-dimensional map, register, and spatial data. Information modelling is also a step towards intelligent digital twins that are not promoted solely through 'traditional' geometry models.

In the CityGML city information model, an object is associated with geometry, topology, semantics and appearance. Object geometry may consist of points, lines, and polygons, and a combination of these. The topology, information content and appearance are semantic features that describe an object as a type of separate geometry.

However, these can also have geometric properties and usually semantics are linked together with geometry. Geometry and semantic information must be compatible when describing the same object. For example, if the wall of a building contains two windows on the semantics side, the same feature must apply to the geometry data.

The OGC adopted version 2.0 of the standard in March 2012. The standard CityGML information model approved in early 2019 is still version 2.0. Alongside this, development of version 3.0 is underway. The new updated version 3.0 of CityGML will expand the capabilities of city information modelling with a number of improvements, extensions, and new features.

More information about the CityGML 3.0 standard is available: <https://github.com/opengeospatial/CityGML-3.0>, <https://www.gis.bgu.tum.de/en/projects/citygml-30/>  
 In the future the CityGML 3.0 standard will be available here: <http://www.citygml.org/>.

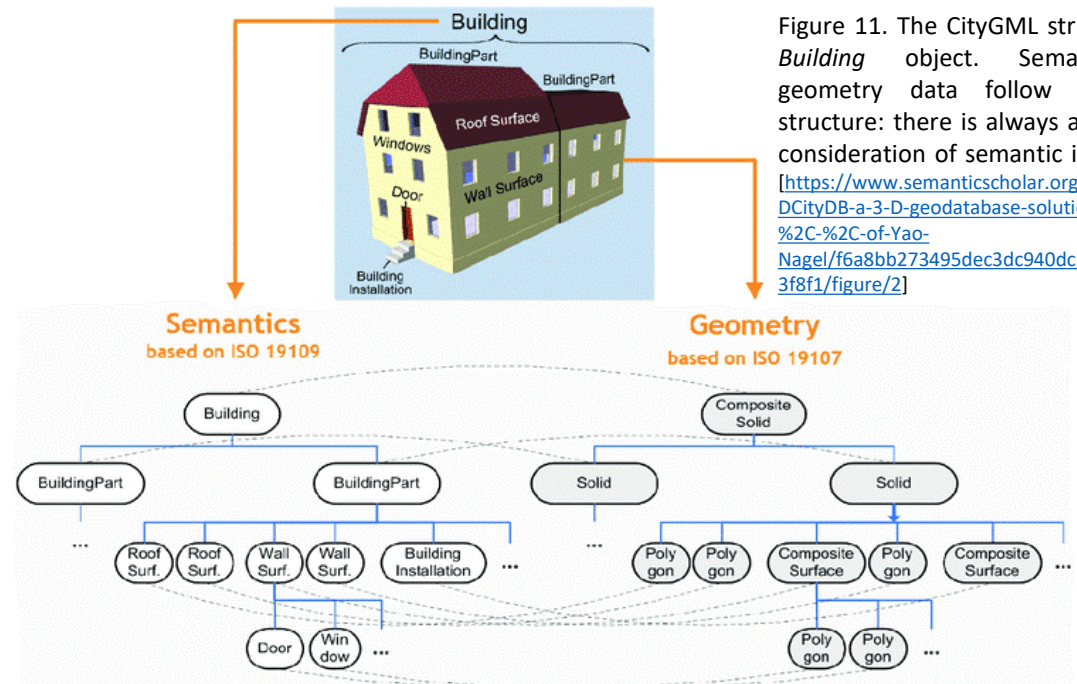


Figure 11. The CityGML structure of a *Building* object. Semantic and geometry data follow the same structure: there is always a geometric consideration of semantic information [<https://www.semanticscholar.org/paper/3-DCityDB-a-3-D-geodatabase-solution-for-the-%2C-%2C-of-Yao-Nagel/f6a8bb273495dec3dc940dc1ec700fe563e3f8f1/figure/2>]

## CityGML Thematic Modules and Classes

The CityGML standard contains definitions of the most important objects and their categories for the city information model. The CityGML information model consists of two types of modules: the core module and the extension modules. The core module contains the basic concepts and components of the CityGML information model. Extension modules allow you to add new thematic classes to your city information model.

The most important objects of the city information model are divided into thematic modules according to their characteristics. These CityGML modelled themes include terrain models, buildings, bridges, water and traffic areas, vegetation and street furniture, etc. Appearance can be added to three-dimensional city objects, and this can be any form of visualization. The appearance can visualize items according to the shape, texture, materials or themes. The Application Domain Extension (ADE) of the CityGML information model is an extension feature that allows new items or properties to be added to the information model. The ADE can be also used to add energy information to buildings, for example.

In the first phase of the Helsinki city information model, all buildings in the Helsinki area were modelled in CityGML. In the following phases, CityGML modules can be added to the city information model using other city objects, such as bridges, trees, waterbodies, terrain models, and different land use areas. In addition, the use of ADE extension features will be possible in the future.

**In the Kalasatama city information model the existing buildings, buildings under construction and planned as well as bridges have been exported to the CityGML database. In addition, the ground (terrain model) and water areas were modelled for CityGML.**

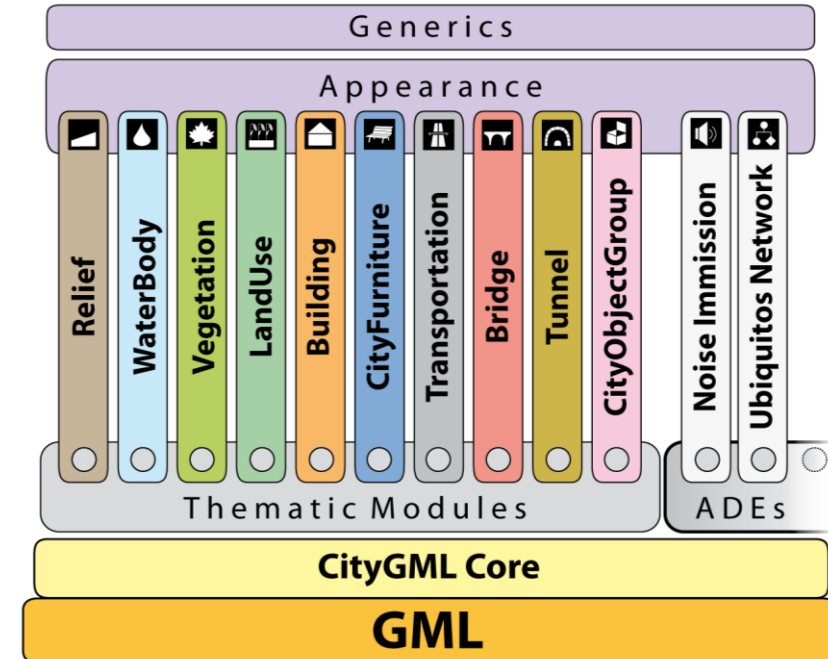


Figure 12. The CityGML modules

[<https://www.virtualcitysystems.de/en/citygml-training/3dcitydb-workshop>]

## CityGML v2.0 Levels of Detail

CityGML allows the geometric, topological, semantic and visual properties of objects to be described on five levels of detail (LoDs). At LoD0, objects are described as two-dimensional, but they can be represented at the correct altitude. At LoD1, the buildings are described as rectangular and with flat roof. LoD2 buildings distinguish different roof forms and surfaces. LoD3-level buildings include architectural features of exterior ceilings and walls, such as windows and exterior doors. At the most advanced level of LoD4, the interior of the buildings and very small details are modelled. Corresponding levels of detail have also been defined for other thematic objects of the city information model, but the definitions are not as complete as they are for buildings.

**LoD0** is the roughest level of performance. Buildings are presented as horizontal polygons so that the buildings' footsteps or ceilings can be identified. Thus, two-dimensional models are easy to convert into LoD0-level models by providing them with elevation levels, for example, using a terrain model. In the terrain model an aerial view or a base map can serve as a texture.

**LoD1** presents objects as box models, i.e., buildings with flat roofs and bridges in general. The model can be produced from a base polygon. For buildings, the height is usually determined by average of the eaves and the roof ridge.

At **LoD2**, the roof contours and other external features of the building can be more accurately represented. The surfaces can also be given a surface texture. Other objects and themes, such as vegetation, can also be generally introduced into LoD2-level city models.

**LoD3** is a more detailed level than the previous ones and the building is presented with doors and windows and smaller overhangs and niches. In addition, precise texture can be given to all surfaces. LoD3 also includes descriptions of traffic-related structures and vegetation.

At **LoD4**, the buildings are also modelled with indoor characteristics. The Room category separates the interior of the building according to its uses, and the class can contain information about class, function and usage.

The level of detail should always be chosen according to the need and purpose. City-wide 3D modelling is worthwhile to perform at either the LoD1 or LoD2, as it is best to utilize semi-automated 3D modelling and existing maps and GIS data. Up to 70 % of the buildings can be successfully modelled automatically. Creating an LoD3 or LoD4 city-wide model is not reasonable at the moment, as more detailed modelling requires more detailed information on all of the objects. The LoD2 model is usually sufficient for the most common analyses and simulations with the city information model.

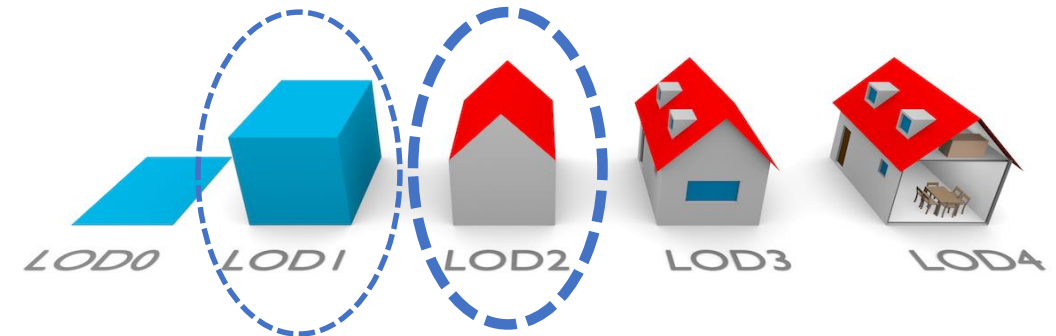


Figure 13. CityGML levels of detail [<http://filip.biljecki.com/phd.html>]

When the CityGML standard is updated to version 3.0, the concept of levels of detail and their use described above will also be updated. The publication by [Kutzner and Kolben \(2018\)](#) briefly describes how, among other things, the detail level concept may be updated in the development of the standard. The publication mentions that LoD4 used to display interiors would be removed and only LoD 0–3 would be retained. Instead, the presentation of the interior aspects would be directly integrated into LoD 0–3. For more information on CityGML v3.0 levels of detail, see [Löwner et al. \(2016\)](#).

## Production of the CityGML v2.0 Model

The city information model of Kalasatama consists of buildings in their present state, separately modelled buildings under construction or planned, in addition to bridges (present and planned), as well as ground and waterways, according to the CityGML standard.

### Source Data








The current representation of Kalasatama's buildings (LOD 1 and 2) are based on the CityGML information model of the entire city. It was created in 2017 and is now under constant maintenance. The source data for modelling are the buildings' footprints on the base map, laser point clouds (via Airborne Laser Scanning, ALS), and unique characteristics (such as attribute information) are added from the municipal register.

**Future buildings** (under construction and still in the planning phase) were modelled using several data sources including: plans, perceptual images, existing CAD models, etc. In practice, the buildings were completely re-modelled, even in cases where the 3D model was ready, because they did not meet the requirements for geometry or cleanliness.

The source data for **the bridges** consisted of a bridge model data set called "bridge cover" maintained by the City of Helsinki, ALS point clouds, and the national cadastral structure register with characteristics.

**The waterbody** (sea area) was modelled using the water areas from the base map and design models. At this point, the theme was not included in the database, but the data is CityGML compatible.

**The terrain** was created with a city-wide landscape model of the present state of the location, as well as with a triangular planned surface (this is CityGML compatible, but not in a database).

Buildings, current stage (CityGML)	
Buildings, under construction/planning process (CityGML)	
Bridges, current and planned (CityGML)	
Waterbody (CityGML compatible)	
Terrain, planned (CityGML compatible)	
Terrain, current (CityGML compatible)	
Model's area of coverage	

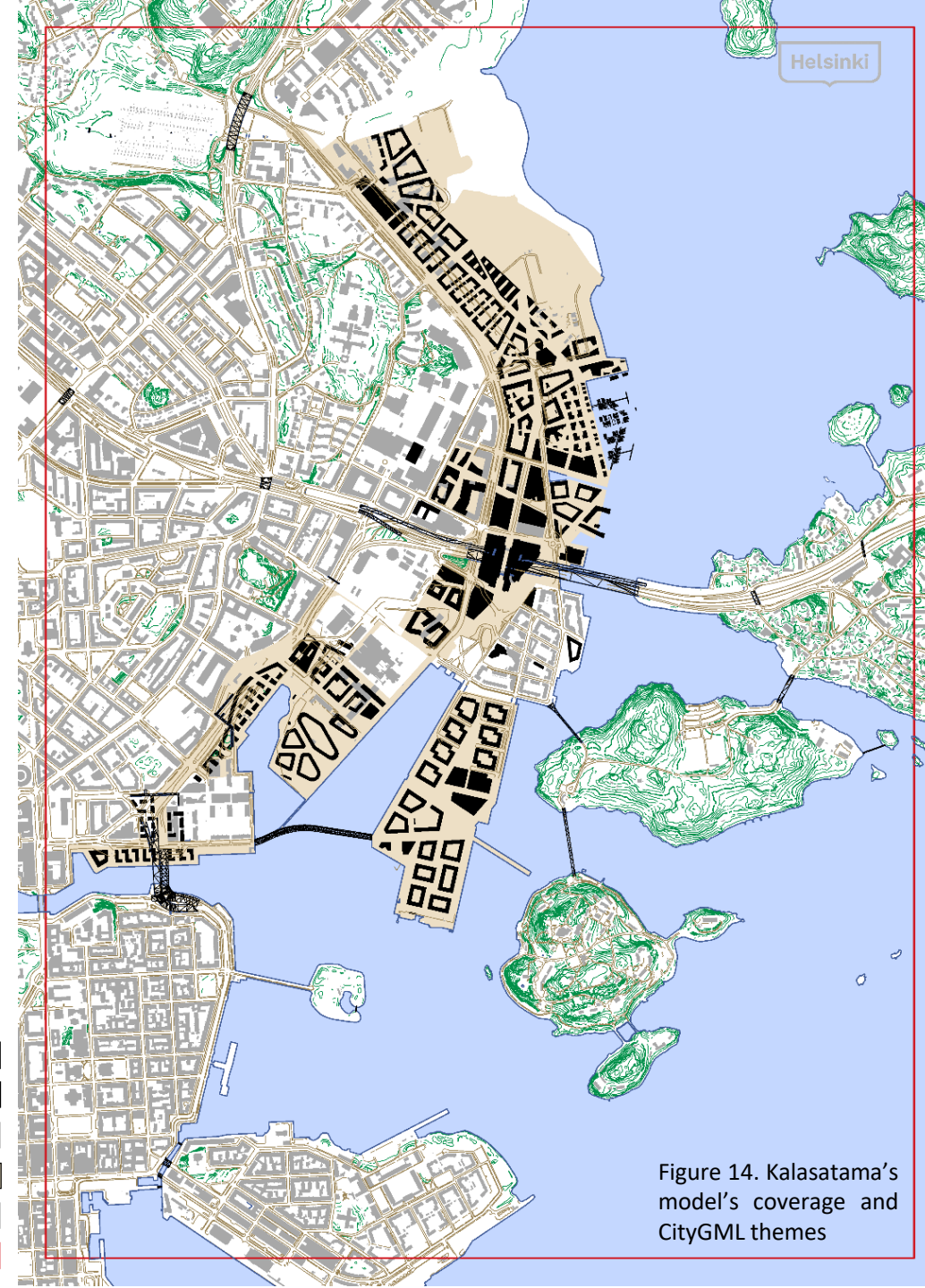


Figure 14. Kalasatama's model's coverage and CityGML themes

## CityGML Buildings

Buildings are one of the most well-defined themes in the CityGML standard. The building theme includes detailed thematic and spatial concepts of buildings and parts of buildings at five levels of detail (LoDs 0-4). A building can be represented simultaneously on different LoD levels by providing separate geometries for the corresponding LoD levels. Importantly, the geometry must always match the semantics.

The main category of the building theme is the *AbstractBuilding* theme, which is a subclass of the *CityObject* class. A *CityObject* is a basic class for all thematic classes. An *AbstractBuilding* represents either a building or part of a building (*Building* or *BuildingPart*). Building complexes, such as colleges or hospitals, consisting of a number of separate buildings, can be labelled a *CityObjectGroup* in the CityGML model, from which it is possible to separate the main building in the complex.

The facade of the building can be separated semantically into the *BoundarySurface* and *BuildingInstallation*. The *BoundarySurface* is a special exterior part with a function such as a *WallSurface* or *RoofSurface*, while a *BuildingInstallation* could be a balcony or stairs.

The *Building* and *BuildingPart* can inherit the following features from the *AbstractBuilding* class: *class*, *function*, *usage*, *yearOfConstruction*, *yearOfDemolition*, *roofType* and the *measuredHeight*. Additionally, an *Address* can be linked to the *Building* and *BuildingParts*. In addition to the features included in the *AbstractBuilding* category, new items and attributes can be added to the template either with the ADE feature or by using the *genericsAttribute* of the *Generics* module.

[OGC City Geography Mark-up Language (CityGML) Encoding Standard v2.0, <https://www.opengeospatial.org/standards/citygml>].

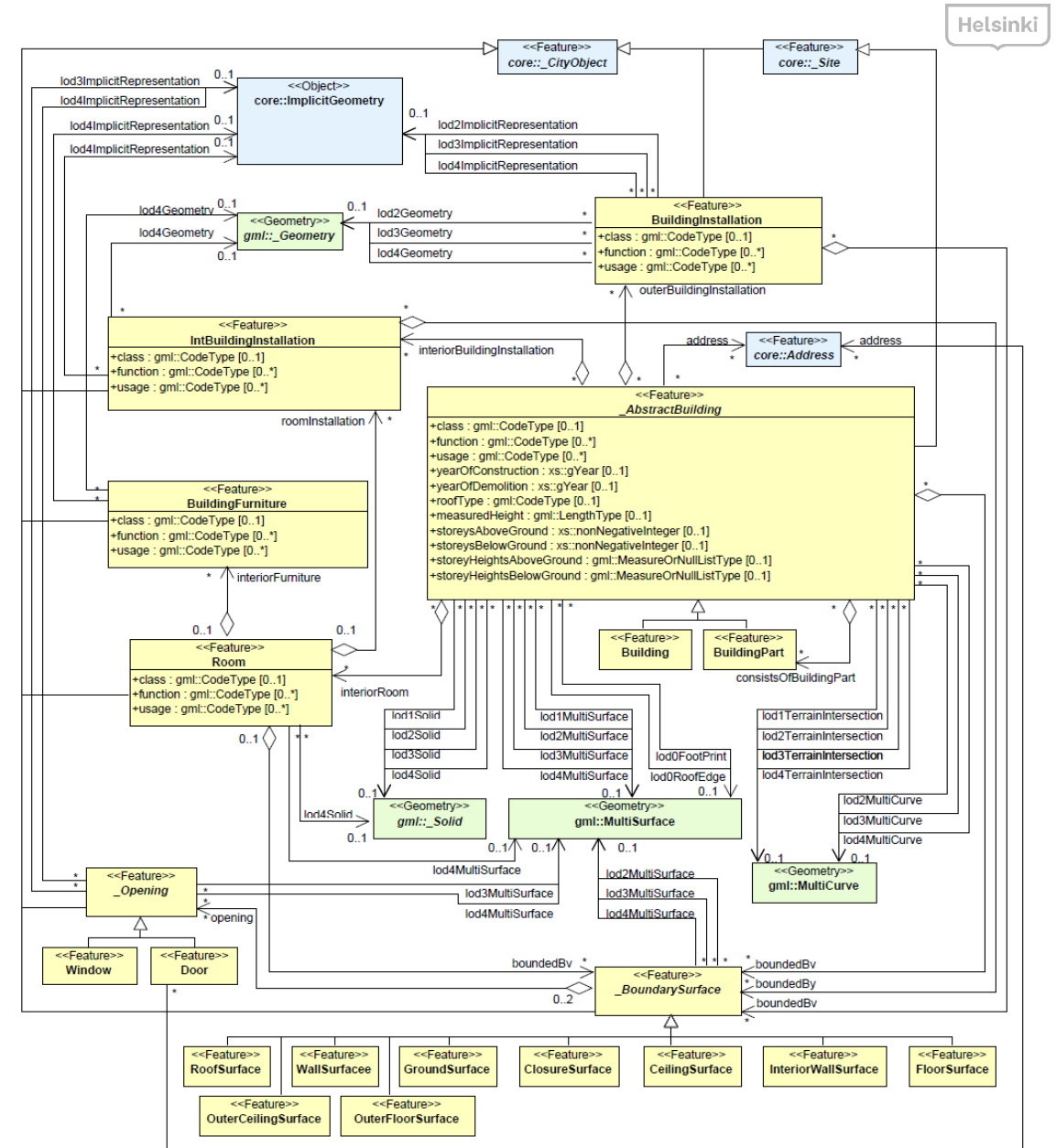


Figure 15. UML diagram of CityGML building [http://www.citygml.org]

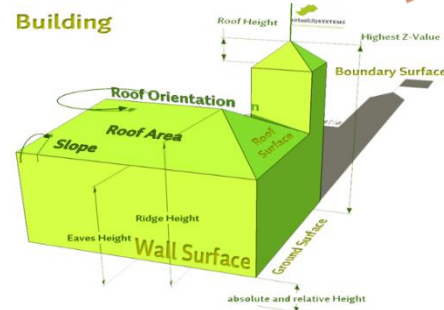
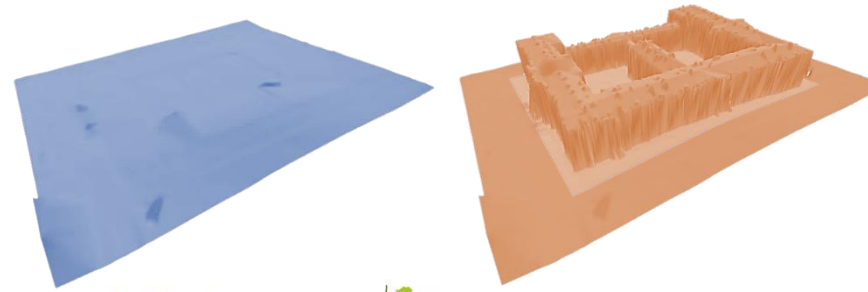
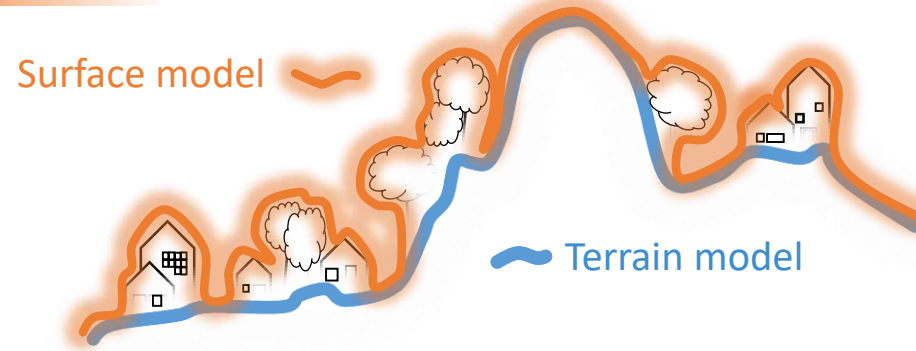


## Modelling the Buildings

The starting point for the building modelling is the building's footprint, point cloud (Digital Terrain Model, DTM) and surface model (Digital Surface Model, DSM). Point clouds (current status) are based on laser scanning (ALS). For more details on page 28.

The terrain model is used to determine the elevation position of the building's footprint (although the base map and design models may not contain direct altitude information). The surface model is used to identify the height and roof shape of the building.

The building footprints of the city information model's buildings are identical to the base map and 2D plans for the building footprints.



- CityGML semantic  
- SHP & CityGML attributes

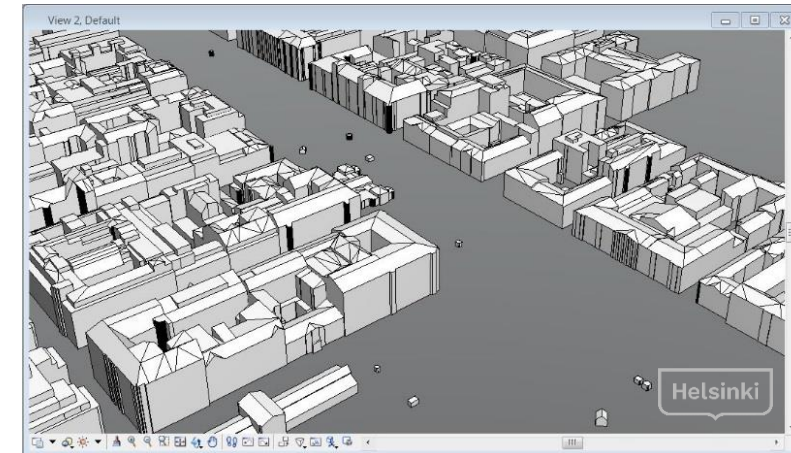
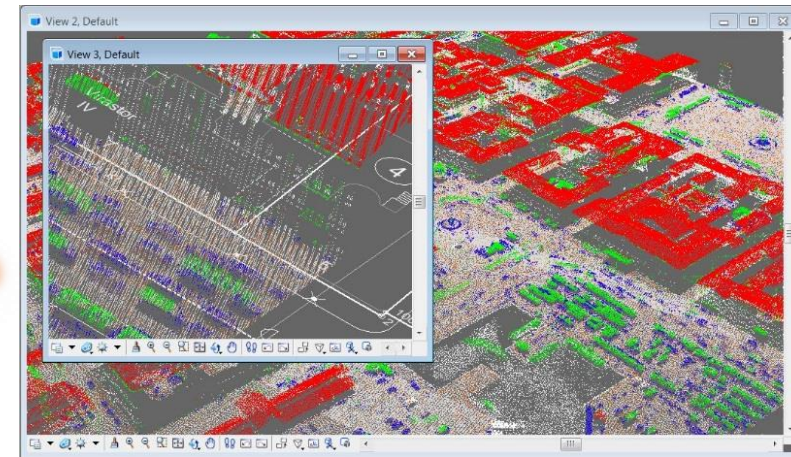
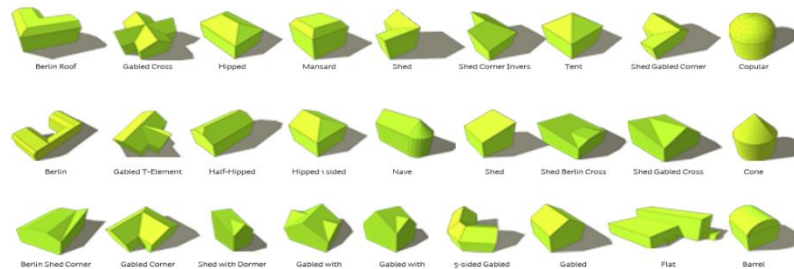


Figure 16. →↓ (captions read from left to right, top to bottom)

- Description of surface and terrain models.
- Base map and point clouds.
- Terrain and surface models from one building area.
- Roof shapes of buildings from point cloud.
- Semantics of the buildings, and roof shape library.
- Geometries of city information modelled buildings.

[Image: [https://www.virtualcitysystems.de/images/vcs/products/buildingreconstruction/1024px/BREC\\_Rooftypes\\_Attributes\\_EN.png](https://www.virtualcitysystems.de/images/vcs/products/buildingreconstruction/1024px/BREC_Rooftypes_Attributes_EN.png)]

## Maintenance of CityGML Buildings at Helsinki City Survey Services

Maintenance and updating the CityGML buildings in the 3D city information model of Helsinki is carried out at Helsinki City Survey Services in the City Urban Environment Division. Buildings are modelled as CityGML models conjointly changing the input vectors of the base map buildings. The process is described below.

The dimensions of the footprint, and the locations of new buildings are taken to the base map. The base map is also edited when a building is demolished, for example. Modelling CityGML buildings begins with reading the base map. The base map searches for all the building vectors in the buildings that have been modified. Building footprints are automatically located, and [BuildingReconstruction \(BRec\)](#) application data is automatically generated from these footprint vectors. The source data is opened directly with the BRec application, which the operator uses to manually convert the building into CityGML. All possible building information is used as the source data. In the case of a new building, the shape of the building is evident from various annexes to the building permit. The building is modelled as accurately as possible on the basis of the existing data.

As shown in Figure 17, the new buildings imported into the city information model are initially without a photo texture. At present, all textures in the city information model's buildings come from aerial photography, but other photographic means could also be utilized. Texturing is done for those buildings that were at least under construction at the time of shooting.

**Observations:** Since the BRec application produces only LoD1 and LoD2-level buildings, the resulting CityGML building can be very different from the building permits. Although the base map maintenance, survey and drawing instructions are currently not directly compatible with 3D modelling, it is appropriate that the maintenance work is linked to the update of the base map,

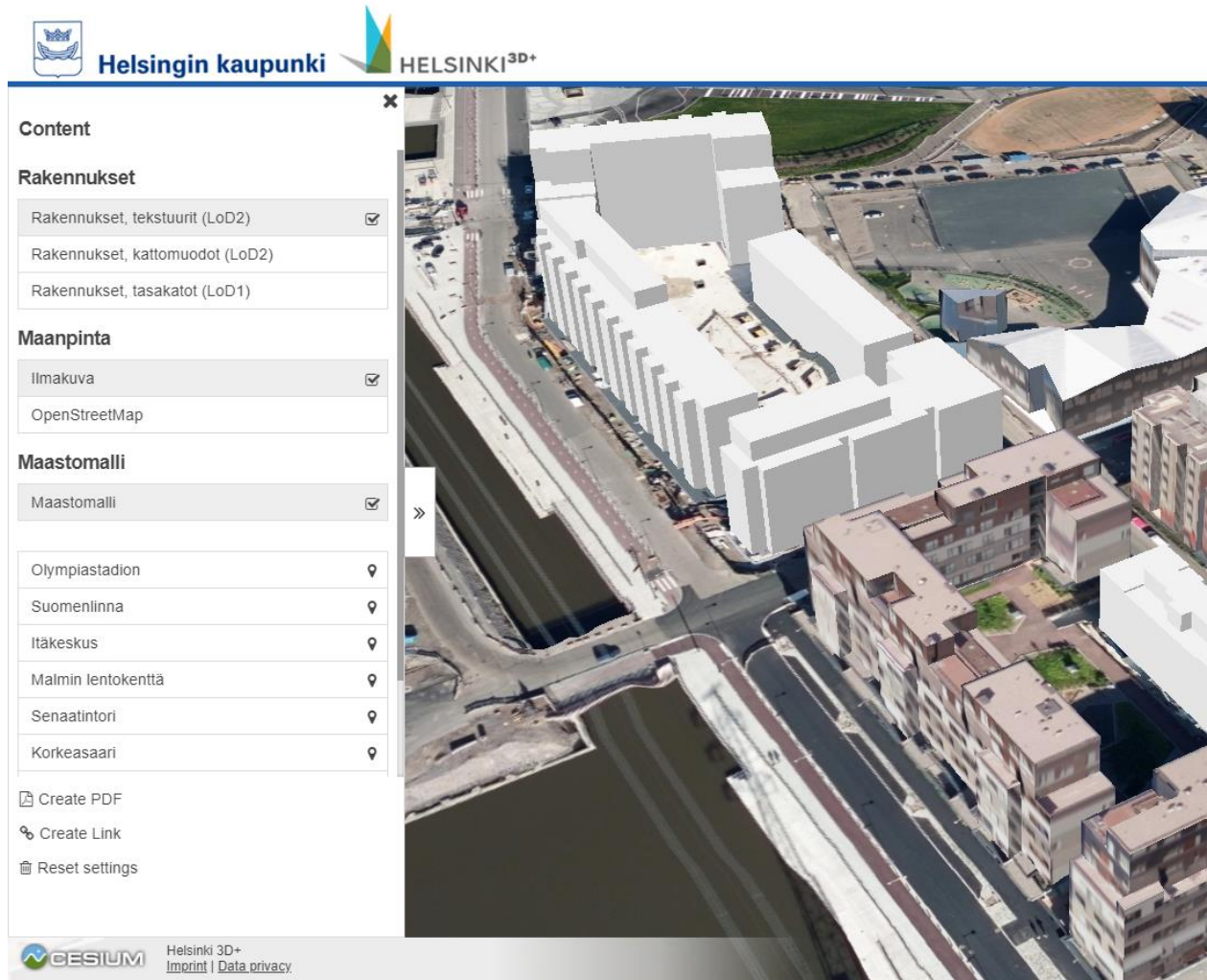


Figure 17. Browser view of [kartta.hel.fi/3d](http://kartta.hel.fi/3d) web site, where 3D city information model is being updated

and that the modelled CityGML buildings and the footprint vectors of the base map buildings are consistent. Maintenance of the city information model is crucial, and it is important that new, modified or demolished buildings can be immediately updated in the city model at the same time as updating the base map.

## Modelling the Future Buildings and Environment

While the construction of the Kalasatama area is only in its early stages, most of the buildings to be modelled in the area are only at the planning stage. Modelling unconstructed areas before building them is one of basic reasons for using digital twins.

The source data for future buildings is received from the Kalasatama–Malmi project. The data is divided into several files and is available in DGN format. In addition, the data set contains illustrations, conceptional sketches and scenes of the entire Kalasatama area. The requirement for modelling was to create a unified surface model, in which the buildings are closed objects. The model must not contain overlapping parallel surfaces and a separate surface model should be created. The source data included a few differences between design patterns and illustration images, but when the illustrations were more actively maintained, it was decided to rely on the illustration images if differences were detected. The process description for modelling is clarified below.

### Planned Models

Modelling the buildings started by copying the buildings in the design templates to their own layers and then modifying the solid volume models. All linear and surface elements were removed. The planned buildings were created with the MicroStation application in DGN format. The buildings were generally been created at LoD1, which means they are presented with flat roofs and without windows.

Conceptual sketches and the mesh model produced of the area were the source data for the terrain model. In the design models, the ground was extruded to three meters throughout, but in the area there are and will be altitude variations, so it was decided to look at the mesh model and the conceptual sketches. By comparing the street edges of the conceptual sketches to the mesh model, the ground objects were moved to the correct height. Terrain modelling was carried out with the [TerraModeler](#) software application and the ground was classified into street, planting, sand, bike, and market areas. These themes were given material and colour specifications in MicroStation. These classes could possibly be used in the production of the CityGML model.

Future buildings are presented in accordance with the latest plans. Thus, not all the design buildings will necessarily come into being as shown in the model.



Figure 18. In general, planned building have been modelled with flat roofs and without windows. View from the OpenCities Planner browser-based application, where the models of the planned buildings have been embedded

## Modelling Planned Buildings into the CityGML Format

By 2018, about 3,000 Helsinki residents will live in the Kalasatama area. By 2040, the region will have 25,000 inhabitants and 10,000 jobs as planned. The total floor area of the apartments will be 1.2 million square meters and 400,000 square meters of office space. The construction of Kalasatama is therefore only at an early stage. Construction will continue for twenty years. By 2030, Sompasaari, Suvilahti and Verkkosaari will be completed, while construction will continue until 2040, at least in Hanasaari, Kyläsaari and Hermanninranta.

The starting point for the production of the CityGML building models were the Computer Aided Design (CAD) models of the buildings in DGN format, as shown in Figure 19. The designs were created on the basis of source data from the Kalasatama–Malmi project (the process described earlier on page 19). First, the integrity and suitability of building models for different CityGML modelers were investigated.

After that, it was decided to use the BuildingReconstruction (BRec) application, which can be used (together with building footprints, terrain models and surface models) to produce semi-automated semantic building models according to the CityGML standard.

### Creating CityGML buildings

For the preparation of the BRec input data, the 3D models had to be separated from the buildings' footprints and roofs. Editing the 3D models began with the removal of footprint from the DGN-shaped buildings as shown in Figure 20. The dimensions of the footprints and the separate buildings, for example, in the case of solid building blocks, were determined by the data processor. As a result, approximately 250 separated building were created (Figure 21).

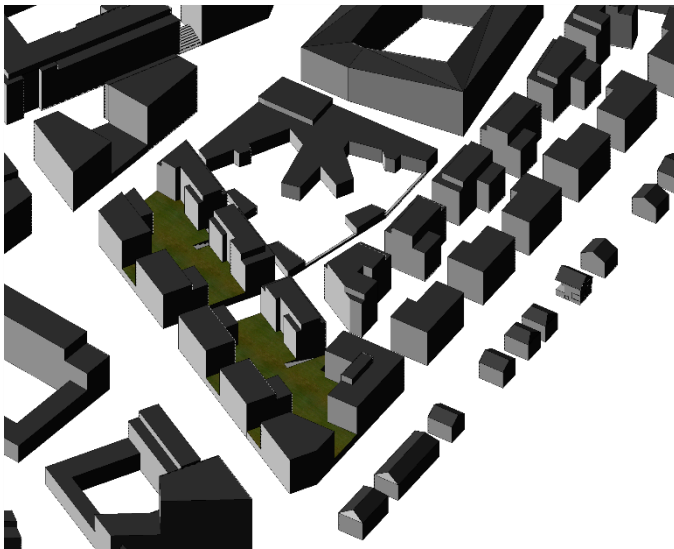


Figure 19. CAD models in DGN format (application: MicroStation)

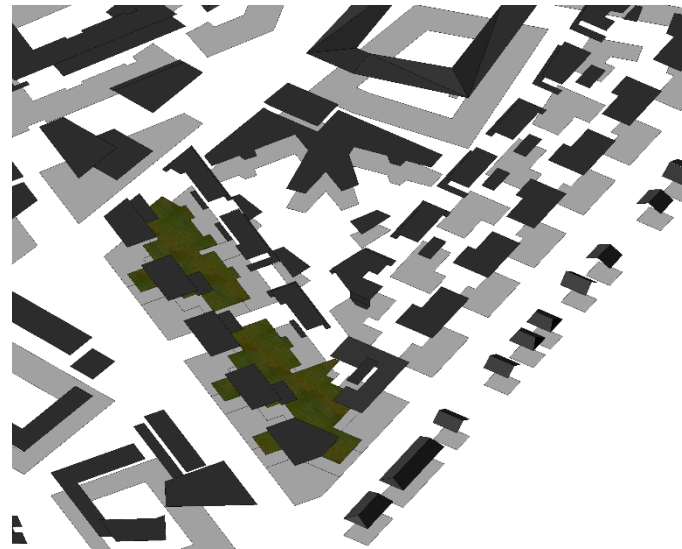


Figure 20. Separation of the footprints and the roofs

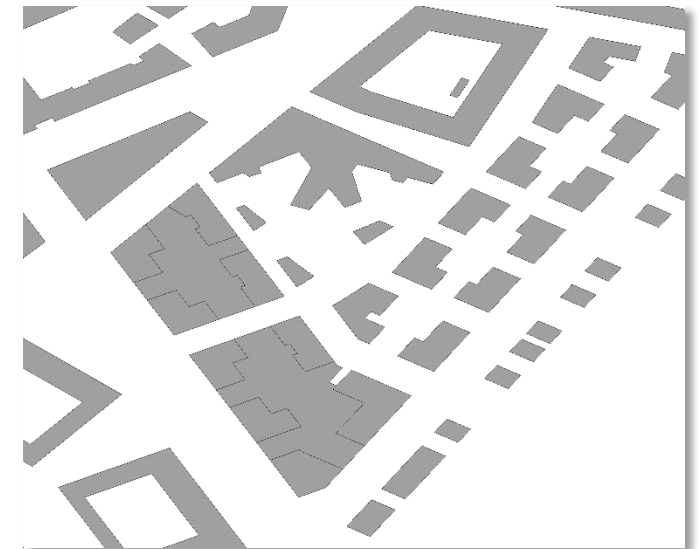


Figure 21. The footprint vectors of the buildings

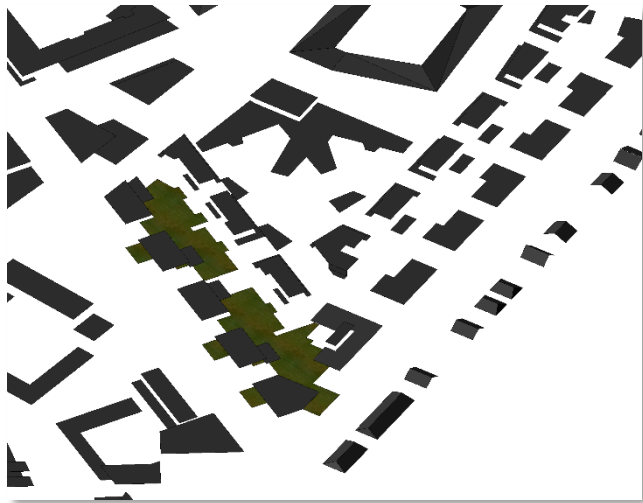


Figure 22. Separation of the roofs (application: MicroStation)

After editing and validating the footprint, the roofs of the buildings were removed from the 3D models (Figure 22). The choice of roof vectors was not as accurate as the determination of the footprint, as long as the roof elements covered the entire building. The detached roofs of the buildings were exported to the [Rhinceros](#) application and converted into a point cloud (Figure 23) using its graphical [Grasshopper](#) algorithm editor. This point cloud was further processed to create a surface model.

Next, a unique identifier was created for each footprint vector using the FME application and the final output data was converted to a format suitable for the BRec application. In addition, the planned buildings were divided into smaller groups so that the modelling could be as smooth as possible with the application. The planned buildings were thus divided into seven areas: Redi, Hanasaari, Sörnäistenniemi, Sompasaari, Suvilahti, South of Verkkosaari and the northern part of Verkkosaari.

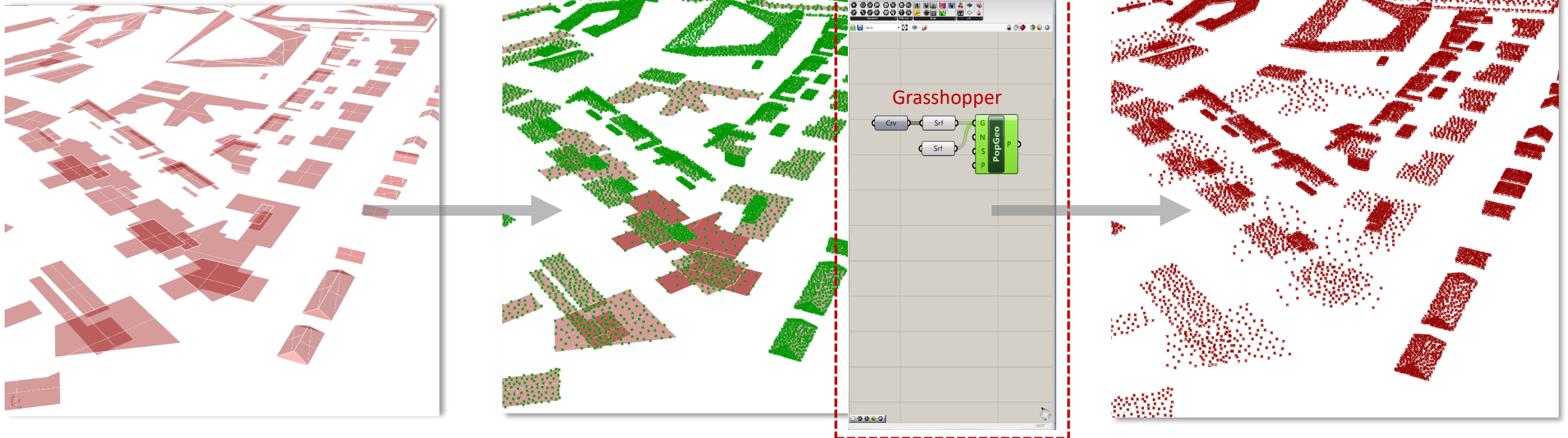


Figure 23. The roofs in the Rhino application (left) and the point cloud (right) produced using the Grasshopper application (middle)

Modelling with the BRec application proceeded as follows: In the first stage of work (Figure 24), the footprints of the buildings to be modelled were imported into the application with ID identifiers. Additionally, the digital terrain model (DTM) and digital surface model (DSM) were imported. In the second stage, the DTM and DSM models were fitted with the buildings so that the footprints were planted on the DTM and the roofs of the building were extruded with the help of DSM. The roof shapes were selected from the roof shape library. This step occurred automatically (Figure 25). The result obtained was examined together with a point cloud to see if the model needed manual processing (Figure 26).

The BRec application semi-automatically produces 3D buildings of up to LoD2 in accordance with the CityGML standard. Fully automatic modelling was successful for medium-sized buildings and buildings with a flat roof. Entire blocks were too large for automatic modelling, as were complex and inclined roof forms. Nevertheless, thanks to the semi-automated modelling, the final modelling of the CityGML buildings with the BRec application only took about 10 hours for almost 250 buildings.

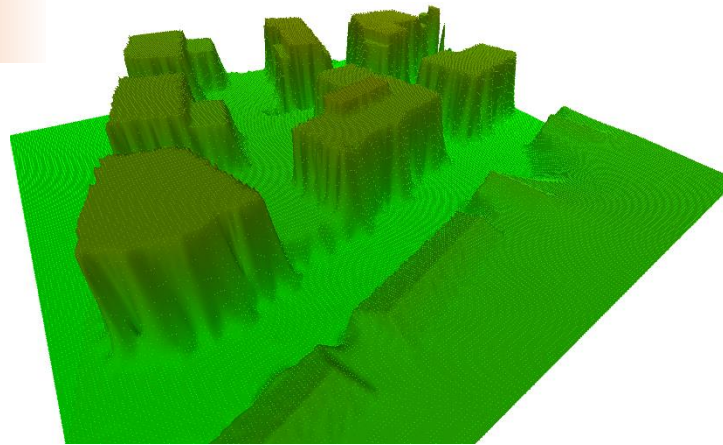


Figure 24. Step 1: DTM and DSM (application: BRec)

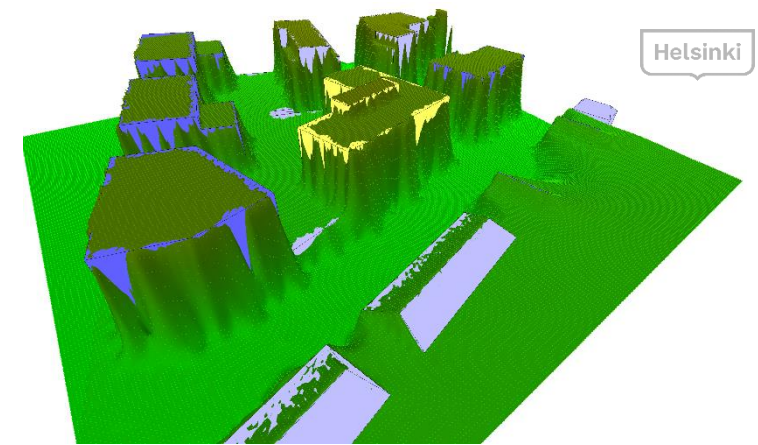


Figure 25. Step 2: Buildings fitted into the DTM and DSM

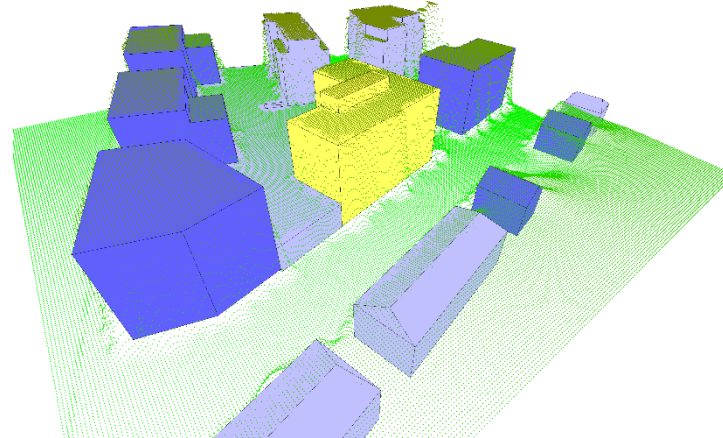


Figure 26. Step 3: Examination utilising the point cloud

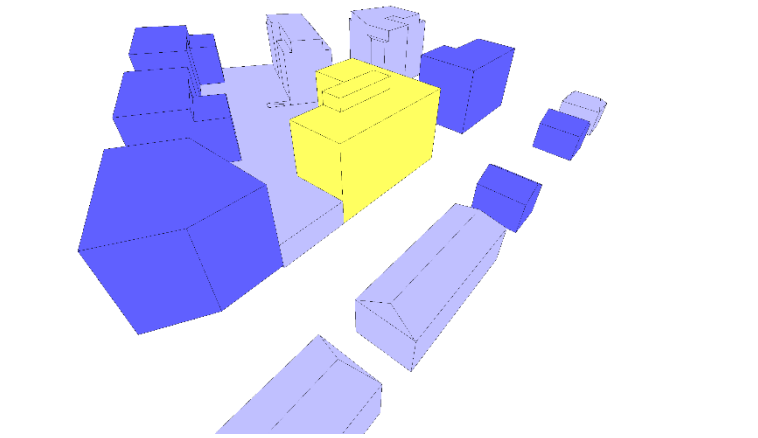


Figure 27. Step 4: The resulting CityGML buildings

**Observations:** The input data for the buildings was challenging for the automatic CityGML conversion. The CityGML conversion of 3D models of the planned buildings was also tested with an FME converter and a SketchUp CityEditor extension. These applications produced incomplete CityGML models because the input data contained modelling errors. The buildings were modelled with BRec application because the source data was naturally corrected and processed to suit this particular modelling method. The semi-automatic conversion from CAD models to CityGML models is an effective way to ensure the functionality of the transformation and enables manual modelling for the most complex buildings. The most important task is to make sure that the output data is watertight and valid for CityGML models.

Some of the courtyards in the new building blocks were at a much higher level than the neighbourhood because the courtyards consist partly of underground garages with a courtyard area. It was decided to model these objects as CityGML buildings for the overall view of the area. The maintenance work of the CityGML buildings at the city survey services office is based entirely on the update of the base map, and these traditional mapping guidelines do not always support 3D city modelling, and the partially underground car storage sites exported to the base map are still not 3D-modeled and therefore they are excluded from the city information model.

## CityGML WaterBodies

The CityGML *WaterBody* is divided into three semantic parts within the *WaterBoundarySurface* class. The *WaterSurface* represents the water surface, the *WaterGroundSurface* represents the water base and the *WaterClosureSurface* represents the water mass, i.e. the space between the surface and the bottom as shown in Figure 29. However, the only mandatory information according to the CityGML v2.0 standard is the *WaterSurface* as it represents the boundary between the air and water. The *WaterBody* is a subclass of the *WaterObject*, which is a derivative of the *CityObject* section. The geometric representation of the *WaterBody* object varies depending on the selected LoD levels. The *WaterBody* object can have attributes such as a *class*, *function*, and *usage*.

Most often, the LoD0 and LoD1 levels used to display the water represent both low-level visualization and coarse generalization. This is why rivers are presented using *MultiCurve* geometry, but streams are completely excluded. Due to their large size, seas and lakes are always presented using *MultiSurface* geometry. However, different types of geometric combinations can be provided for *WaterBody* objects, for example, the water flow can be represented by *MultiCurve* geometry.

[OGC City Geography Mark-up Language (CityGML) Encoding Standard v2.0, <https://www.opengeospatial.org/standards/citygml/>]

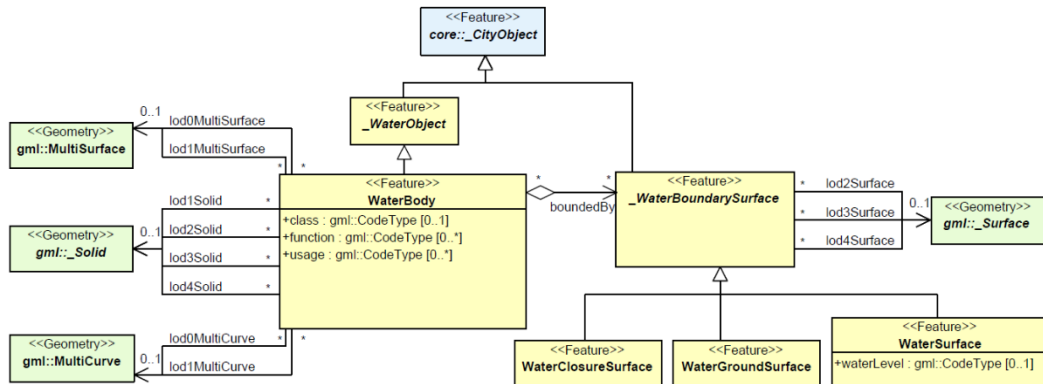


Figure 28. UML diagram of CityGML waterbody [<http://www.citygml.org>]

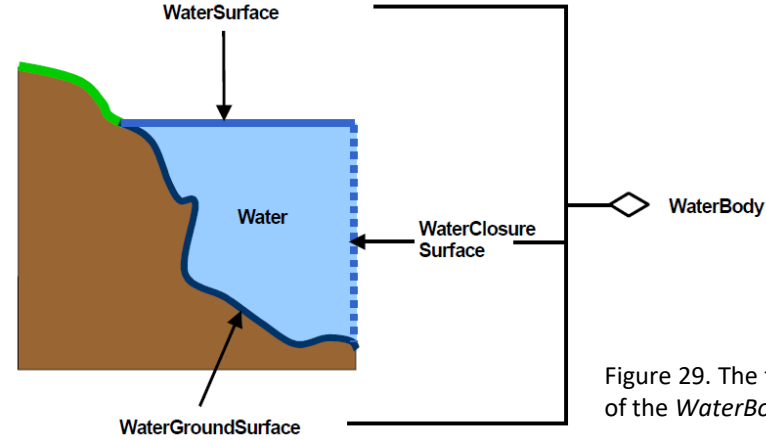


Figure 29. The three semantic parts of the *WaterBoundarySurface* class

## Modelling the CityGML Waterbodies on a Theoretical Level

The sea is an essential part of the Kalasatama area and waters should be a significant part of the CityGML model. However, the actual modelling of the waterbody was not implemented within this project. Nevertheless, the CityGML modelling process for water areas had to be reviewed at the theoretical level. The final modelling may be implemented later.

The source data for the CityGML waterbody is vector-based data from the water areas. The exact dimensions of the water are produced from the terrain model (DTM), as well as the orthophotos and the coastline vectors from the base map. The water areas were digitized once in 2016 for the mesh model of the entire city, so the CityGML modelling could use vector-based data from existing waters. With the 2017 orthophotos, the base map and the terrain model, the vector-based water data were updated and corrected to fit the CityGML modelling. Seabed information is currently unavailable, so the *WaterClosureSurface* should be set for example X meters below the real water level. Finally, the vector-shaped data would need to be read into the FME application, where the data would be converted to CityGML. The definitions of this CityGML waterbody model are as follows: *wtr:WaterBody* and *LoD1MultiSurface*.

## CityGML Bridges

The CityGML bridge model enables the thematic, spatial and visual aspects of bridges and bridge parts at four levels of detail, LoD 1–4. A bridge is presented by a *Bridge* class object, which inherits its attributes and relationships from the abstract base class *AbstractBridge*. The modelling of bridges is included in the *CityObject* section. The bridge model is based on the building model with regard to its structure and attributes. Like the building models, the semantic and geometric accuracy increases with LoD levels. LoD1 represents a box model, and LoD3 already represents a fully architectural model. The LoD4 level also describes the inner structures of the bridge.

Bridges can be represented in *MultiSurface* geometry if, for example, the first layer of the bridge is not uniform. Additional visual features of bridges, such as wires or ropes, can be represented as *MultiCurve* elements, but semantic objects such as antennas or other functional bridge parts can be represented as *BridgeInstallation* or *BridgeConstructionElement* objects. A *BridgeConstructionElement* describes a part of the bridge that is essential for the structure of the bridge (such as a support element), but a *BridgeInstallation* element may be a feature that does not affect the operation or stay of the bridge. Parts of the bridge that are separate from the other bridge, such as ramps, can be presented as *BridgePart* objects. One bridge may consist of several *BridgePart* objects. A *BridgePart* object is also a subclass of the *AbstractBridge* class whose semantic attributes are *class*, *function*, *usage* and *is\_movable*. The *class* describes the type of bridge and the means of construction, *function* describes the use of the bridge, *usage* describes a non-standard operation of the bridge and *is\_movable* states whether it is a mobile or a fixed bridge or part of a bridge. Attributes are shown as *gml:CodeType* and the bridge name can be represented in *gml:name*.

[OGC City Geography Mark-up Language (CityGML) Encoding Standard v2.0, <https://www.opengeospatial.org/standards/citygml>].

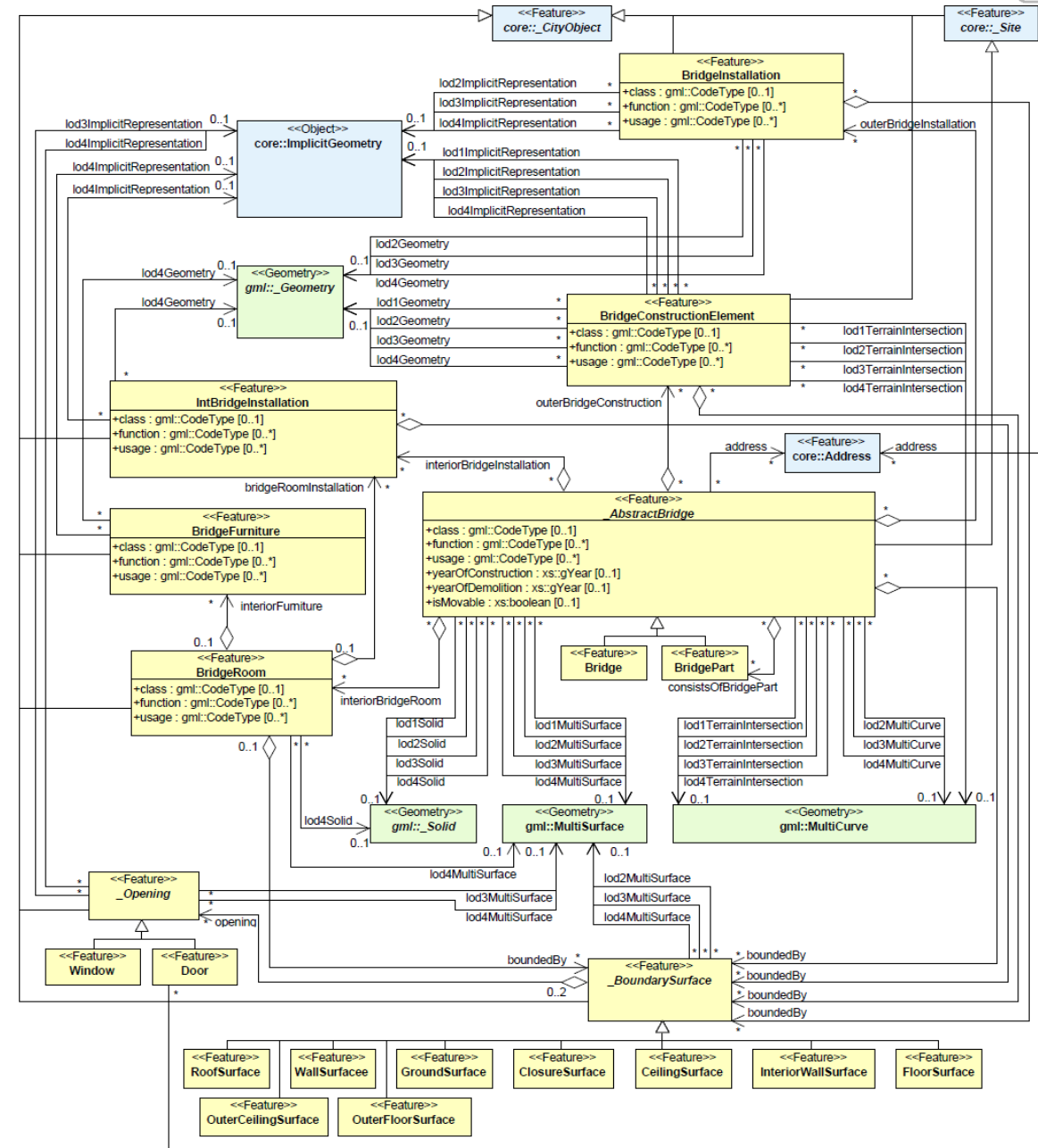


Figure 30. UML diagram of a CityGML bridge [<http://www.citygml.org>]



## Modelling Bridges into the CityGML Format

There are about 30 bridges in Kalasatama and the surrounding area. The City of Helsinki has two separate data sets for bridges: vector-based data in DGN format and registration information. All items that were in the border area were retrieved from the bridge register and close to 50 of these items were found. The modelling of the bridges is described below.

In the DGN vector data of the bridges, only the bridge cover is modelled as shown in Figure 31. For the 3D city information model, it was decided to give the cover structure of the bridges a standard thickness of two meters, resulting in a box model, i.e., a *LoD1MultiSurface* object. The two-meter thickness was used to make the bridges more visible in the city information model, but still remaining lightweight.

This stage was performed manually with the MicroStation application, but so that all the selected items could be processed simultaneously. First, the bridge cover was converted into a single object instead of separate triangles to separate the outline of the cover. The cover piece was then copied two meters below the cover, and finally the outline of the cover was extruded to the edge between the cover and the bottom, resulting in the sides of the bridge (Figure 32). It is important to check the appropriate direction of the normals of the bridge deck, base and sides (Figure 33). The normals should point away from the object so that the model appears correctly in the city information model. At this point, the cover, the base and the sides of the bridge are uniform objects, so they need to be converted back to individual triangles for CityGML conversion.

The bridges are also included in the bridge register, where you can find the characteristics of the bridges, such as the name, function, and dimensions of the bridge as well as the year of construction. These attributes to be added to or linked to the objects are an integral part of CityGML city model objects.



Figure 31. The starting point: a vectorized bridge in DGN format (Kaikukatu's light traffic bridge, in Hakaniemi)

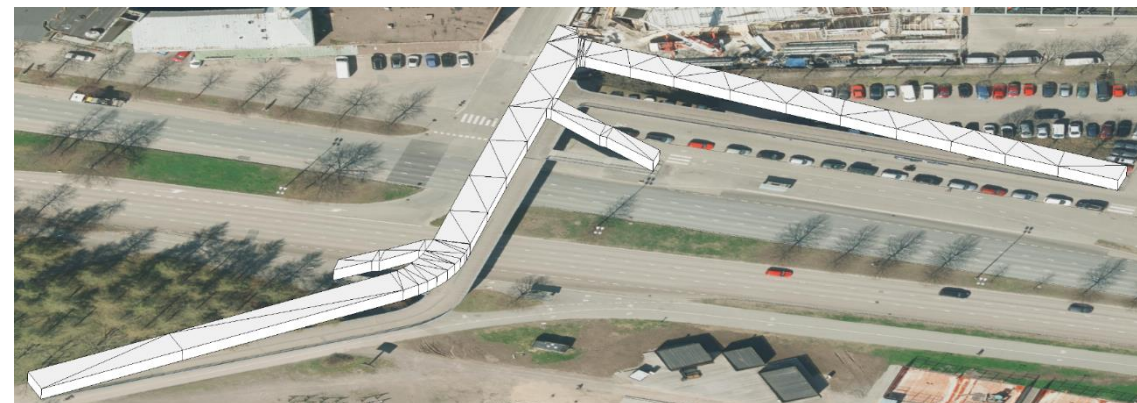


Figure 32. The distance between the cover and the bottom was set to be two meters, so that the bridges would be more visible in the model

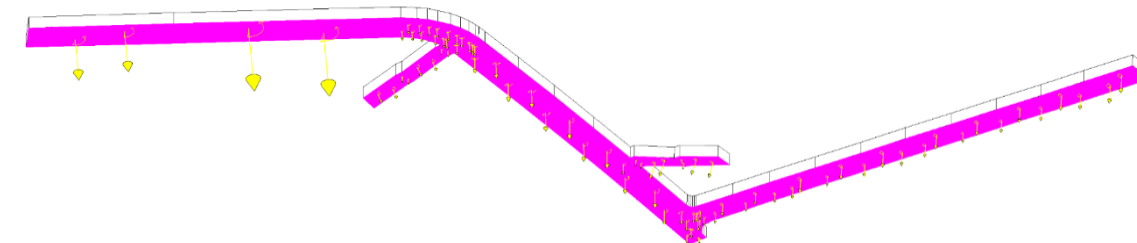


Figure 33. Defining the correct direction of the normals of the bottom

The bridges' geometry data was combined with the bridges' registration data, whereby each bridge received attributes from the bridge register's database. The work phase was carried out in an [FME](#) application, which created a code between the geometry and the table containing the attributes from the bridge register. Based on the code, the application found the correct bridge information from the bridge register for 28 bridges with the help of geocoding. No information was found in the bridge data for one of the bridges, so the name on the map was given to it. Of the vector data, three bridges had to be completely abandoned because one of them turned out to be a dock, one was a ramp of a house, and one was just an insignificant object. Finally, 29 bridges were converted to the CityGML format using the FME application, whereby each bridge object received a CityGML ID code that allows the object to be traced and edited or updated.

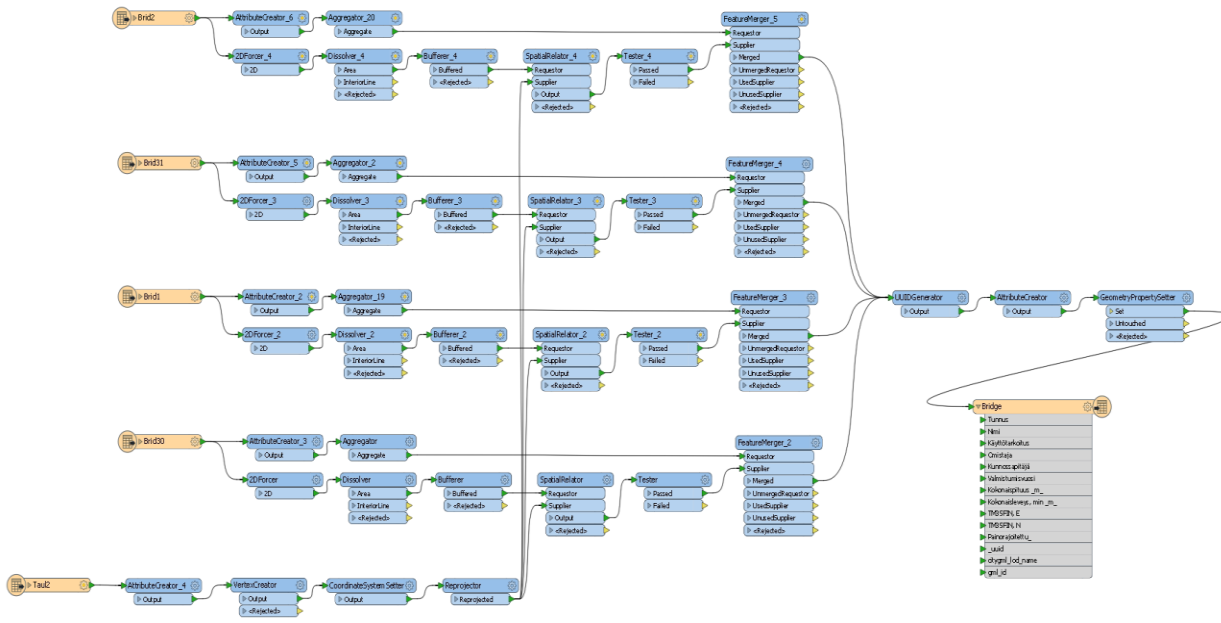


Figure 34. Combining bridge geometry and registry information together, and producing the final CityGML model in the FME application

**Observations:** In addition to existing bridges, two planned bridges were modelled. Due to modelling accuracy and time management, the planned bridges are not modelled in the same way as the 29 bridges mentioned above. The planned bridges are modelled on the same source data as the planned buildings, and at the level of accuracy they are in the source data. The planned bridges were given a new feature, *Sillan\_tila* (=the bridge state), whose value in this case is "design".

The bridges were defined at LoD1 level, although bridge support structures such as piles, posts, etc. are missing. No registry or geometry information was available for the support structures, so their manual modelling was not considered necessary for the overall city information model. The intersection of the larger bridges and the ground was very carefully considered, and the ground was modified accordingly, for example, in the case of simulations, the model must not have any significant errors.

Validations of all CityGML objects were carried out with the help of the geometric validation tools for 3D primitives and schema (CityGML schema validator) developed by the Geotechnical Research Group of the University of Delft. The [Val3Dity](#) tool validates the object geometrically and the [Schema](#) tool checks the completeness of the schema. If something does not match, the applications tell you where the problem can be found, making it easier to fix it.

There is still some development needed editing the attribute data to match the pre-defined characteristics of the standard. However, this can be done quite easily even afterwards.

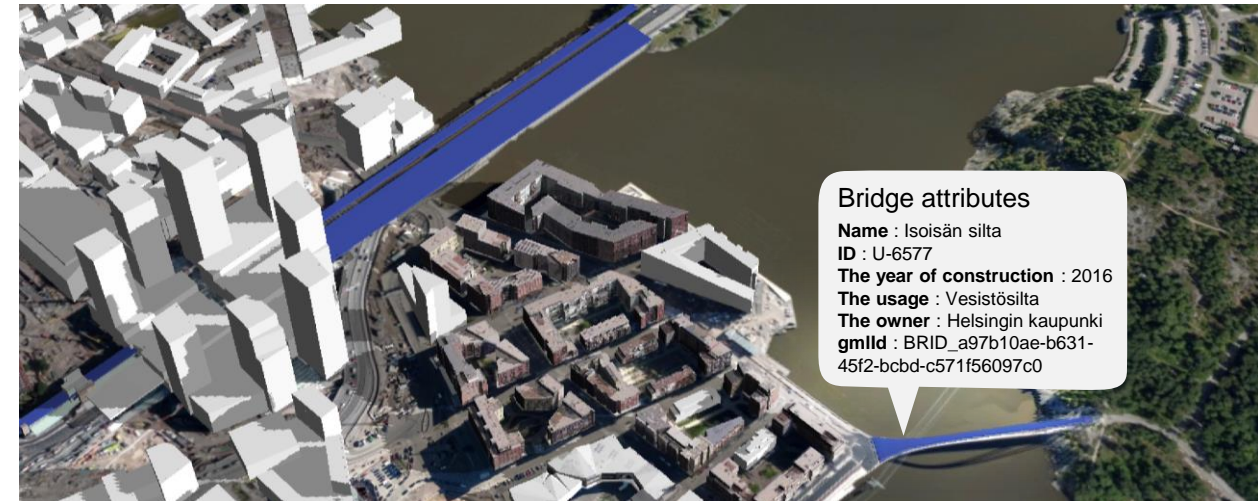


Figure 35. CityGML bridges in the city information model (blue objects)

## The CityGML Terrain Model

The terrain model is an essential part of the city information model. A digital terrain model (DTM) works in the *Relief* extension module. In CityGML, the terrain surface is presented as a *ReliefFeature* and the accuracy level can vary between LoD 0–4. A *ReliefFeature* consists of one or more *\_ReliefComponent* classes. Because *ReliefFeature* and *\_ReliefComponent* are *\_CityObject* derivatives, their properties and relationships are inherited. The *ReliefFeature* can consist of parallel ground types: grid elements, a triangulated irregular network (TIN), and contour lines or mass points. These four types are implemented with corresponding geometries: the grids are *RectifiedGridCoverage*, the contour lines are *MultiCurve*, the mass points are *MultiPoint*, and the triangles are either *TriangulatedSurface* or *TIN*.

In the CityGML model, four types of terrain models can be combined in many ways. First, each type can be represented at different levels of accuracy. Secondly, some of the terrain can be described by a combination of several types, such as grids and contour lines or triangles and contour lines. Thirdly, a certain area can be represented in the same plane on different terrain models, and each terrain model has a certain spatial dimension. Often the grid model is accurate enough to represent simpler terrain shapes, but in a more varied terrain, a TIN model is used if the level of accuracy of the CityGML model requires it.

[OGC City Geography Mark-up Language (CityGML) Encoding Standard v2.0,  
<https://www.opengeospatial.org/standards/citygml>]

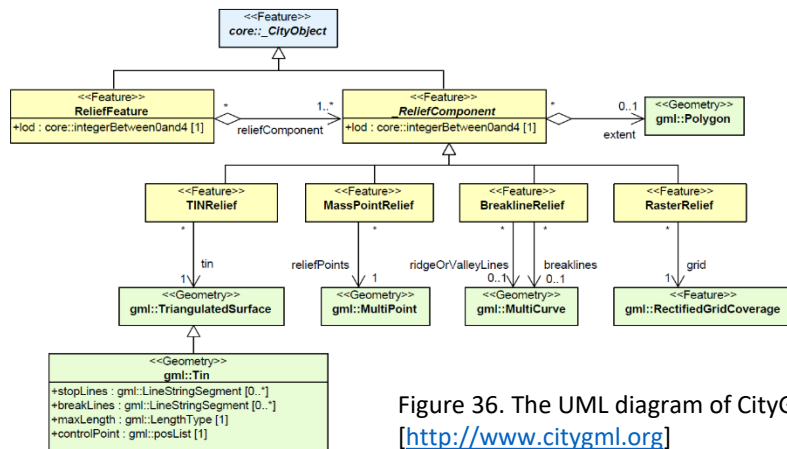


Figure 36. The UML diagram of CityGML terrain  
[\[http://www.citygml.org\]](http://www.citygml.org)

```

<?xml version="1.0" encoding="UTF-8"?>
<CityModel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/citygml/2.0 ./CityGML_2.0/CityGML.xsd"
  xmlns="http://www.opengis.net/citygml/2.0" xmlns:xAL="urn:oasis:names:tc:ciq:xsdschema:xAL:2.0"
  xmlns:app="http://www.opengis.net/citygml/appearance/2.0"
  xmlns:wtr="http://www.opengis.net/citygml/waterbody/2.0"
  xmlns:gen="http://www.opengis.net/citygml/generics/2.0"
  xmlns:luse="http://www.opengis.net/citygml/landuse/2.0"
  xmlns:tran="http://www.opengis.net/citygml/transportation/2.0"
  xmlns:grp="http://www.opengis.net/citygml/cityobjectgroup/2.0"
  xmlns:veg="http://www.opengis.net/citygml/vegetation/2.0"
  xmlns:frn="http://www.opengis.net/citygml/cityfurniture/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:tex="http://www.opengis.net/citygml/textures/2.0" xmlns:gml="http://www.opengis.net/gml"
  xmlns:brid="http://www.opengis.net/citygml/bridge/2.0"
  xmlns:core="http://www.opengis.net/citygml/2.0"
  xmlns:dem="http://www.opengis.net/citygml/relief/2.0"
  xmlns:blgd="http://www.opengis.net/citygml/building/2.0">
  <gml:boundedBy>
    <gml:Envelope srsDimension="3" srsName="urn:ogc:def:crs:EPSG::3879">
      <gml:lowerCorner>25497328.48 6673625.36 -4.35</gml:lowerCorner>
      <gml:upperCorner>25499264.27 6676487.97 9.97</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <core:cityObjectMember>
    <dem:ReliefFeature gml:id="Terrain_Kalasatama">
      <dem:lod>3</dem:lod>
      <dem:reliefComponent>
        <dem:TINRelief>
          <dem:lod>3</dem:lod>
          <dem:tin>
            <gml:TriangulatedSurface gml:id="PolyID1_Terrain_Kalasatama">
              <gml:trianglePatches>
                <gml:Triangle>
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList srsDimension="3">25497385.56
                        6673885.77 2 25497380.19 6673895 2
                        25497370.97 6673899.95 2 25497385.56
                        6673885.77 2</gml:posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Triangle>
                <gml:Triangle>
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList srsDimension="3">25497385.56
                        6673885.77 2 25497370.97 6673899.95
                        2 25497356.55 6673899.12 2 25497385.56
                        6673885.77 2</gml:posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Triangle>
              </gml:trianglePatches>
            </gml:TriangulatedSurface>
          </dem:tin>
        </dem:TINRelief>
      </dem:reliefComponent>
    </dem:ReliefFeature>
  </core:cityObjectMember>

```

Figure 37. The first lines of the Kalasatama CityGML terrain model

## Modelling the CityGML Terrain Model on a Theoretical Level

At this level, one alternative production process for the CityGML model was conducted on a theoretical level. The final CityGML terrain model will be produced later. The terrain representational data, the steps of which are described on the next page, serves as the source data for the CityGML modelling of the terrain model. It was decided to describe the CityGML terrain model of the Kalasatama project area as a triangulated irregular network (TIN) in order to bring out the terrain as accurately as possible. The source data can be read, for example, in an FME application that can convert the data into a CityGML model. Figure 37 shows the first lines of a possible CityGML terrain model.

## Production of the Terrain Model

The terrain of the Kalasatama area is part of the entire city-wide terrain model and the planned terrain model for the future.

### The Source Data

The most important source data for producing the terrain model was a laser scanned point cloud from the summer of 2017. The laser scanning was performed by Terratec with an Optec Titan multi-channel lens (wavelengths 532/1064/1550 nm) at a height of 550 meters with a 50 % side cover. The point density was about 20 points per square meter. The point cloud was classified by conventional methods using Terrasolid software according to the ASPR classification. The ground level key points from the point cloud classes were used to produce the terrain model.

The footprint polygons of the base map, the watercourse, and the bridge deck data (CAD-shaped 2.5D data that roughly depicts the position and dimension of a bridge deck) were also used in classification and modelling the terrain model.

The terrain of the new construction areas was derived from plan-based elevation data; while a decided base elevation was used for the rest.

### Creation of the Model

It was decided to implement the Kalasatama terrain model as a triangulated irregular network (TIN) instead of a grid model, in order to make the model more detailed and lighter. The advantage of a TIN model is that wide similar areas can be implemented with just a few triangles, while the grid pattern always has the same number of grid ranges in a given size range, regardless of the characteristics of the region.

Areas (triangles) that should be displayed as their own theme were removed from the terrain model. This was done for waters, as they are modelled separately as their own objects. For buildings, the terrain model can be opened or unified. The choice depends on whether you want to look at the ground as an independent object.



Figure 38. Terrain model from the area in a planning stage

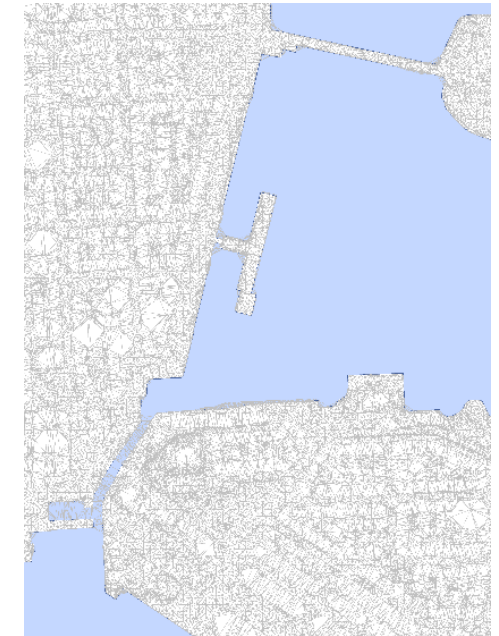


Figure 39. The current stage terrain model

**Observations:** The terrain models are CityGML compliant, but they have not yet been added to the database as CityGML. An LoD3-level *dem:TINRelief* object will be created from the terrain models in the near future. It is challenging to find a level that is accurate enough, but at the same time as light as possible, so that database operations, analyses and simulations can succeed. The *TINRelief* object also places restrictions on surfaces (e.g. overlaps and verticals) that need to be resolved.

## CityGML Land Use Areas

*LandUse* objects can be used to describe areas for specific land use, but also to describe areas with specific soil or vegetation such as sand, rock, forest or meadows, that is, to describe their physical appearance. In the CityGML model, both land use and soil are presented under the *LandUse* class, but are refined with attribute data.

Each *LandUse* object must have *class*, *function* and *usage* attributes. *Class* represents land use areas such as residential or industrial areas; *function* describes the purpose of a land use area, such as a residential area; *usage* is used only if the area's usage differs from its land use category.

The *LandUse* object is assigned to all LoD levels (0–4) and its geometry may vary depending on the level. Creating a *LandUse* object geometry requires 3D coordinate information. The object must also be a *MultiSurface* object so that the appearance features can be used.

[OGC City Geography Mark-up Language (CityGML) Encoding Standard v2.0, <https://www.opengeospatial.org/standards/citygml>].

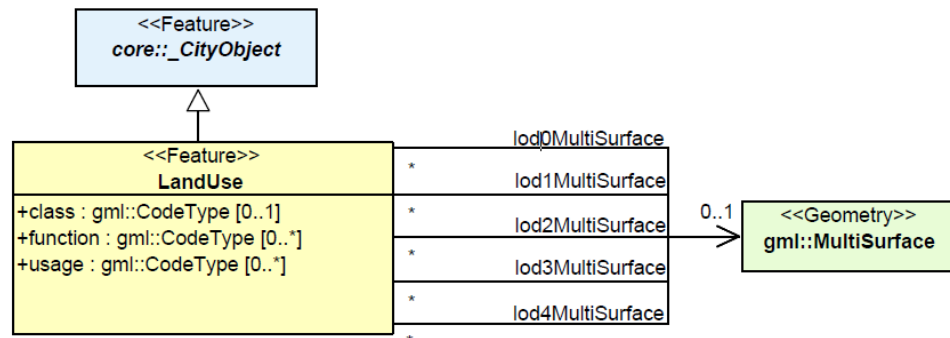


Figure 40. UML diagram of CityGML LandUse [<http://www.citygml.org>]

## Modelling the CityGML Land Use Areas on a Theoretical Level

Within this project, one possible process for modelling CityGML land use areas was planned. The actual modelling may be implemented later. The CityGML modelling of the land use areas of the Kalasatama project area is based on the CityGML terrain model produced from the project area, which in this case is classified according to land use. The objective is to obtain land use information for all terrain models. For example, information contained in the City of Helsinki's register of public spaces could be used for classification. Possible land use object groups could be those in Table 1.

Table 1. The different classes of CityGML land use and source data used in classification

class_CodeSpace : class	Data source
1000 : Settlement Area	2D data from WFS service (area's of specific land plots)
1100 : Undeveloped Area	The remaining areas from terrain model
2000 : Traffic	Register of public spaces (streets)
3000 : Vegetation	Register of public spaces (green areas)
4000 : Water	Digitized waterbody in DGN format

According to the table, land use areas can be classified into five categories: settlement areas, underdeveloped areas, traffic, vegetation and water. The register of public spaces provides information on street and green areas, and information on built-up areas can be obtained through the Helsinki City open data WFS service. The register of public spaces is also available on the open data WFS service. Water bodies are categorized using vector-form data of the water areas. The source data for land use modelling is read directly from the interface as well as from local data files. Unclassified areas are still underdeveloped.

## Twin Models as Open Data

The reality mesh model and the CityGML city information model are shared as open data. The free availability of models creates a new order of opportunity for the utilization of the city model. With this project, open data is especially hoped to reach a wide range of construction and real estate actors and their stakeholders.

City model data sets are available under license of [Creative Commons Attribution 4.0 \(CC BY 4.0\)](#).

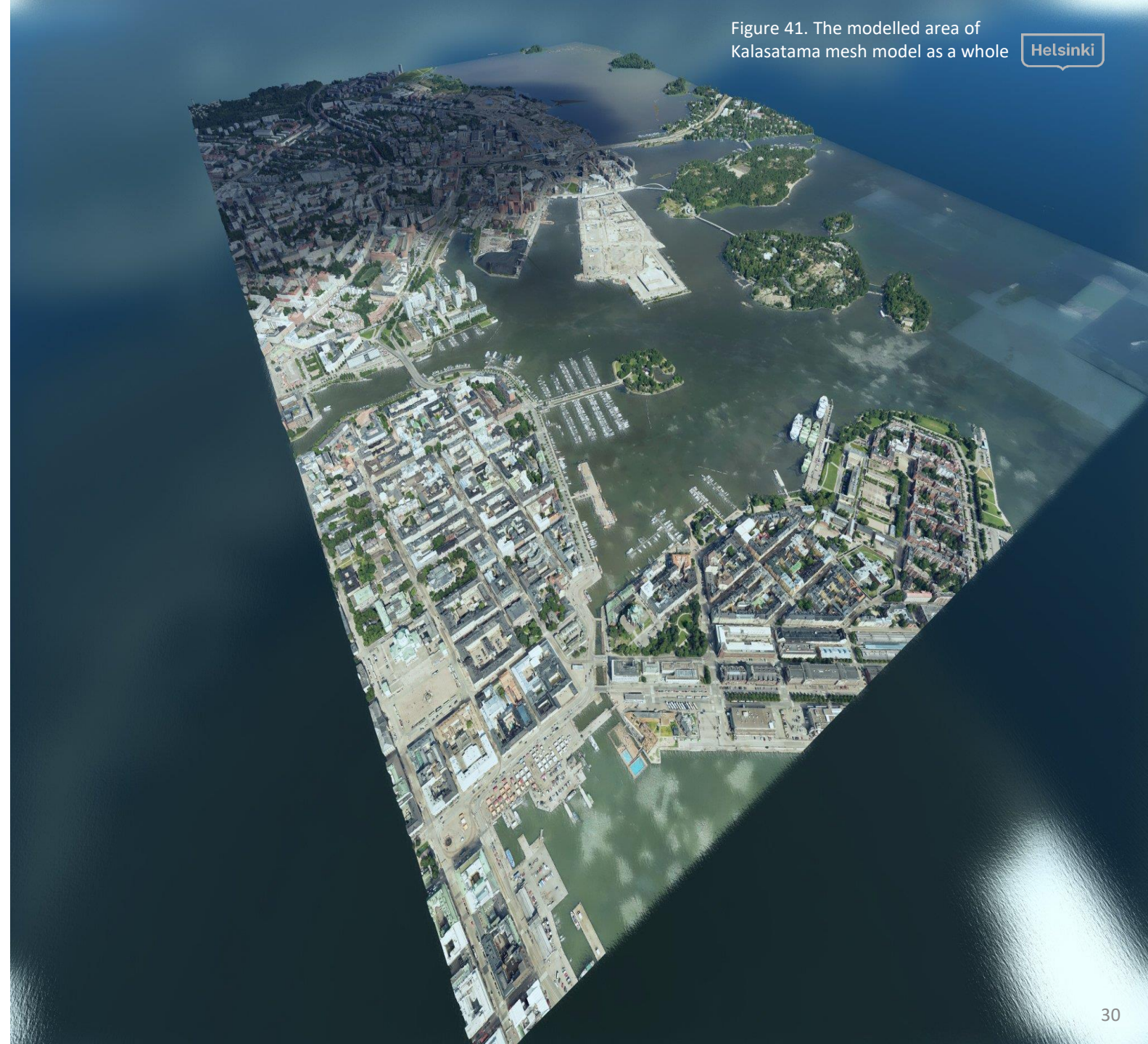
Particular attention has been paid to the usability of the models' data in open source tools, whereby, for example, educational institutions and developer communities can build their own test environments without special procurement.

## The Kalasatama Mesh Model

The mesh model can be downloaded in DAE, FBX, OBJ, 3D Tiles and 3MX/3SM formats from Helsinki Region Infoshare (HRI): [https://hri.fi/data/en\\_GB/dataset/helsingin-3d-kaupunkimalli](https://hri.fi/data/en_GB/dataset/helsingin-3d-kaupunkimalli).

The Kalasatama mesh model is also viewable and navigable on the CESIUM platform with WebGL-enabled browsers at: <https://kartta.hel.fi/3d/mesh/Kalasatama>.

Figure 41. The modelled area of Kalasatama mesh model as a whole



## The Mesh Model in the Unity Streaming Service

The mesh model for Kalasatama was available for use as open data, free of charge, during the period 1 June – 31 December 2018 also via the Unity online streaming service provided by our partner [UMBRA](#). The data can still be downloaded by request and can be navigated at: <http://bit.ly/2t9RSCA>.


Umbracomposit™ is the leading 3D content optimization platform for the Unity gaming engine. It provides an automated solution for processing, streaming, and displaying content on any device, regardless of device performance. The platform provides a cloud-based backend service for 3D content, including processing, storage, and streaming. The service can handle very large files and has no restrictions on the size of the templates. The service is based on Umbracomposit's visibility optimization technology, LoD technology and data compression algorithms. The model can also be viewed using AR and VR compatible mobile platforms.

**WHAT IS UMBRA?**

**UMBRA COMPOSIT™**

Is a cloud-based system for optimizing and viewing high-detail 3D data in realtime solutions.

Not only can Umbracomposit optimize any size 3D data sets, it has the ability to stream in your 3D content based on predefined memory limits, visibility and distance to the camera in runtime.



**Submit for the Helsinki 3D Content**

First Name

Last Name

Company name

Email\*

Figure 42. Kalasatama's mesh model data is available to everybody

The ability to use the 3D city information model on the Unity gaming platform opens a whole new area of potential use for large city model files. The functionality of game engines makes it possible “to convert city services into games” or create completely new syntheses. For example, the planning of city events, the handling of applications by the authorities, the organization of crowdfunding, marketing and furnishing arrangements can be handled by a browser-based game engine application using the 3D model.

The City of Helsinki's Open Data Policy (CC BY 4.0 license) also allows the development of commercial services such as UMBRA. There are just under 100 gaming companies in Helsinki and practically all of them have Unity expertise. Channelling the innovation and creative potential of these companies into city-based services and utility games will benefit both Helsinki and businesses.

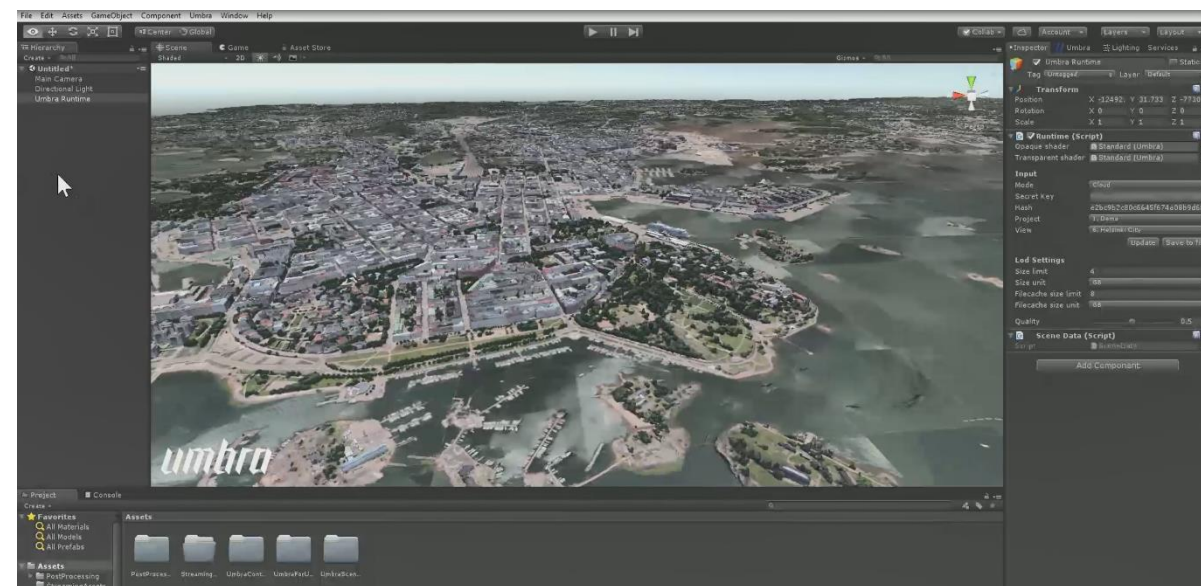


Figure 43. 3D mesh model on the Umbracomposit™ application

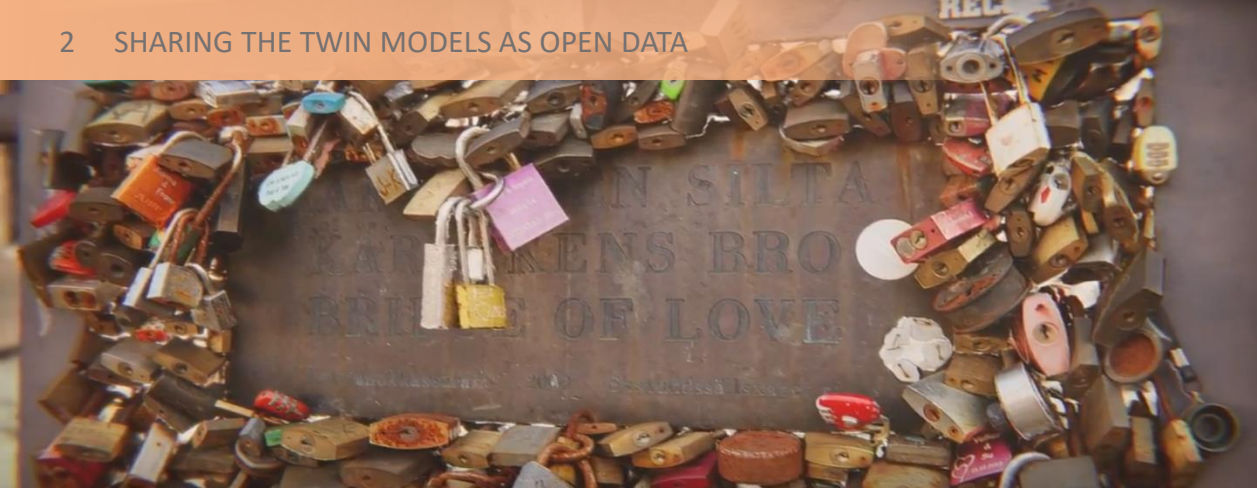


Figure 44. Highly detailed mesh models can be uploaded to the Umbra Composit™ service and its UNITY application as base data

**For a developer** the streaming service means:

- the ability to use large 3D data in UNITY applications without size limitations and optimization
  - for example, the whole of Helsinki consists of 700 GB of data, and 2 billion polygons
- to minimize the size and load time of your application
- polygon count and memory sufficiency no longer cause problems
- a refresh rate of 60 to 120 frames can be achieved in all situations

**For the end user**, the service means a smooth operating experience on all smart devices, including phones and tablets. The wide availability of services is also in line with the goals of the City of Helsinki, as it aims to make the service accessible to as many people as possible.

**According to Umbra's developers**, the modelling will be further refined as cities' data collection and processing capacity increase. It is already possible to produce millimeter-like mesh models using automated methods, as can be seen in Figure 44. Only the accuracy and computational capacity of the base data will limit the production and maintenance of the models. In the future, it will be possible to equip vehicles with laser scanners connected to the data network. The real-time information from these will allow

the city model to be constantly updated and its accuracy will be selected according to the need for use. Thus, it will be possible to achieve streaming of real-time city information model for useful applications.

Umbra offers a 14-day free trial of Composit™, the fastest and easiest way to visualize large 3D environments on any VR or AR compatible device.

Read the blog post about Helsinki 3D data on Umbra's website:

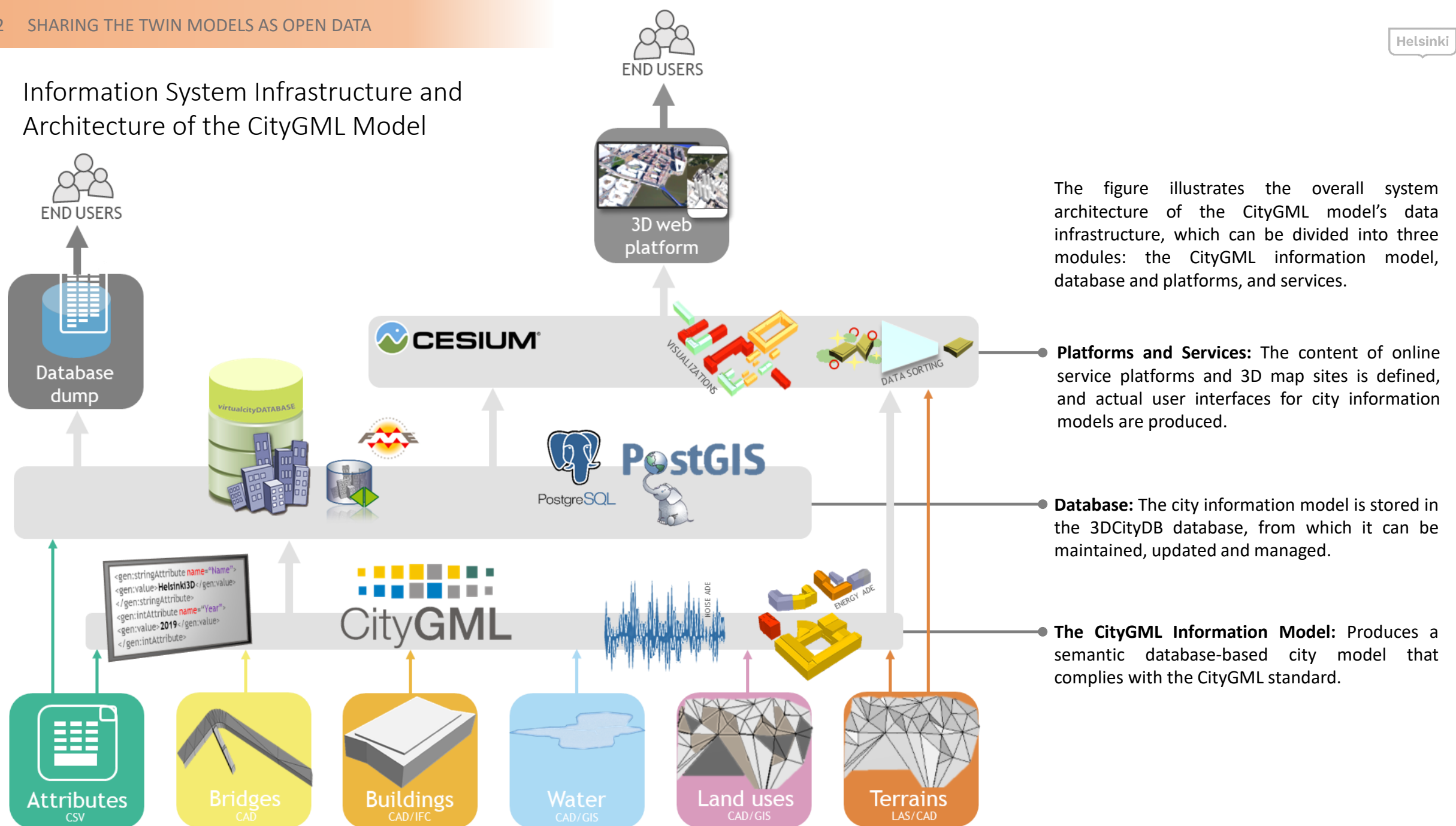
<https://blog.umbra3d.com/blog/large-and-complex-3d-content-in-unity>

Watch a video about Unity streaming:

<https://youtu.be/eQpXqcTiGQk>



## Information System Infrastructure and Architecture of the CityGML Model



The figure illustrates the overall system architecture of the CityGML model's data infrastructure, which can be divided into three modules: the CityGML information model, database and platforms, and services.

• **Platforms and Services:** The content of online service platforms and 3D map sites is defined, and actual user interfaces for city information models are produced.

• **Database:** The city information model is stored in the 3DCityDB database, from which it can be maintained, updated and managed.

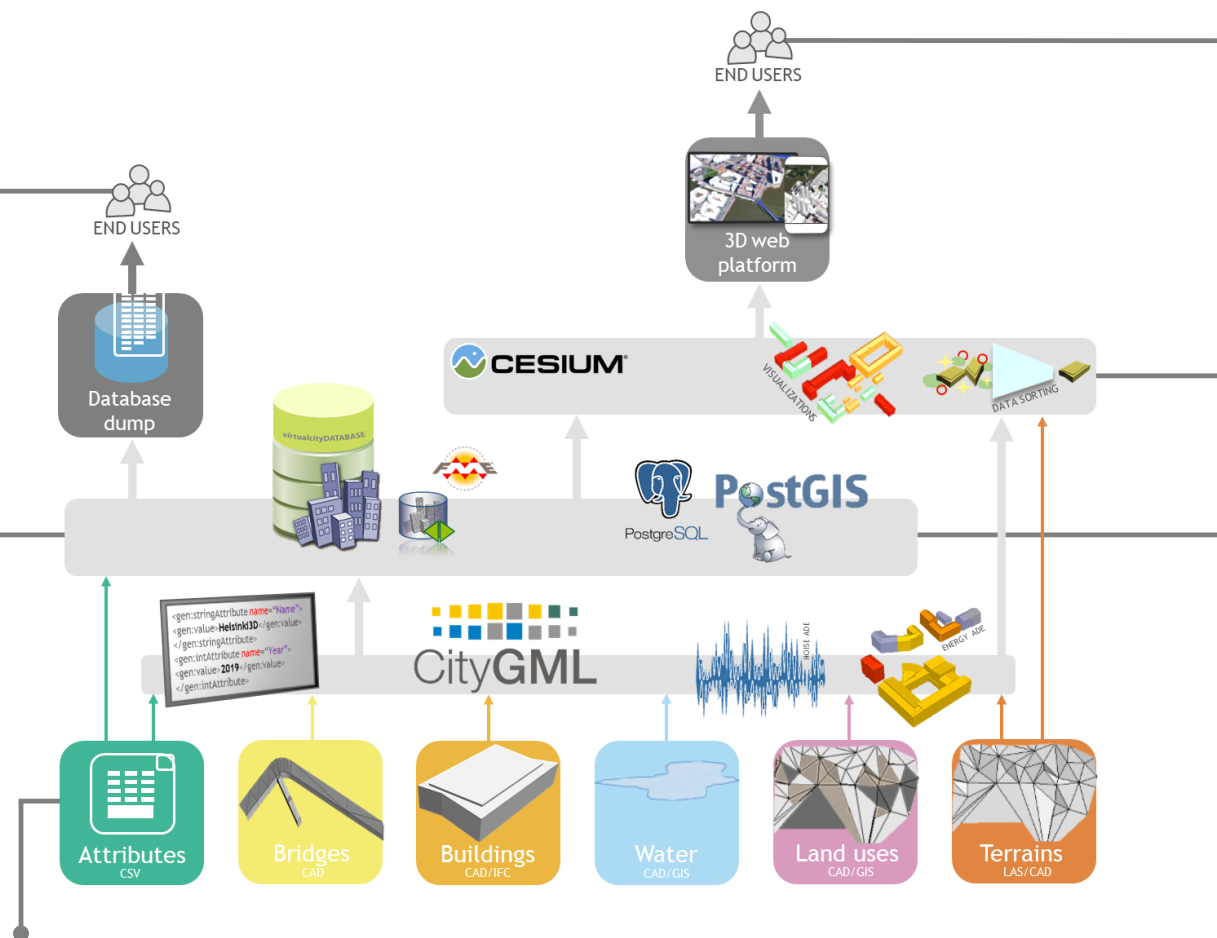
• **The CityGML Information Model:** Produces a semantic database-based city model that complies with the CityGML standard.

Figure 45. Overall architecture of CityGML model's data infrastructure

## The Database Dump of Kalasatama's CityGML Model

The database dump is available at [https://hri.fi/data/en\\_GB/dataset/helsinki-3d-kaupunkimalli](https://hri.fi/data/en_GB/dataset/helsinki-3d-kaupunkimalli).

Exporting the data to the database is carried out with the Importer/Exporter tool. The same tool can also read information from the database. A downloadable database dump can be used to set up your own CityGML development environment.



The CityGML city information model of Helsinki and the CityGML model of Kalasatama are available at: <https://kartta.hel.fi/3d>.

The city information model's web service works with WebGL-supported browsers. The CityGML model's online services have been created with the help of [virtualcitySYSTEMS'](#) products. The user interface uses an open [CESIUM](#) platform that can be accessed with JavaScript code.

The CityGML model of Helsinki is stored in the [3D City Database called 3DCityDB](#) which has been extended with a [virtualcitySUITE](#) product provided by virtualcitySYSTEMS. The 3DCityDB is used for storing, using and maintaining the city information model. The 3DCityDB is a free, open source database of 3D geospatial data sets. The 3DCityDB uses the [PostgreSQL](#) database and its [PostGIS](#) spatial extension. PostGIS is a free spatial extension for PostgreSQL. The [pgAdmin](#) management system for the 3DCityDB is also based on open source. The contents of the relational database largely correspond to those of the CityGML information model. In this project, CityGML-modelled objects were read into the 3DCityDB.

New objects can be modelled in the city information model by using various source data such as 3D or CAD models, building façade or roof textures in image format, attribute data in table format or laser scanning data. In this project, the planned buildings of the Kalasatama area were modelled with both existing bridges and two planned bridges.

## Smart Kalasatama

Smart Kalasatama is the main partner of the Kalasatama digital twins pilot project and is the flagship of smart city development in Helsinki. Throughout the area, the Smart Kalasatama project, together with residents, businesses and other actors, will explore new intelligent solutions based on smoothness of use, user-orientation and inclusive city planning. Intelligent experiments are divided into three themes: energy, sustainable mobility and smart everyday services. Under the energy theme, solutions for solar energy utilization and carbon neutrality are being tested and presented. Smart everyday services have been named a 25/7 theme, as intelligent housing services aim to provide additional daily hours of leisure time, and intelligent solutions for the complex include, among other things, multipurpose use and centralization of service packages. The mobility theme aims to improve carbon-neutral mobility with shared electric cars and support for future rail traffic.

The development of Kalasatama is driven by the Living Lab complex, which consists of the area itself, the cooperation networks operating there, and the joint development space called "The Kalasatama Urban Lab".

## The SMART City Platform

Digital twins are intended to be used as a virtual development platform for experiments carried out for the Smart Kalasatama project, thus promoting the planning, implementation, interaction and communication of smart city projects in the Kalasatama region. The model platform enables advanced digital design, testing and communication of projects before their actual implementation. The virtual platform for Smart Kalasatama was built on the browser-based [OpenCities Planner](#) application. OpenCities Planner was acquired with 6Aika funding for Forum Virium Helsinki.

## Creating the Platform

OpenCities Planner is a browser-based application that is suitable for visualizing projects on a real scale. Its strength is that it is also available on mobile devices. It is possible to import vector data from other applications into KMZ or Collada/DAE formats. When the input is completed in the pre-defined coordinate system, it is also placed in the same coordinate system in the OpenCities Planner. GIS data can also be added in shapefile format. Reality mesh models are suitable as a base model for the application, as was done in this case.

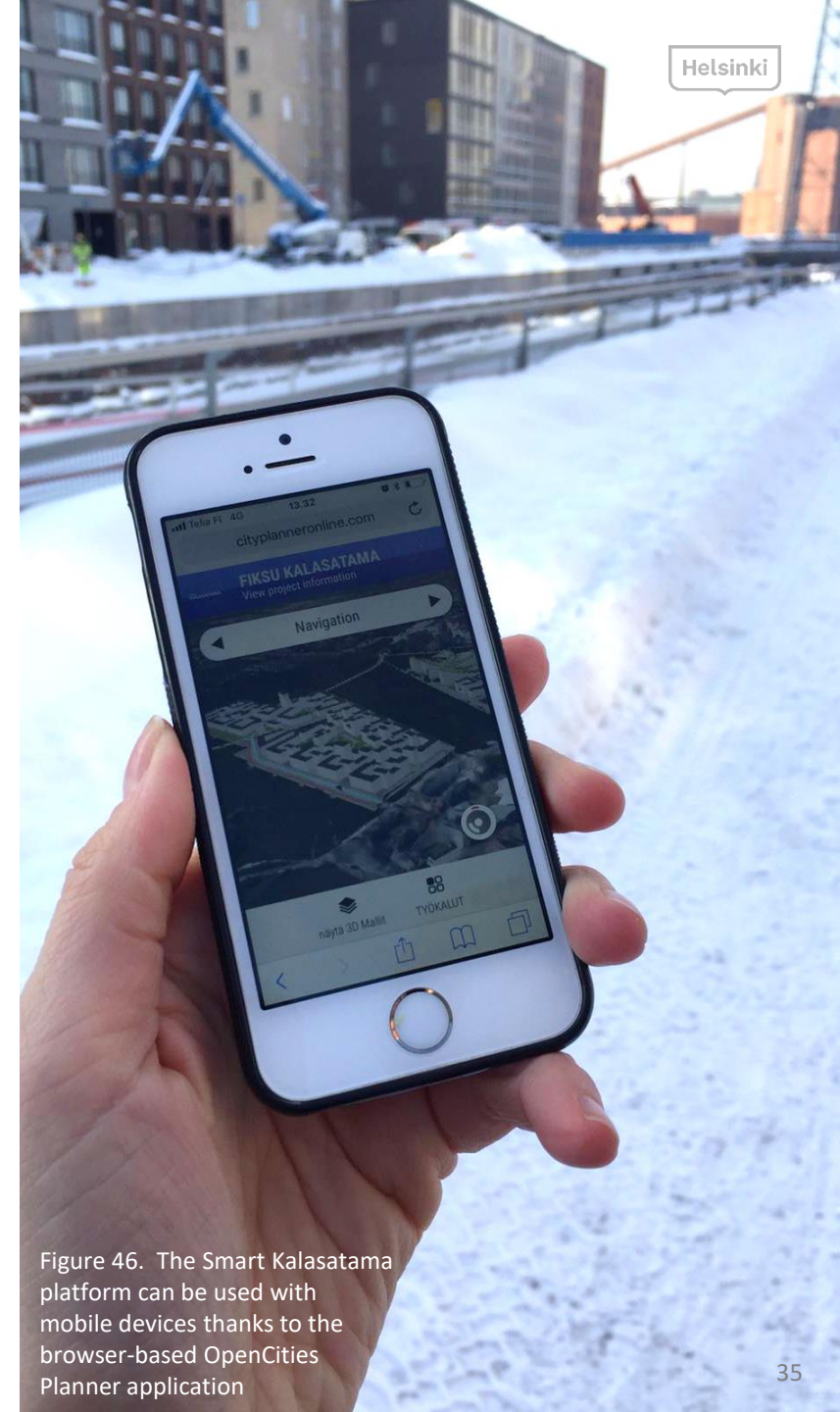


Figure 46. The Smart Kalasatama platform can be used with mobile devices thanks to the browser-based OpenCities Planner application

The mesh model of Kalasatama produced in this project is used as the basis for use by the OpenCities Planner application. Adapting the mesh model to the OpenCities Planner takes place through the application's cloud service run by administrators. 3D models of planned buildings have also been added to the application, as well as terrain models.

### Importing Planned Models to OpenCities Planner Platform

The DAE plan model created with the MicroStation application was not transferred as such to the OpenCities Planner, so the model was first introduced as a [SketchUp](#) application, where it was possible to save the model in another file format, in KMZ. SketchUp does not fully support the city's coordinate system because the coordinates are too large, so the application's accuracy is not sufficient. Thus, the files were transferred between the applications first so that the origin of the model was close to the center of the model. The process is as follows:

1. Create a model with the MicroStation application in city's coordinate system using Connection Edition (CE) release 11, as it better supports the use of mesh models. The terrain model is created with the TerraModeler application that runs on MicroStation.
2. Create a new blank file that will be backed up by a design model and set to zero, that is, to another local coordinate system. Doing this means there is no need to change the coordinate system of the original file.
3. Save visible items in the SketchUp format.
4. Check the file with SketchUp and edit lightly if needed. Save to KMZ format.
5. Import the file to the OpenCities Planner and position the design model manually using the mesh model.
6. Align the mesh model with the terrain of the design model and with terrain of the mesh model hidden, to the extent that the design model changes the terrain, e.g. coastlines, bridges and ground shapes. The mesh model can be levelled by using the Terraform tool of the OpenCities Planner. For example, the visible coal pile in Hanasaari was flattened in the mesh model as it will disappear in the next few years from the new buildings.
7. Finally, the OpenCities Planner user interface is modified to meet the needs of the Smart Kalasatama project.

The platform is available at: <https://cityplanneronline.com/helsinki/kalasatama>.

**Observations:** At the end of 2018, [Bentley Systems](#) acquired *Agency9*, which originally created the OpenCities Planner application. MicroStation is a product owned by Bentley Systems, so in future interoperability between MicroStation and OpenCities Planner is expected to develop. OpenCities Planner was originally called CityPlanner.



Figure 47. The reality mesh model of Kalasatama and the design model for new planned buildings on the browser-based OpenCities Planner platform

## Smart Kalasatama Projects and Innovations

In the early stages of the KIRA-digi project, about 20 Smart Kalasatama's projects were selected because these projects were seen as an opportunity to benefit from the Open Cities Planner platform. However, as work progressed, it was found that some of the projects would be better implemented later on the city information model's platform.

The first implementation was the modelling of the construction projects of the Kalasatama regional construction project described above. Subsequently, the current projects of the Smart Kalasatama project were brought into the platform as a separate layer. The layer structure supports project communication well and makes it easier to find projects locally. Currently, a separate research project is underway in collaboration with the MySmartLife project and the Technical Research Centre of Finland (VTT) is studying the presentation of real-time energy data in the model.

Points of interest were created in the OpenCities Planner to describe ongoing projects. The items were named and an icon depicting the project was selected. A project-related web page (on click > embed code) was linked behind the object. Some websites had problems and did not open in the window, so in these cases the link opens in their own tab.

The platform is available at: <https://cityplanneronline.com/helsinki/fiksukalasatama>.

In the future, these bases can be used as a backdrop when creating various queries, for instance. The goal is to refine the way in which the progress of a regional construction project is presented, for example by marking the completion schedule for each item and some contact information in the model. It has been mentioned that a finalized 3D model might make some residents assume that the plan is final. However, on the basis of experience, users learn to read patterns and, for example, displaying buildings as light figures helps to emphasize their "incompleteness" and gives a more accurate picture of the planning phase.

**Observations:** As a practical lesson it can be said that projects are most easily implemented with a technical implementer who is able to handle models and create layers with the OpenCities Planner. In addition, there is a need for a content expert who can identify and create the necessary information content. The maintenance and coordination of the model must be adequately resourced and clearly defined. There is also a need for more experience as to who all the final users of the model are, i.e., how developers and companies can take advantage of the model.

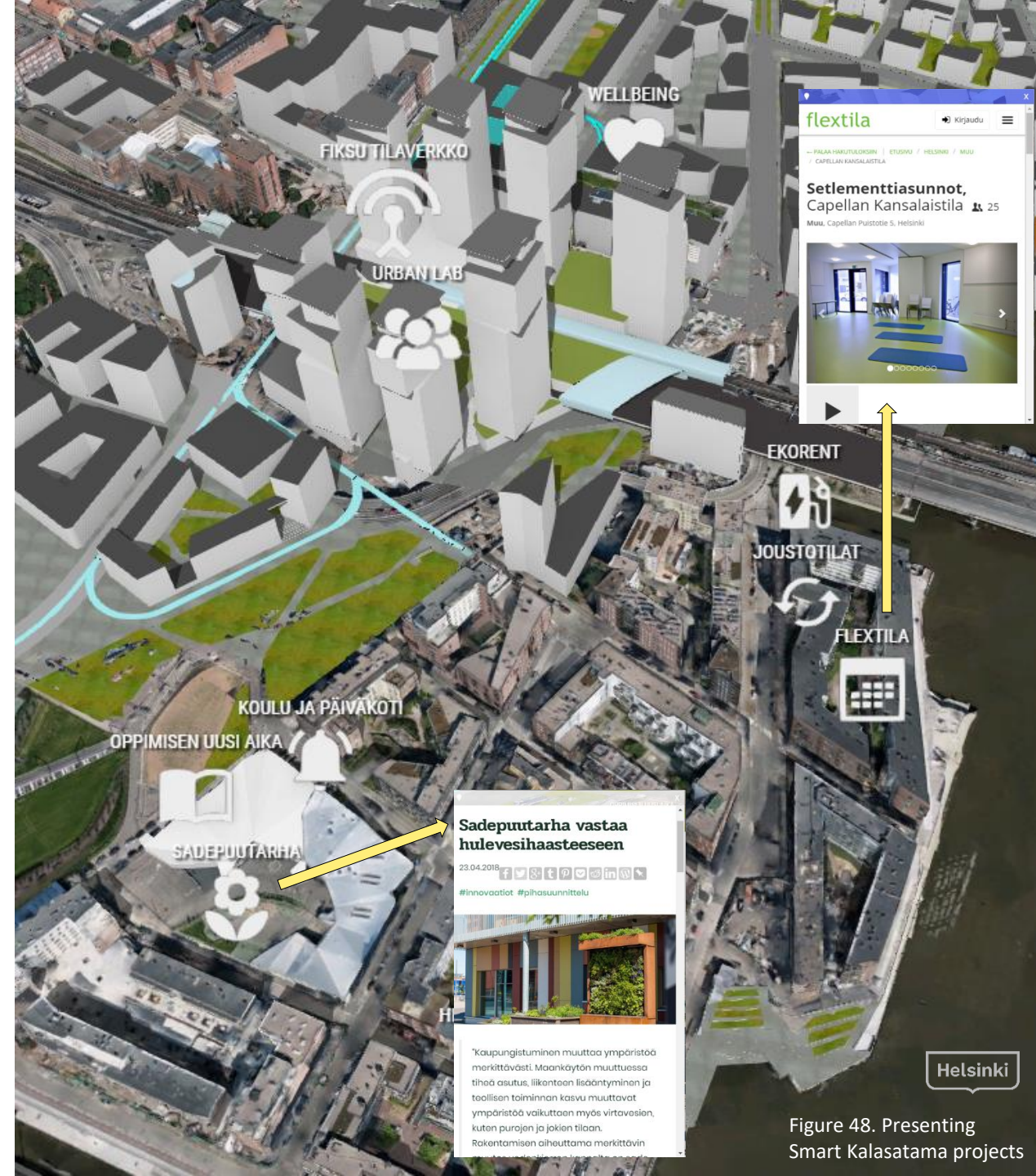


Figure 48. Presenting Smart Kalasatama projects

## Smart Interaction

The project also tested how the OpenCities Planner can be used as a platform for interaction and piloted the use of queries in the Smart Kalasatama project.

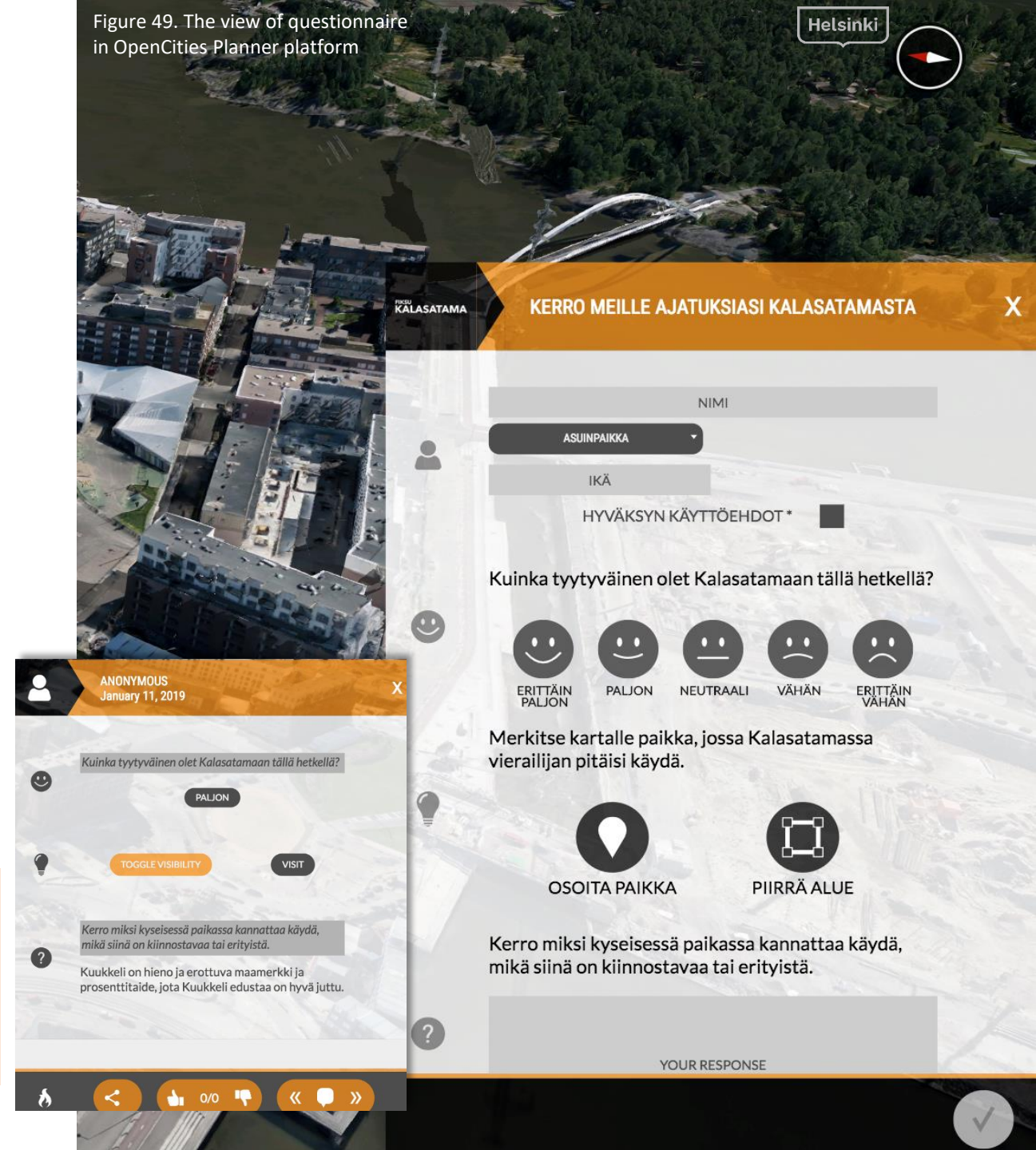
The OpenCities Planner platform was used to introduce the 3D city model of Kalasatama at the Urban Lab in Kalasatama. The platform was used in its offline mode, as it does not require an Internet connection. The model worked well and was also easy to use on a touch screen of a phone. At the same time, an interactive questionnaire was carried out to find out what local people in the area would recommend to visitors, and what kind of smart solutions they currently use.

Building an interactive PPGIS survey (Public Participation GIS) was easy on the OpenCities Planner using some ready-made question modules. However, the responses were presented in a sidebar which made it difficult to locate the answer for a particular locale if the respondent plotted multiple locations for different answers in the questionnaire.

The advantages of using a 3D model to implement built environment queries compared to a 2D map is that it illustrates the plans in a more concrete way. Previously, the Maptionnaire tool, which is based on a 2D map, had been used in the Smart Kalasatama project, so a comparison of the tools was possible.

**Observations:** The challenge was to meet the technical requirements for offline use, which was not immediately achieved with the earlier model, but had to be updated. A few challenges in creating questionnaires were identified. The questionnaire reveals the respondent's information, which is questionable from the point of view of anonymity, and it was difficult to implement a solution where the respondent's information would only remain with the inquirer, but that the content of the answers would still be visible. From the user perspective, the main issue was the capacity of the user's device, which affects how fast the 3D model works and the quality.

Figure 49. The view of questionnaire in OpenCities Planner platform

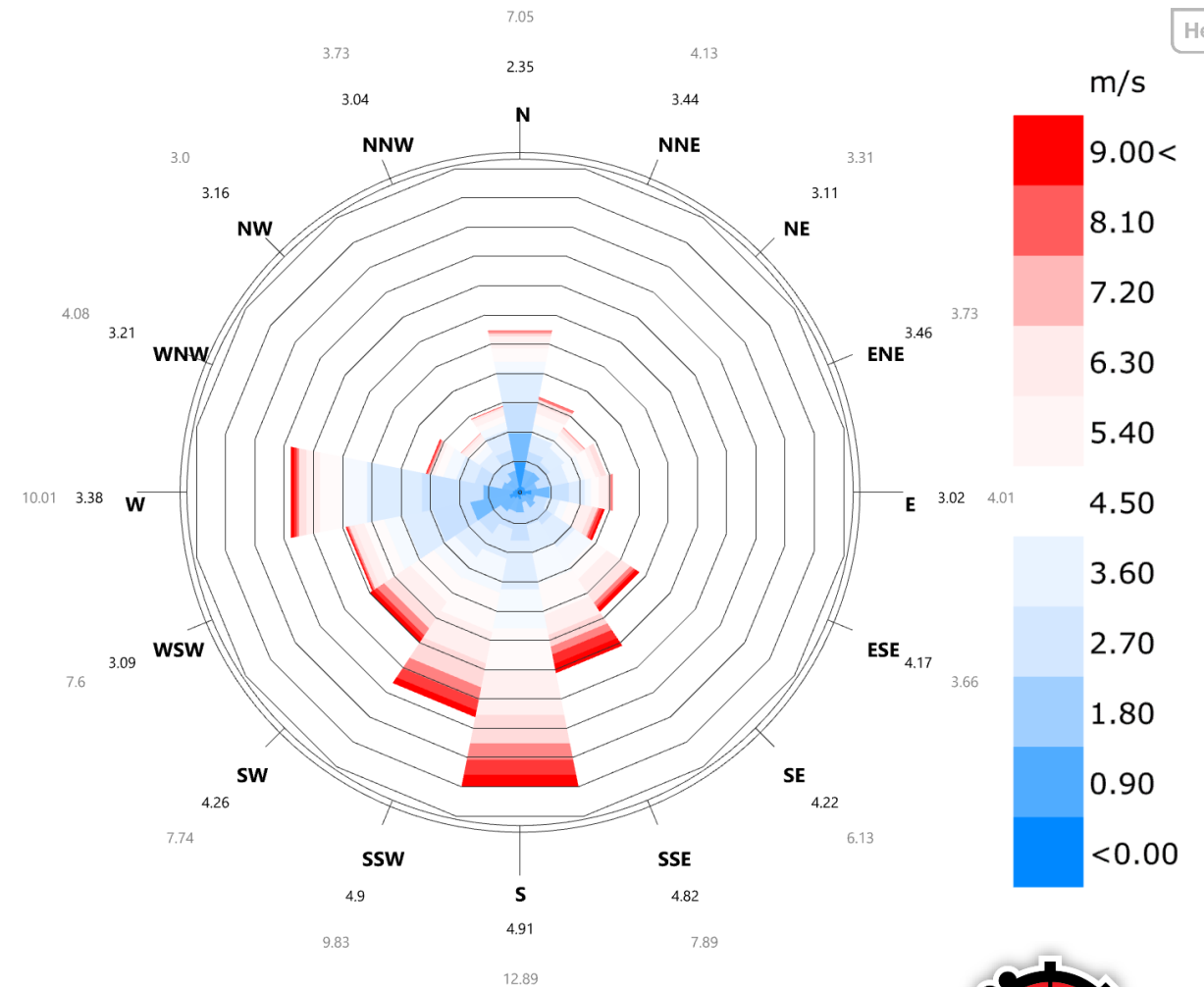


## Wind Simulation

Wind simulations are necessary especially when planning areas for efficient construction close to the sea. Large constructed units and high-rise buildings can increase the negative effects of wind in the city. The wind has a significant impact on the micro climate, comfort and safety of the streets and pedestrian areas. 3D virtual models make it possible to study the effects of the wind for the current state and for future plans. The wind in the city can be explored with computational fluid dynamic techniques. Implementing wind simulations at the earliest possible stage of planning provides a good basis for planning a comfortable and safe urban environment.

### Exploring the Speed and Direction of the Wind

For wind simulations, speed and direction parameters need to be determined. Wind directions and velocities were investigated using the [Ladybug Tools](#) software (built using Grasshopper) to carry out [a wind rose analysis](#). Ladybug Tools are a collection of free computer applications that support environmental planning and development. Ladybug is one of the most comprehensive software programs that combines 3D modelling and environmental simulations. The Ladybug tool directly uses [EnergyPlus Weather](#) statistic data, and the data available from Helsinki dates back to the 1990s. It would have been possible to update the data, but it was not considered necessary, as according to the [Finnish Meteorological Institute \(2018\)](#), the wind in the Gulf of Finland could grow by two to four percentage points from 2000 to 2100. In addition, there have been no significant changes in the wind during the observation period 1971 – 2000. On this basis, the implementation of wind simulations with the wind data of the 1990s was perceived to be sufficiently accurate. The wind roses on this and the next page have been made with this data.



Wind-Rose  
 HELSINKI\_FIN  
 1 JAN 1:00 - 31 DEC 24:00  
 Hourly Data: Wind Speed (m/s)  
 Calm for 1.20% of the time = 105 hours.  
 Each closed polyline shows frequency of 1.3%. = 112 hours.



Figure 50. The Grasshopper application's Ladybug Tools creates a wind rose that shows the percentage of wind speeds and directions for the whole year. Significantly the wind blows most strongly in Helsinki from the south and south-west, i.e. from the sea

Wind roses can easily tell the average direction and speed of the wind in Helsinki. Figure 51 shows the monthly wind roses calculated from the findings of the 90's in Helsinki. Based on these wind roses, the strongest winds in the winter months blow south and south-west. In the spring and summer, the strongest wind comes from the south.

Wind simulations were carried out from one wind direction and one speed at a time. Therefore, it is advisable to base the values of wind speeds and directions on the average of the month or season.

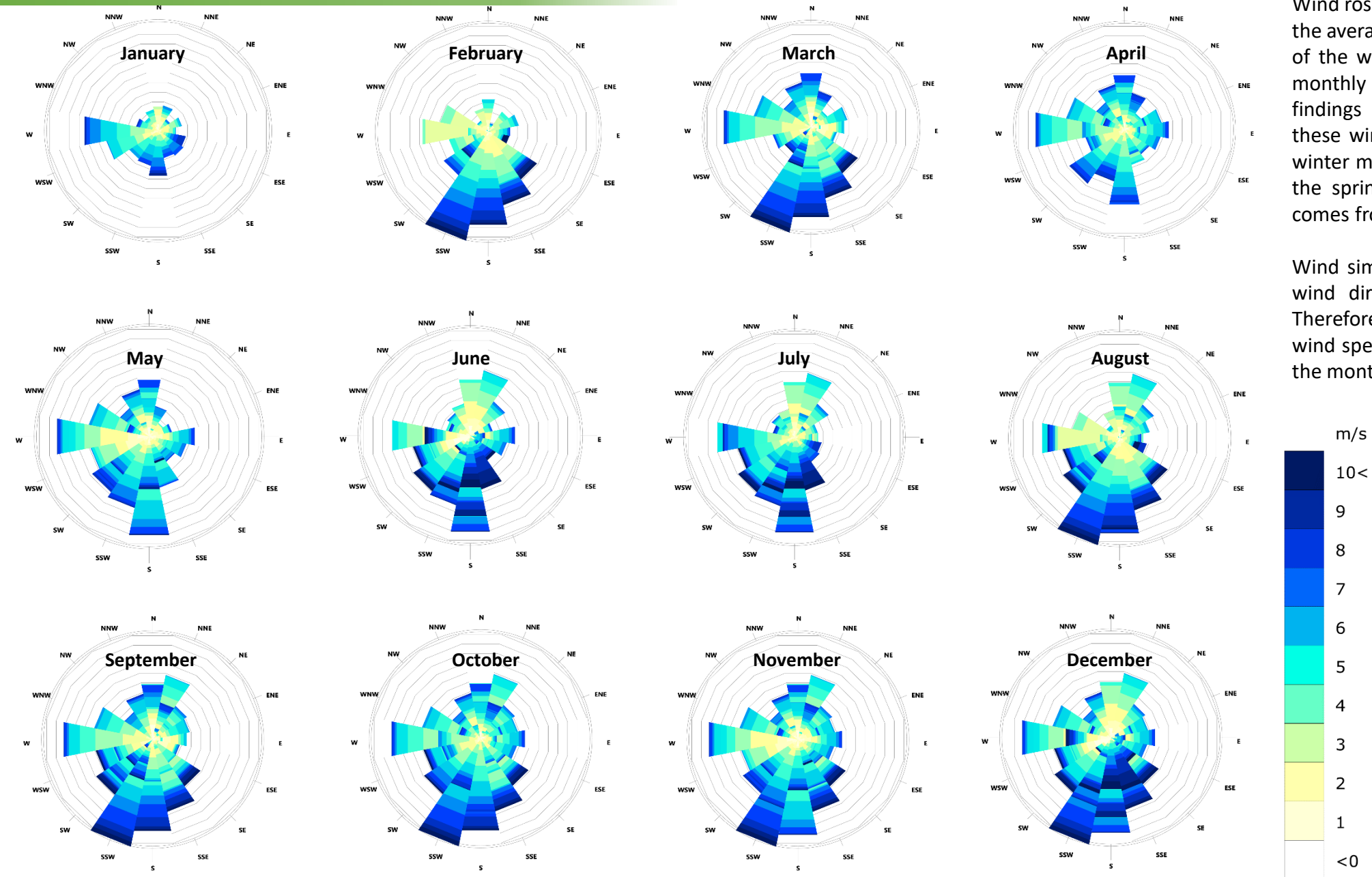


Figure 51. Monthly wind roses from Helsinki on the basis of 90's observations



## Implementation of the Wind Simulation

The Kalasatama wind simulations were made using the CityGML city information model (produced as the first intermediate objective of the project) and importing it into [ANSYS Discovery Live](#), which is capable of performing accurate calculations of air flow behavior with various objects. The speed of the wind was selected to be 15 m/s and the direction was selected from the south as the southern wind blows into Kalasatama from the sea almost without any obstacles. The following images 52 – 61 illustrate the behavior of the air flows in the Kalasatama area.

**Observations:** The data used in the simulation must be accurate and intact so that there are no gaps or defective objects. Any gaps in the data should be filled before the simulations are completed. By simplifying the data, the simulation can be optimized and LoD2 has a sufficiently high degree of accuracy. Wind simulations can also be done without a real terrain model, so there is no need to bring the terrain into the application to carry out the calculation. It should also be noted, however, that the buildings must be caught in an object on the terrain so that the air flow cannot pass under the buildings. For the wind simulation, the desired part of the city model must be selected with a sufficient surrounding space for the wind to blow in.

From the application, the results of the simulation test were immediately displayed on the computer screen, illustrating the behavior of the air flows in the Kalasatama area. For example, the simulator was able to investigate how the wind behaved around the buildings and their roofs and courtyards, as well as to look at what kind of wind occurred between the buildings on the street level.

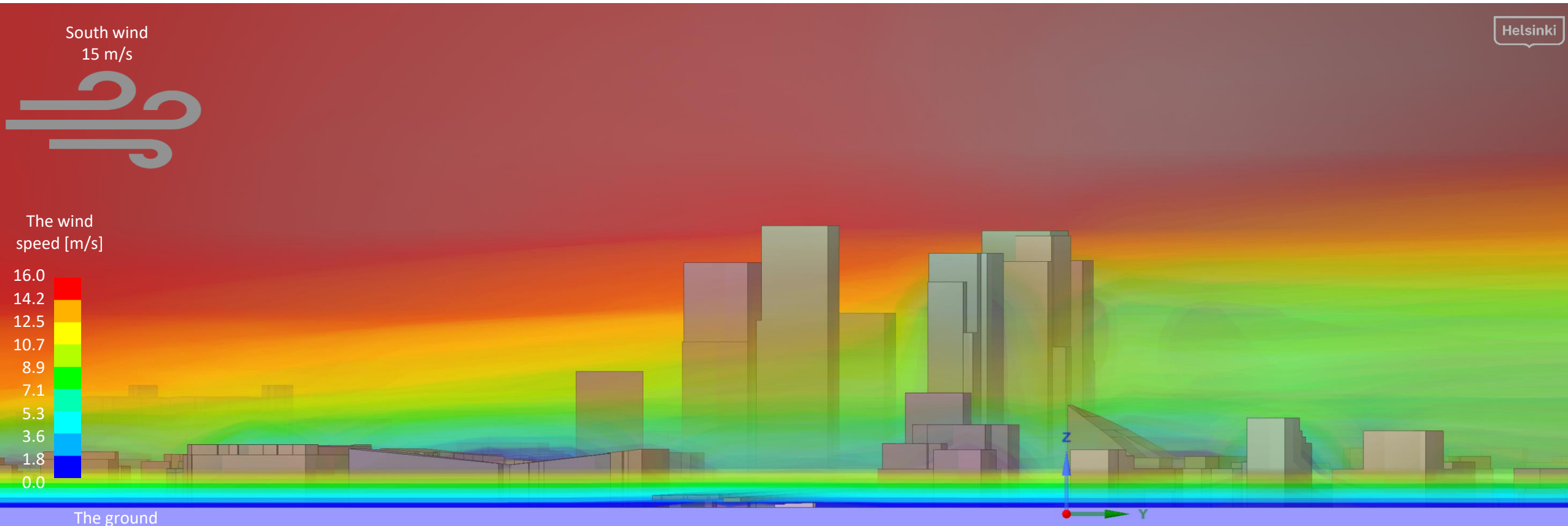


Figure 52. View of Kalasatama from the east. A south wind blows 15 meters per second

Figure 53 shows a sectional view of the wind intensity at street level. The colder the hue, the weaker the wind in the area. From the image is easy to detect places that are protected from the wind, and the areas where the wind blows the most. In the southern part of Kalasatama centre, the street level is clearly windier than in the northern area.

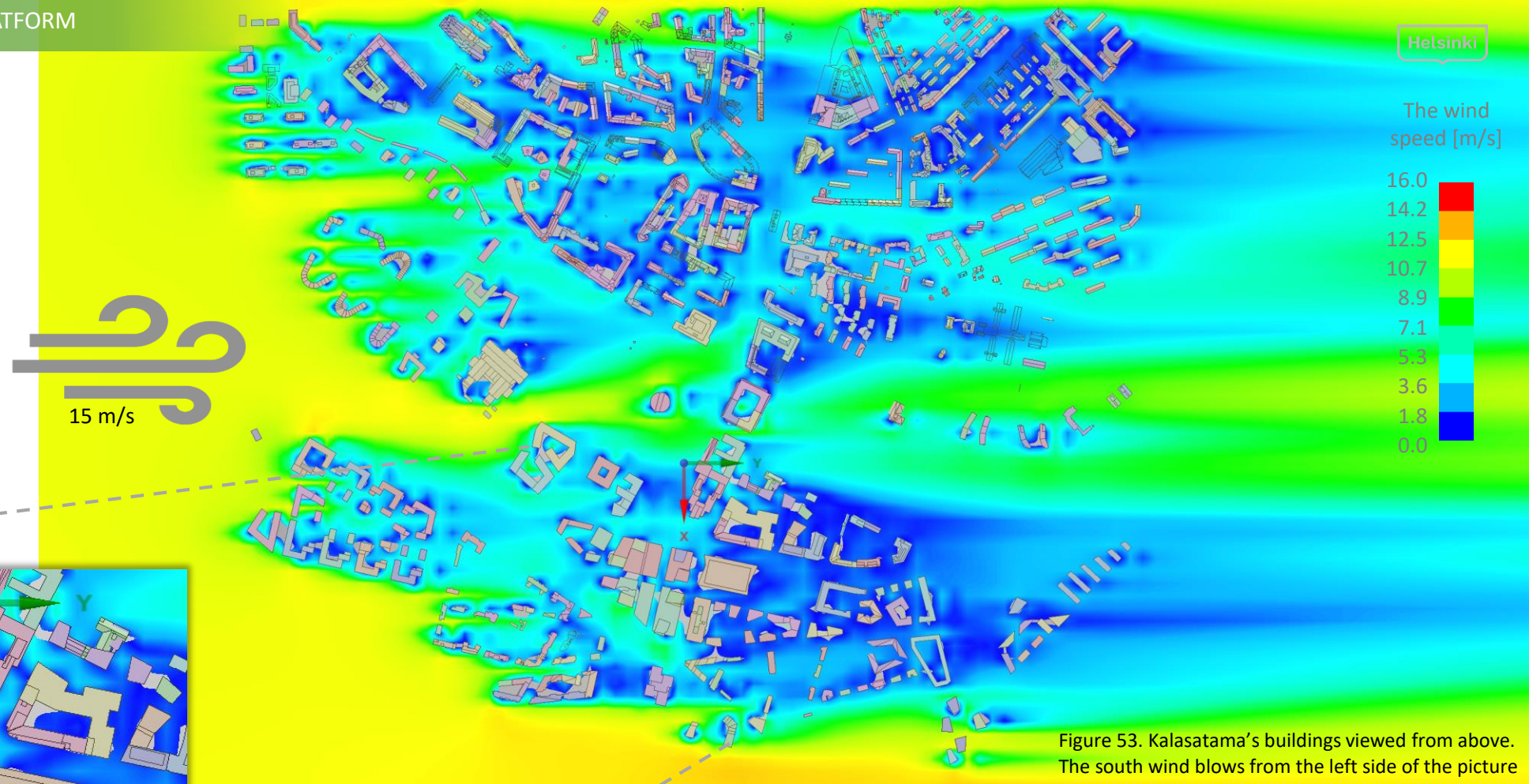


Figure 53. Kalasatama's buildings viewed from above. The south wind blows from the left side of the picture

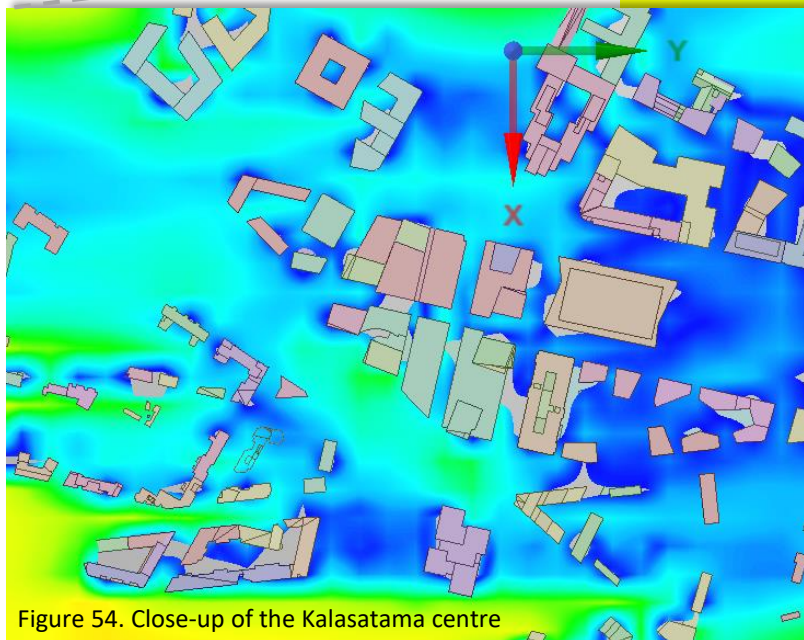


Figure 54. Close-up of the Kalasatama centre

A wind speed of 15 m/s was found to be useful for the simulations because it matches the Finnish Meteorological Institute's definition of a hard wind (14-20 m/s), but is not close to a storm wind (21-32 m/s). From the monthly wind roses in Helsinki, it can be noted that the winds from the sea during windy months easily go up to about 10 meters per second. Therefore during the day when the sea breeze tends to be stronger, it is easy to have continuous 15 m/s winds without being considered a strong wind.

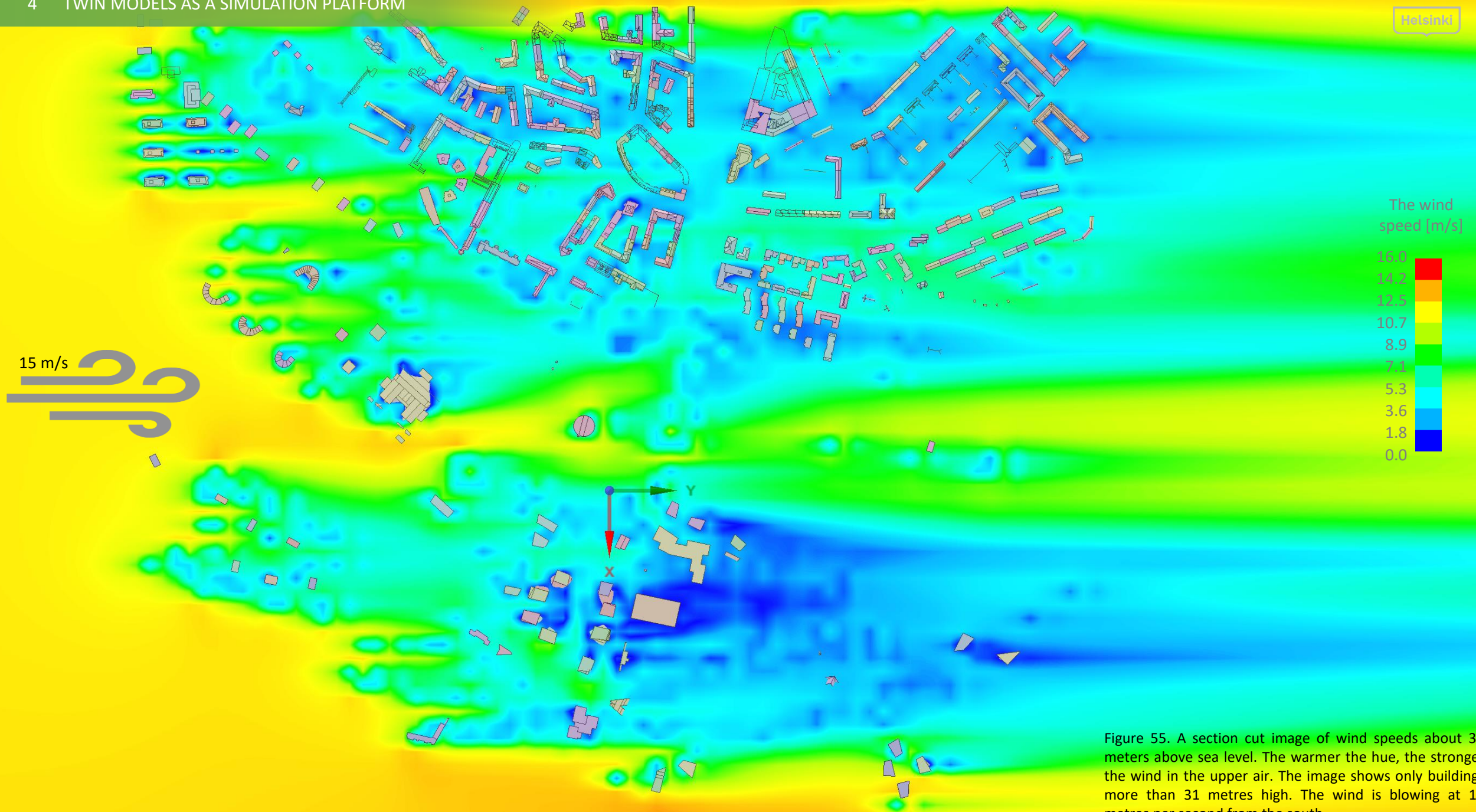


Figure 55. A section cut image of wind speeds about 31 meters above sea level. The warmer the hue, the stronger the wind in the upper air. The image shows only buildings more than 31 metres high. The wind is blowing at 15 metres per second from the south

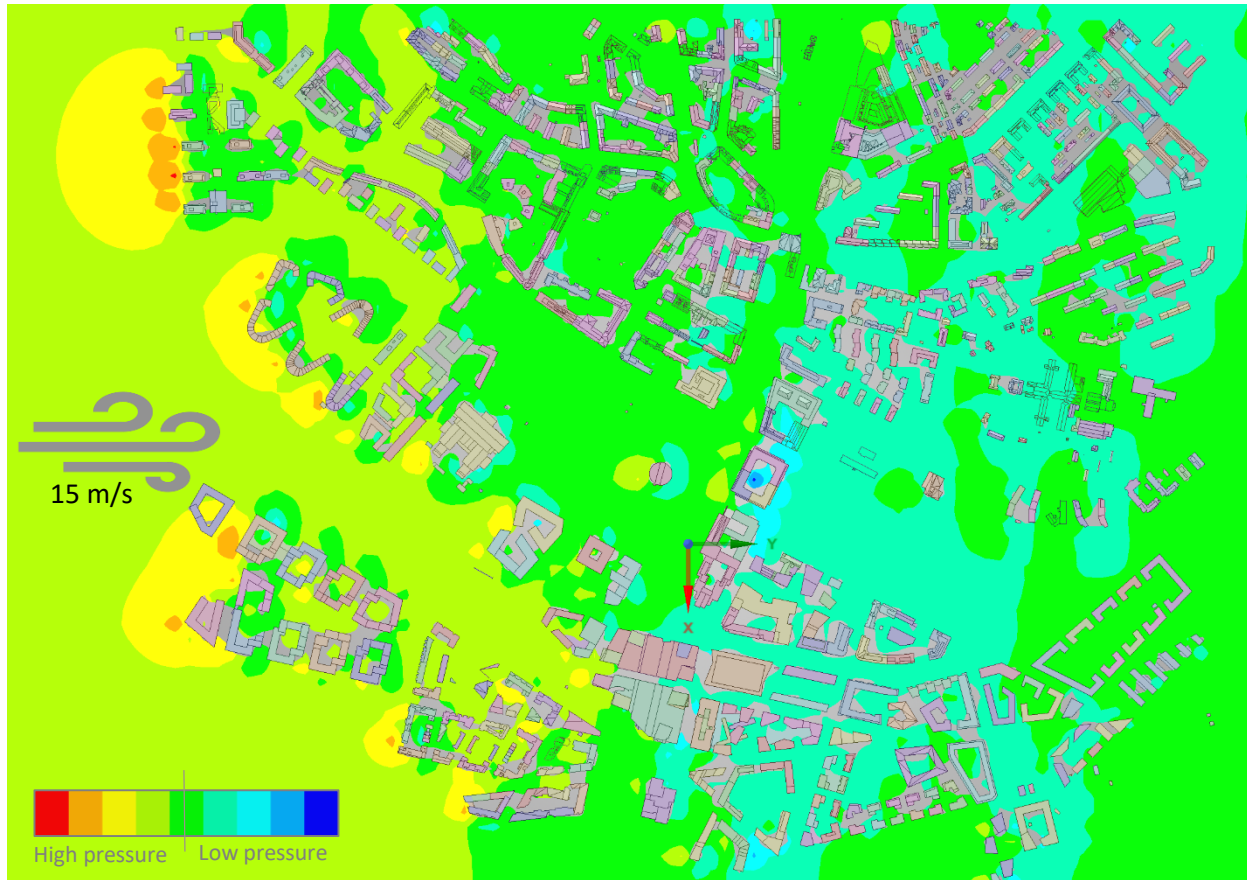


Figure 56. Kalasatama's buildings from above. This section cut image represents the air pressure distribution at street level caused by the wind from the south (from left side of the picture). The warmer the hue, the higher the pressure in the area

In addition to the wind speed, wind simulations can be used to study the air pressure caused by air flows. Large air pressure differences are not desirable because both high and low pressure have a negative impact on the condition of the buildings. By examining the air pressures, it is also possible to avoid problems that cause snow accumulation. Figures 56 and 57 show that in Kalasatama's simulation area, overpressure naturally forms to the south of the blocks close to the sea.

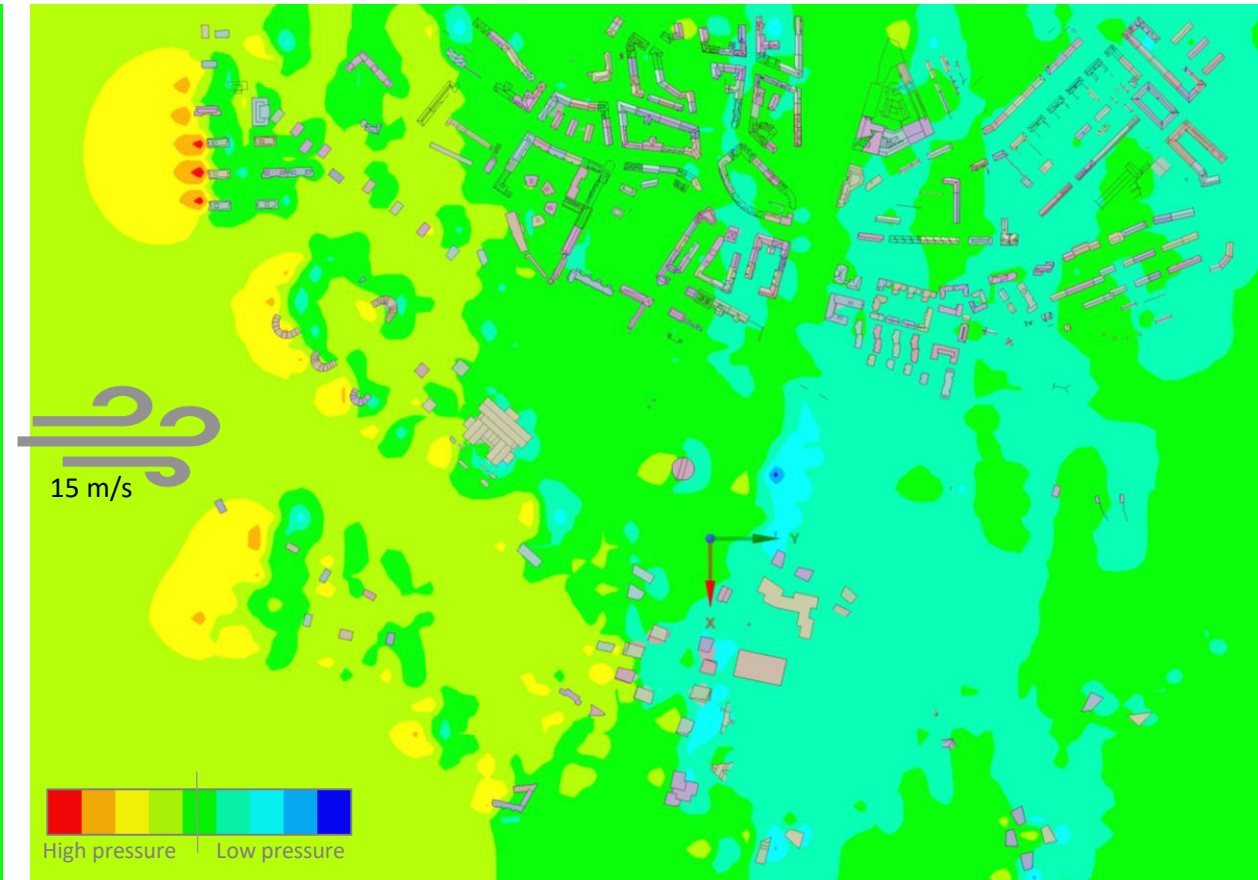


Figure 57. Kalasatama's buildings from above. This section cut image represents the air pressure distribution at 31 meters above the sea level caused by the wind from the south (from left side of the picture). A cold hue shows that a low pressure is directed towards the buildings

The low pressure is made up of individual courtyards and the north of the buildings. However, there are no significant differences in pressure generation as the viewing height changes. Examining air pressure without this kind of analysis is very difficult. The digital twin creates an opportunity to look at different weather conditions in the planning phase of the area.

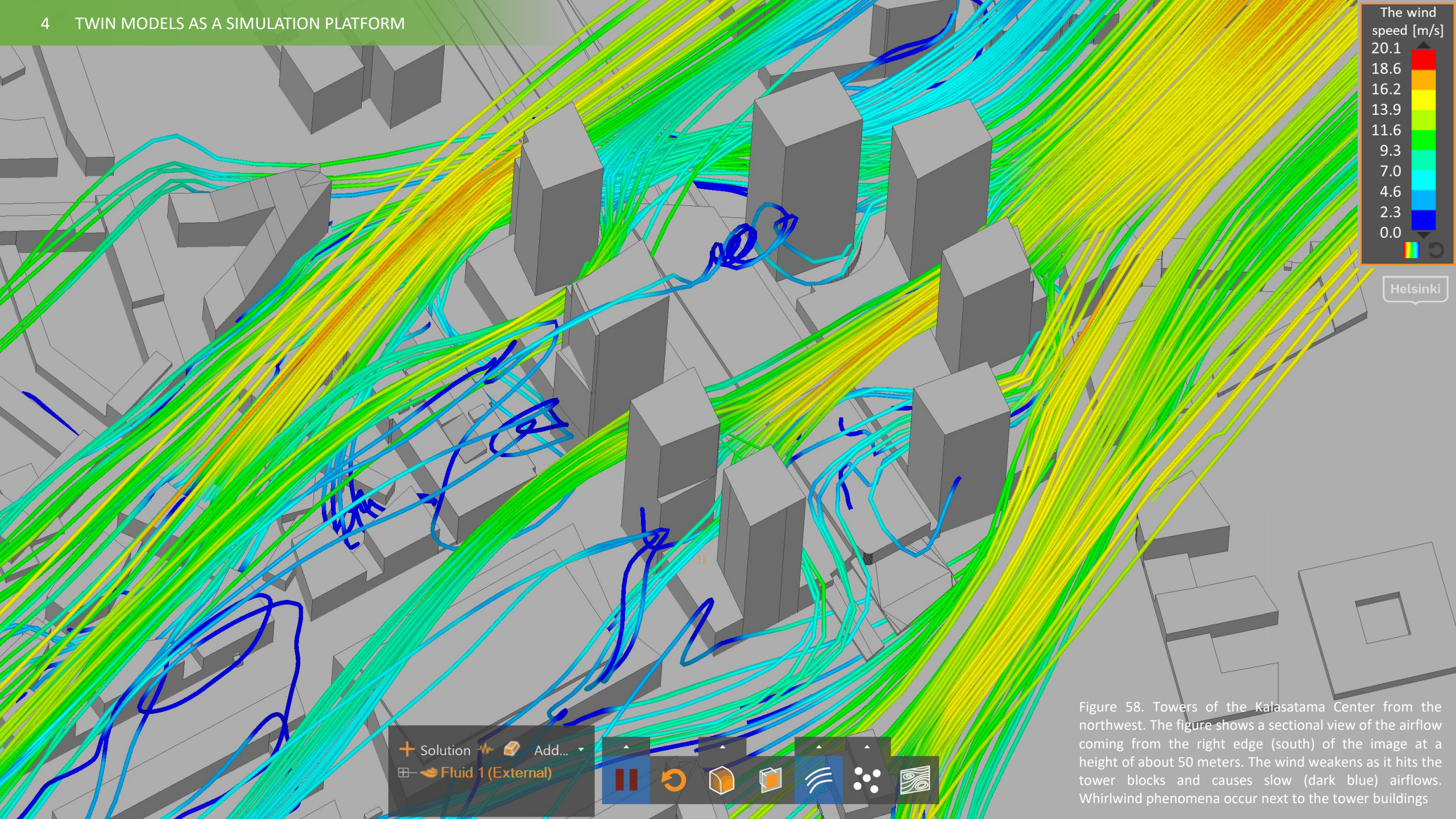


Figure 58. Towers of the Kalasatama Center from the northwest. The figure shows a sectional view of the airflow coming from the right edge (south) of the image at a height of about 50 meters. The wind weakens as it hits the tower blocks and causes slow (dark blue) airflows. Whirlwind phenomena occur next to the tower buildings

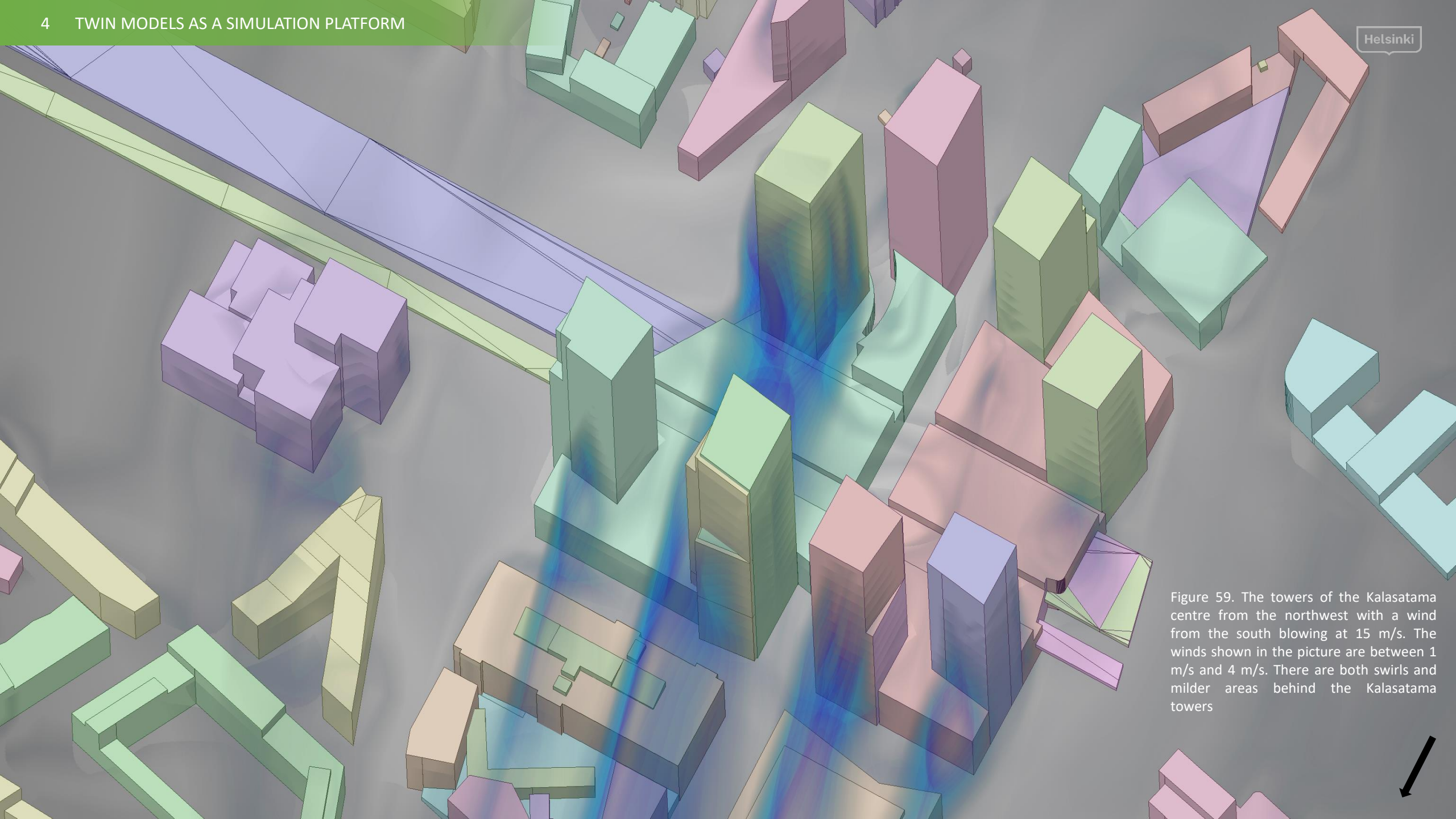


Figure 59. The towers of the Kalasatama centre from the northwest with a wind from the south blowing at 15 m/s. The winds shown in the picture are between 1 m/s and 4 m/s. There are both swirls and milder areas behind the Kalasatama towers

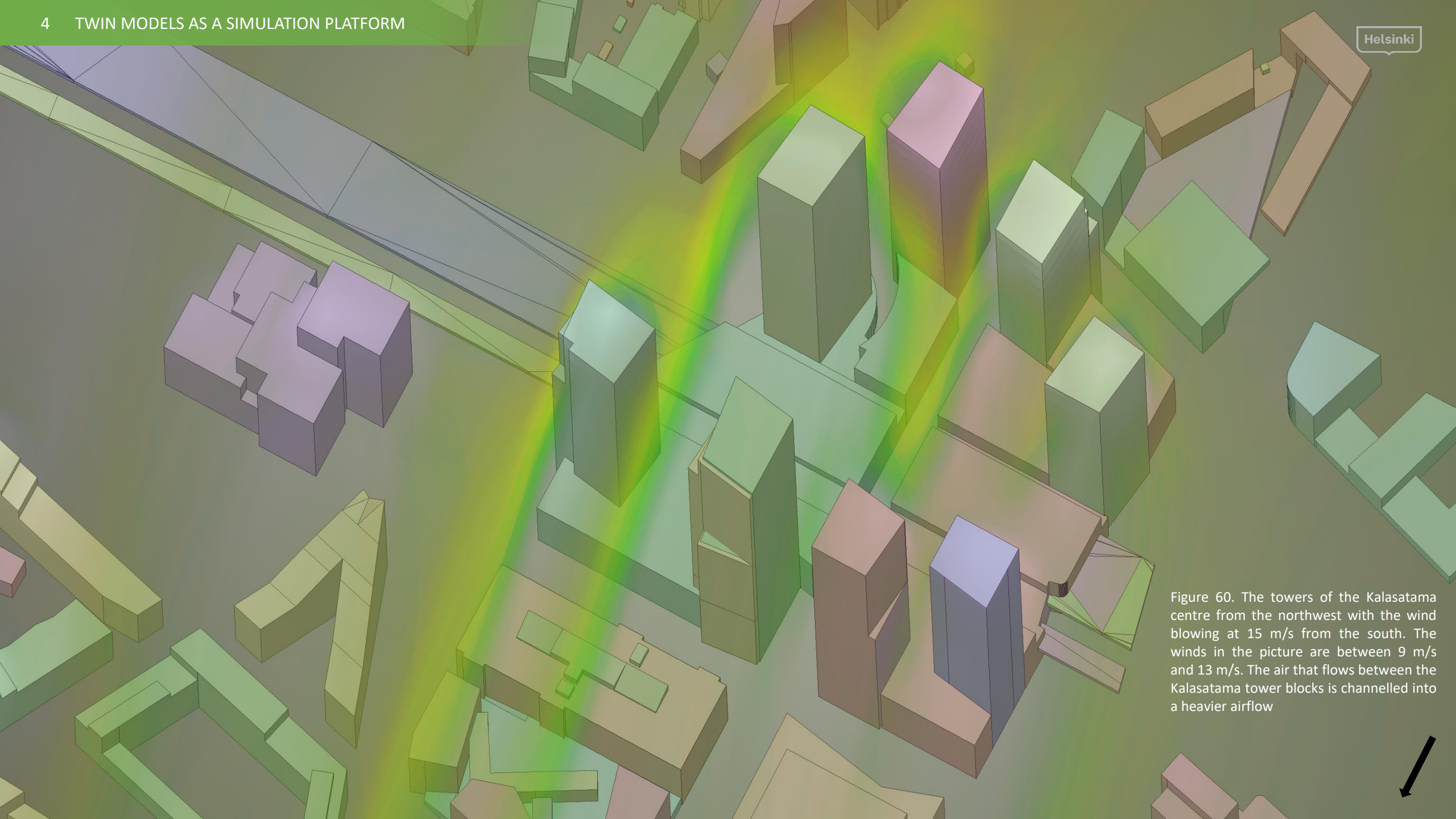


Figure 60. The towers of the Kalasatama centre from the northwest with the wind blowing at 15 m/s from the south. The winds in the picture are between 9 m/s and 13 m/s. The air that flows between the Kalasatama tower blocks is channelled into a heavier airflow

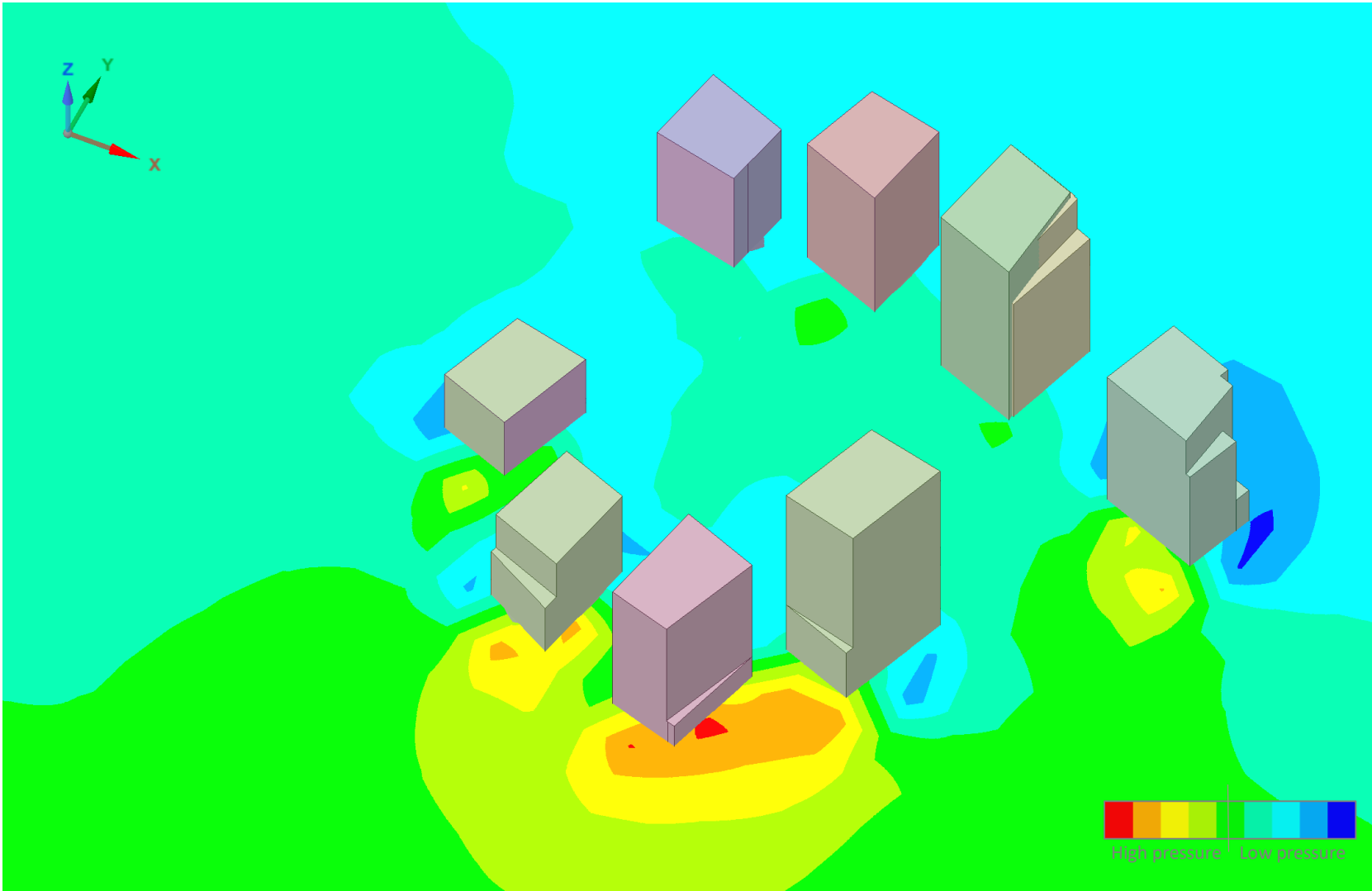


Figure 61. The high-rise buildings of Kalasatama from the south-east. A southern wind is blowing at 15 m/s. The section cut image presents the air pressure distribution at 75 meters above sea level

The simulations were calculated using the ANSYS Discovery Live application, which has been used in product development, such as the German automotive industry, for designing cars and exploring air flows. As a result of this pilot project, the application was tested for the very first time for modelling and analyzing airflow behavior using a city information model. The application produced very accurate results of both large and small areas. It would also be possible to model the wind and air pressure conditions, for example, on a single street.

A video can demonstrate the wind simulations and behavior of the wind flows in Kalasatama more comprehensively. The link below shows the ANSYS Discover Live wind simulations.

Urban air flow video:

[https://youtu.be/Zynv4a\\_TIIU](https://youtu.be/Zynv4a_TIIU)

**Observations:** The wind simulation experiment showed how city information models could be used to support city planning work. Digital twin models at their best realize this idea of planning, testing and building things digitally first.

The simulation application was able to produce a variety of results, such as numerical values in graphs, section cut surfaces showing wind speeds and pressures and 3D flow animations. The simulator used for wind modelling produced robust real-time observations of the behavior of the wind flows, as illustrated in this and previous pages of the report.



## Sun Study: Solar Hour Analysis

The aim of conducting solar simulations was to try out new ways to observe the behavior of sunlight in a built environment. In this project, two simulations were carried out: a solar hour analysis and a shadow analysis. The solar hour analysis examined how many hours the sun shines for in the Kalasatama area. The shadow analysis, on the other hand, examines what kinds of shadows the built objects cause. The analyses apply the basic idea of digital twins: to design, test and build things digitally first.

In a northern country such as Finland, the amount of sunshine is of great importance both as a factor affecting the quality of life for people, and also the potential for using solar radiation for energy production. Kalasatama is located at a latitude of 60° north, so the angle of incidence of solar radiation varies greatly from season to season.

The number of solar hours was studied using the CityGML city information model of Kalasatama. A solar hour refers to the time when the ground or the courtyards of the buildings receive sunlight. The solar hour analysis and related images were carried out using the Rhino application and its Grasshopper tool. The sun was set on the route according to the latitude as shown in Figure 62.

The analyses only take into account the solar radiation on the ground or in the courtyards. However, a park built on the roof (on the fifth floor) of the REDI shopping center is an important recreational area in Kalasatama. For this reason, REDI's building was removed from the model so that it was possible to observe the solar hours on that location too which can be seen in Figure 64. Images of the analyses are on the next two pages and a video of the analyses can be found at the following link: <https://youtu.be/3AhcdA900dQ>.

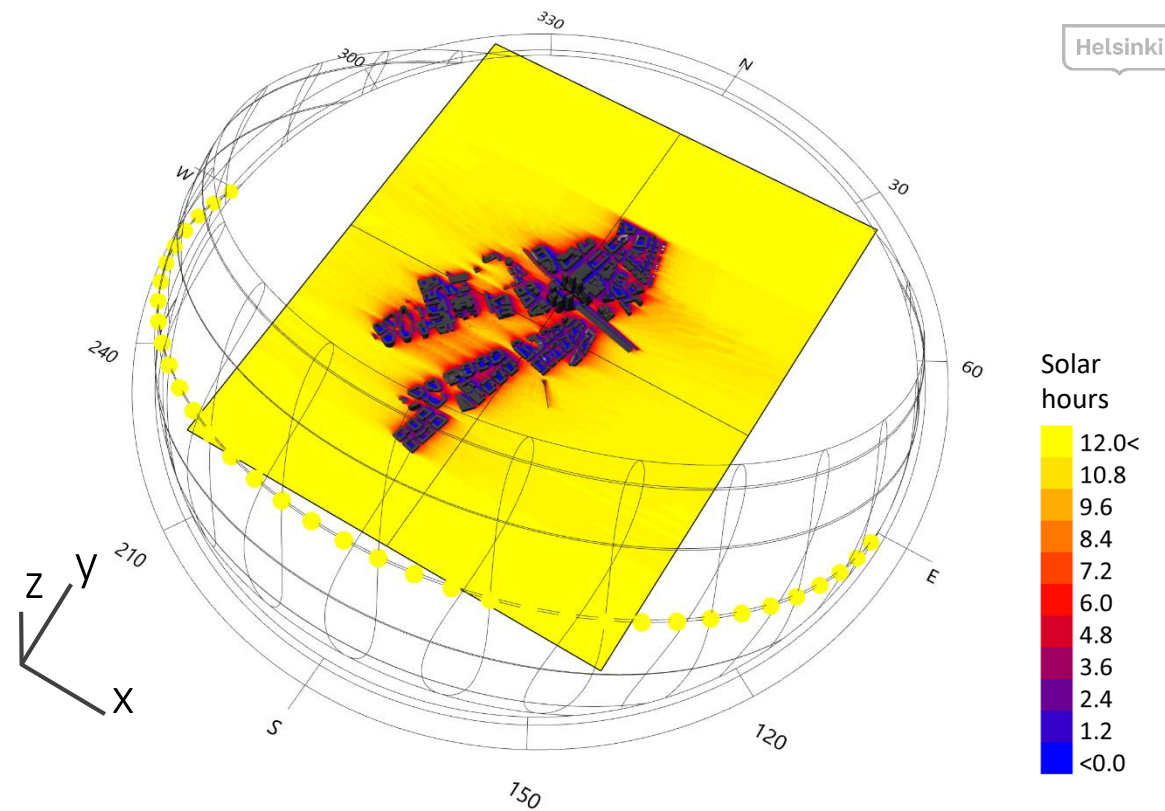


Figure 62. The results of a solar hour analysis for the 21st of March.  
A video of the solar hour analysis can be viewed by clicking the picture

**Observations:** The variation in the angle of incidence of sunlight and solar radiation over the season is wide. In the summer, there are more hours of sunshine and less in the winter, which can already be taken into account in the planning stage using this kind of sunshine analysis. As can be seen from the analyses, the shadows of the buildings extend far in the winter months when the sun shines at a very low angle. The shadows of just a few planned buildings should not be observed alone; the surrounding area already constructed should be included in the analysis as much as possible.

With the help of a high-quality city information model, the implementation of the analysis does not require the complete rebuilding of the models. Instead only the necessary part of the city information model can be added, buildings under design can be added and/or demolished buildings can be removed.

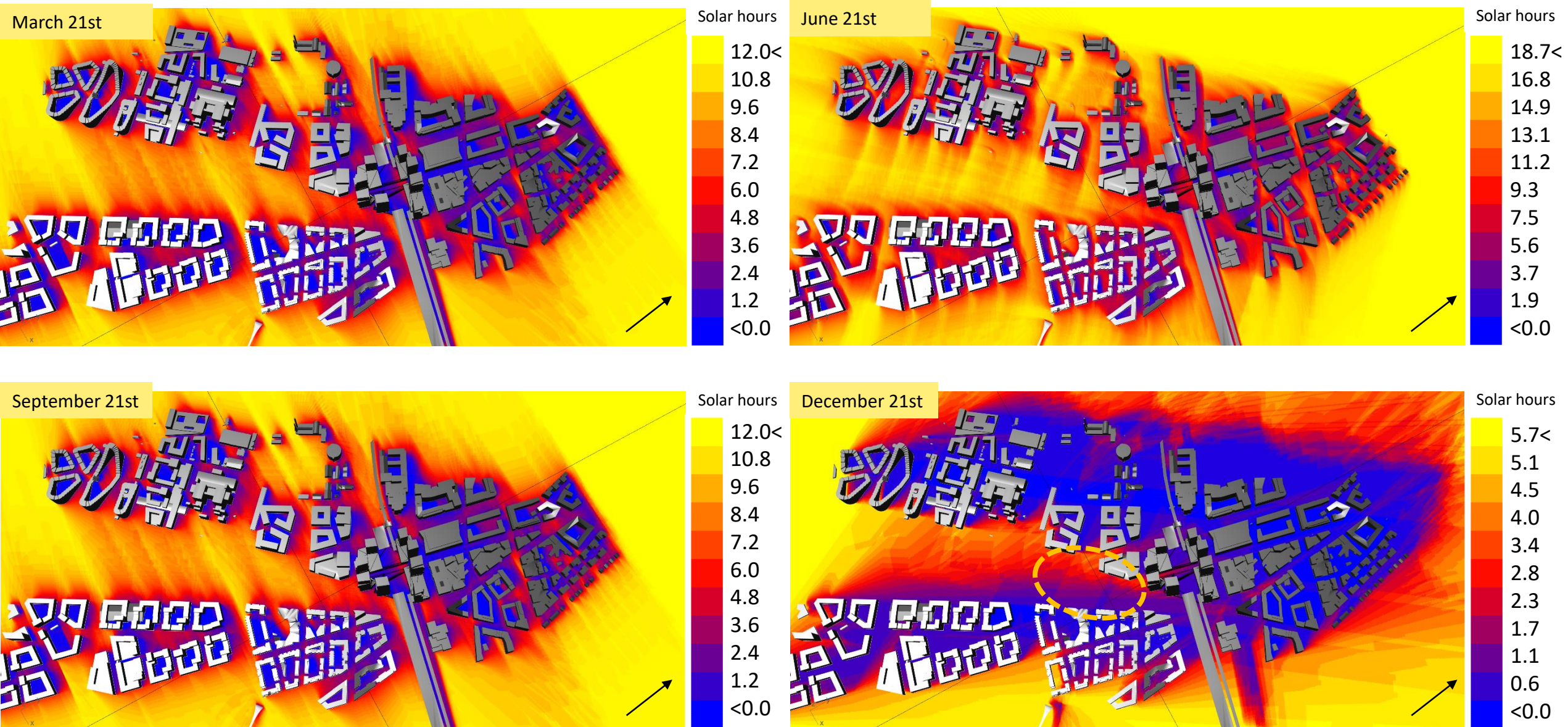


Figure 63. Sunshine analyses of the Kalasatama downtown area in March, June, September and December. A slight amount of direct sunlight can be noticed in the courtyards. The circled area in December's analysis is the park in Kalasatama. It can be seen from the analysis that the park is not completely shady even at the darkest time of the year

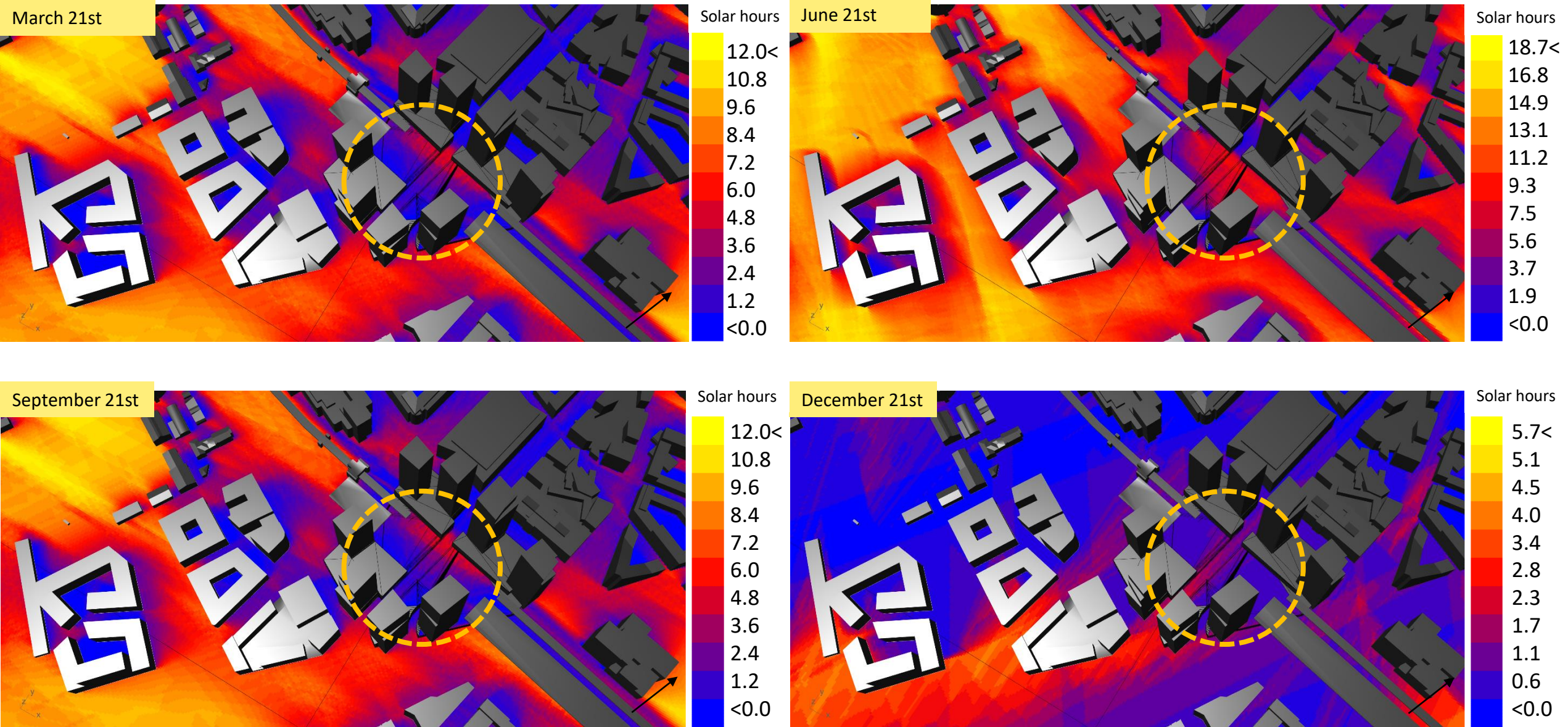


Figure 64. Solar hour analyses from central Kalasatama at a closer distance. The solar hours received by the REDI area in focus. As mentioned before, the REDI building has been removed from the model to enable the calculation of solar hours in the area

## Sun Study: Shadow Analysis

The aim of this analysis was to observe the shadows created by the proposed planned buildings in the center of Kalasatama's in different seasons.

The OpenCities Planner was chosen as the tool to observe the shadows created by the planned buildings on selected days and times. In the following figures, shadow analyses are performed for the same area every three months, starting with the year's longest month, i.e., June 21. The analyses were conducted for three different times: 10 am, 1 pm and 4 pm.

The shadow analyses can be done with the OpenCities Planner, if the source data is intact. The application only produces visual data, but its strength is its ease of review. With a few clicks of the application, the selected planned buildings can be removed or a simple new building design can be created and added to the analysis. This method is a very easy way to implement shadow analysis at all stages of the design.

June 21st at 10:00



June 21st at 13:00



June 21st at 16:00



Figure 65. A shadow analysis of Kalasatama centre on the 21st of June

September 21st at 10:00



September 21st at 13:00



September 21st at 16:00



Figure 66. A shadow analysis of Kalasatama centre on the 21st of September

On the basis of the solar and shadow analysis, the most significant dates of the year are the longest day of the year in the northern hemisphere, i.e., June 21st and the darkest day of the year, i.e., December 21st. If, in the brightest day of the year, an object receives a zero hour value in a solar hour analysis, it is unnecessary to look at the same area at other times.

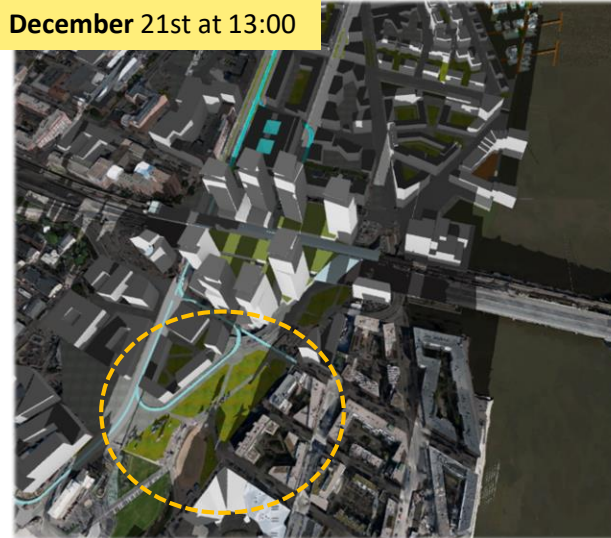
Based on the analysis, it can be seen that the Kalasatama Park (the yellow circle) receives light even at the darkest time of the year, at noon, which means that the park receives sunlight every day of the year (at least to a degree).

**Observations:** A high quality city model and a suitable application make it easy to determine the realistic behavior of shadows. Exploring shadows with the city model is useful and cost-effective, especially when studying the impact of planned buildings on existing buildings, for example when the plans for increasing the density of urban areas are in process.

December 21st at 10:00



December 21st at 13:00



December 21st at 16:00



Figure 67. A shadow analysis of Kalasatama centre on the 21st of December

March 21st at 10:00



March 21st at 13:00



March 21st at 16:00



Figure 68. A shadow analysis of Kalasatama centre on the 21st of March

## The Use of Digital Twins in City Processes and Service Production – Helsinki's Digitalisation Programme

This project has inspired active discussion on 'digital twins' as a term and method on the City of Helsinki's internal discussion forums.

The draft for Helsinki's digitalisation programme, released on 14 February 2019, presents digital twins as one of the programme's key projects. The programme suggests the following objectives:

*The City adopts an operating approach based on the utilisation of an information model-based digital twin, which enables the joint participation of all operators and the building of a good life for the city's residents.*

*The digital twin approach comprises a data resource, a service platform, a toolkit and the competence to develop the operation. It is an entity consisting of the real world and the virtual world that enables the planning, testing and implementation of processes, services and projects – digitally at first. The digital twin is a new competence that enables sophisticated analyses and simulations to produce optimal decisions.*

*Decision-making, interaction and communication will be based on well-founded facts and a reliable process, taking into consideration all perspectives and building mutual understanding. The information model enables the optimisation of the city's biological, social and technical*

*components as a whole. The use of a digital twin enables the exploitation of many and varied data sources. It is not a singular information model.*

*To promote all the above objectives: All matters processed by the City's Urban Environment Committee will be presented on the 3D city model platform. The matters in process will form a snapshot of the change taking place. This can be used in communications, interactions and in future project stages. The project is carried out in stages, and the final objective is to achieve operations that are holistically based on information models. This requires new kinds of expertise, adequate resources and a roadmap to guide changes.*

### USE CASES OF 3D CITY MODELS

#### UTILIZATION / USE CASES

SMART CITY,  
[SMART] TRAFFIC & TRANSPORTATION,  
[TELE] NETWORKS & LIGHTNING

CARBON NEUTRAL CITY,  
MICROCLIMATE

CITY PLANNING,  
[PROJECT] MANAGEMENT, BUILDING &  
INFRA DESIGN, CONSTRUCTION

SERVICES & WORKFLOWS,  
BUSINESS, INNOVATIONS & PRODUCTS,  
EDUCATION & RESEARCH

ASSET MANAGEMENT,  
EXCEPTIONAL SITUATIONS, RESCUE &  
SAFETY, MAINTENANCE

PROJECT PLANNING,  
BUILDING PERMISSIONS, DECISION  
MAKING, WEB SERVICES

COMMUNICATION & INTERACTION,  
[CITY] MARKETING, TOURISM &  
NAVIGATION, CITY EVENTS, EXHIBITIONS

UTILIZATION RATE  
(1Q2017)

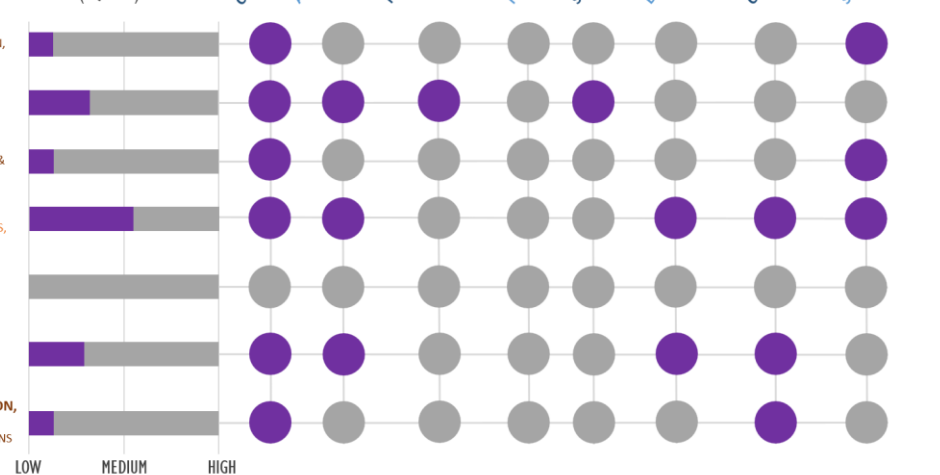


Figure 69. There are many different uses for 3D city models. City of Helsinki had tested already some uses (violet) by the beginning of 2017 but there are still much more left to try

**Observations:** The inclusion of the digital twin operating method in Helsinki's digitalisation programme is a major strategic policy affecting the modernisation of operating methods. The definition of the method as an entity consisting of data resources, a service platform and competence helps people understand the complexity of the objective. The use of a city model platform to support decision making is a realistic start for gradual, long-term development.

## From the BIM Model to a CityGML Model – Overview of Research and Development

In this part of the project, the focus is on exploring what possible ways exist for conversions between Building Information Models (BIM) and CityGML city information models. Next, in the form of a review, projects, studies and tools for transformation work are presented.

### Summary of the National University of Singapore's Research and Development Work

In 2018, the Singapore government, together with the National University of Singapore, completed a project titled *Achieving Complete and Near-Lossless Conversion from IFC to CityGML*, which aimed to promote the creation of a 3D city information model of Singapore. This goal was promoted by researching and developing methods and algorithms to automate the conversion of IFC building information models (BIM) into a CityGML information model, while retaining both the geometric and semantic information contained in BIM data.

The project's work plan included the development of the CityGML model with an application domain extension (ADE), which allows the storage of IFC data relevant to its spatial context, which is not currently supported by the CityGML standard format. In addition, the suitability of the TGG (Triple Graph Grammar) method for the conversion between IFC and CityGML models, for both the semantics and geometry, was tested.

**Observations:** The project came to the conclusion that a completely lossless conversion between the formats could not be achieved. However, the ADE extension and TGG were found to be good alternatives to support data conversion from IFC to CityGML. Nonetheless, with regard to the TGG and ADE extension, more research and development is needed. Further research is also needed to refine the selected conversion methods for BIM models and practices as well as for different end uses. In addition, it is particularly important, as research progresses, to demonstrate the suitability for practical applications.

Rudi **Stouffs**, Helga **Tauscher** and Filip **Biljecki**

*Achieving Complete and Near-Lossless Conversion from IFC to CityGML* (2018)

Department of Architecture, School of Design and Environment, National University of Singapore

ISPRS Int. J. Geo-Inf. **2018**, 7, 355 <https://www.mdpi.com/2220-9964/7/9/355/pdf>

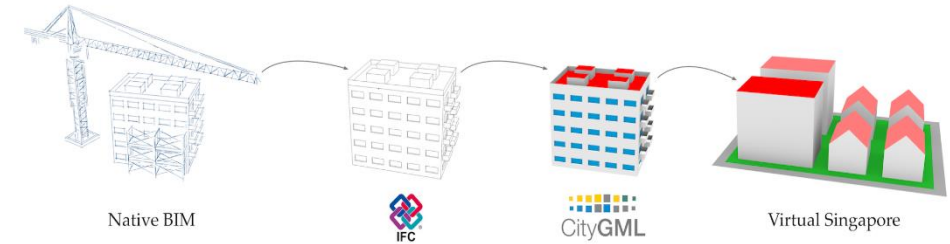


Figure 70. The project in a nutshell: the native BIM model, the CityGML model, and finally a component of Virtual Singapore

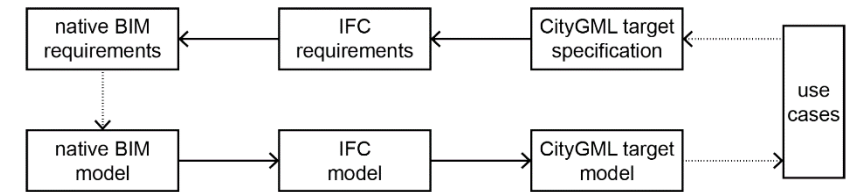


Figure 71. Project stages: from use cases to native BIM requirements, and from the native BIM model to the CityGML model and its uses

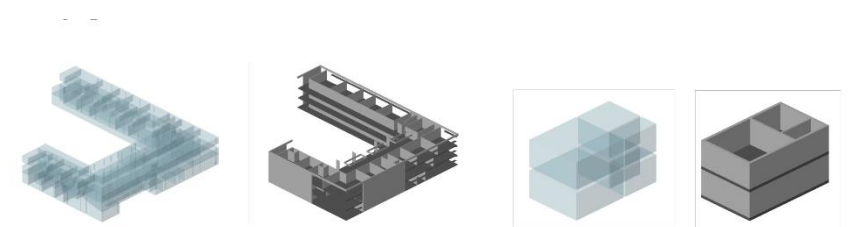


Figure 72. Conversion cases: an advanced experiment for a large office building (on the left) and a simpler case for a two-storey residential building (on the right). Conversions include rooms, walls, tiles and roofs. The roofs have been removed from the image for visual clarity

## Summary of Delft University's Research and Development Work

The GIS and BIM models overlap with city information modelling, but each has its own focus. The BIM data focuses on building and construction design aspects and construction information and therefore it includes a very rich semantic and detailed information model of the building's internal and external physical elements. GIS data, on the other hand, broadly describes environmental information at different times, making the data less detailed, but the possibility and the need for regular updating is essential.

The GIS and BIM data differ from each other in terms of their semantics, geometry and details. In addition, the modelling methods differ, which makes it difficult to achieve an optimal conversion between the models. It is assumed that combining these two data models would be quite useful and could possibly be a decisive step in the future 3D modelling of cities. This integration could avoid unnecessary efforts and intermediate steps, enabling new data flows towards new applications. Essentially, the BIM data could enrich existing GIS data, and in turn, the BIM data could obtain spatial and environment information from the GIS data that is currently absent from BIM data.

**Observations:** The GeoBIM project at Delft University of Technology explored the integration of GIS and BIM databases. Based on the results of the project, it can be concluded that the complete integration of GIS and BIM data is still far from a practical implementation. From a data point of view, this is mainly because the current BIM models contain many geometric and topological errors that need to be handled properly and often corrected before the conversion can succeed. In addition, it would be unrealistic to develop a separate transformation process for all data categories in IFC geometry. It would take years to cover all the categories of data, especially when many classes are very rarely used.

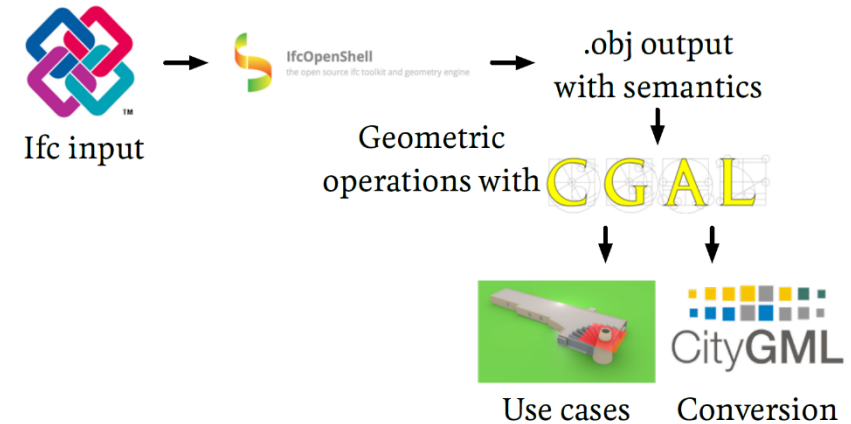


Figure 73. The final methodological approach of the GeoBIM project  
[GeoBIM Project Final Report]

In general, BIM data is much more detailed than the GIS data is expected to be. Therefore, it is highly unlikely that there will be a need to include all the details of the BIM database in the GIS database. Instead, the required benefits could be obtained by converting a generalized version of the BIM model (with GIS world features) into a GIS model. The 3D city information model could serve as an interface between the two databases. However, it is important that the unique identifiers are maintained at the interface so that the separated BIM models remain consistent with their generalized similar models.

Ken **Arroyo Ogori**, Thomas **Krijnen**, Abdoulaye **Diakit **, Hugo **Ledoux** and Jantien **Stoter**  
*GeoBIM Project Final Report* (2018)

TU Delft

The final report is available at: [https://3d.bk.tudelft.nl/ken/files/18\\_geobim.pdf](https://3d.bk.tudelft.nl/ken/files/18_geobim.pdf).



## The Simplebim<sup>®</sup> and CityGML Conversion Add-on (Beta)

The [Simplebim<sup>®</sup>](#) application is a tool to manage and process IFC data. The Simplebim<sup>®</sup> application is able to validate IFC files to better suit different uses. The application can, for example, disassemble or simplify IFC models or modify their properties.

The Simplebim<sup>®</sup> CityGML add-on (beta version) can be used to make simple conversions of detailed IFC models for use in CityGML models, resulting in, among other things, shell models. According to the product developer, the add-on is still under development and has some shortcomings, but the goal is to develop a tool for modelling clean and lightweight CityGML outer shell models.

According to the software developer, with CityGML extension (see add-on [documentation](#)), and from neatly modelled IFC models the software should be able to make a successful and highly optimized CityGML model. However, the reality is that the requirements are often not respected and the models are not always very clean. The requirements have been published by [BuildingSmart](#), but unfortunately, no one has yet taken the lead in carrying the development further.

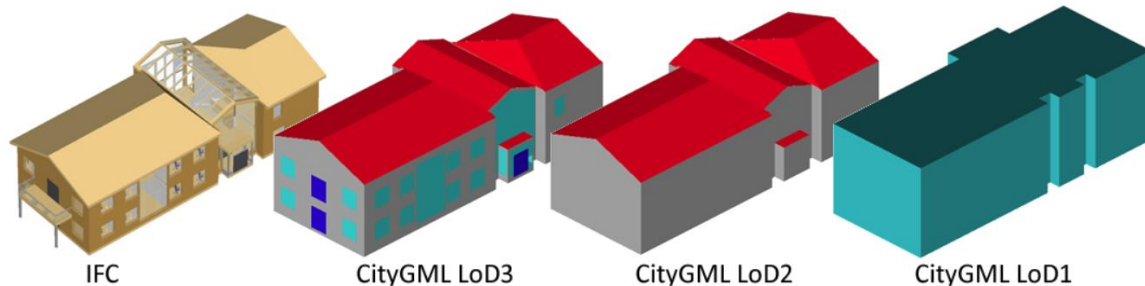


Figure 74. IFC model and CityGML models with different LoDs [<http://datacubist.com>]

With the Simplebim<sup>®</sup> application, models can be made to meet the requirements mentioned above for their content, including classifying objects as required by the CityGML standard or making a new layer division. However, it is not yet possible to do a great deal of geometry even though there are different geometry correction options in the extension. Correcting the geometry would be a considerable development effort that no developers have at present had the resources to do.

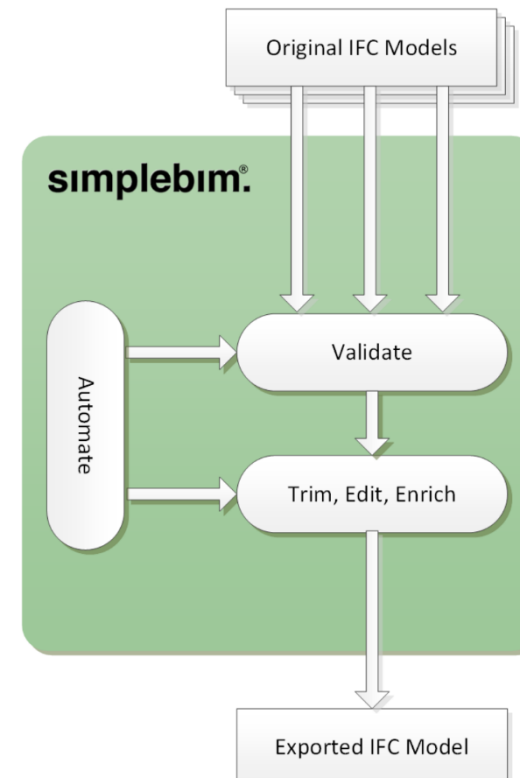


Figure 75. The principle of Simplebim<sup>®</sup> [<http://datacubist.com>]

BuildingSmart modelling instructions (in Finnish): <https://buildingsmart.fi/wp-content/uploads/2016/11/19.5.2016.IFC2CityGMLmallinnusohjeet.pdf>.

**Observations:** The software's exemplary data is the IFC model shown in Figure 74, from which CityGML test variants can be created without problems.

In the Kalasatama area, the IFC model of the K-campus building was available for quick experimentation. The building is very versatile with a ground floor pull-in and facade relief. Reducing the model with the program was effortless, but adding a missing floor slab from the IFC model to a building would have required a special software that was not available. Hence, the IFC conversion could not be implemented and the requirement for compliant output models presented in the text was found to be highly appropriate.

## Conversion Experiment from an IFC Model to a CityGML Model

Digital twins play an important role in both the construction and city modelling sectors. The digitalisation of the construction industry covers all stages of the life cycle of a building, from design to construction and throughout its use and up to the final dismantling phase. Building information modelling enables the use of models also in the city information model. The accurate geometry and attribute information contained in BIM models can be used to enrich city information models.

The [BIM inside ANSYS](#) application is an extension that is integrated into CADFEM ANSYS applications and is mainly used to prepare building models for simulations. The application enables semi-automatic generation of suitable models for simulations. The BIM inside ANSYS includes a graphical user interface for viewing IFC building models and converting them to the desired format. The application is able to visually (and through the model specifications) locate the correct building components for the final building model and/or for the accuracy of modelling. A virtualcityBIM application developed from the BIM inside ANSYS enables an IFC model to be converted to the CityGML format. In the framework of this project, it was attempted to simplify the building models in the BIM inside ANSYS because the project did not have a virtualcityBIM application available. However, the

building models are very complex and not simple to manage or handle so that a straightforward conversion of the IFC model into another information model format would be possible instantly or automatically. The BIM/IFC models are a very important part of the maintenance and updating of city information models, so within the framework of this project, one IFC model was commissioned to successfully convert as a CityGML model.

**Observations:** Applications are well suited for viewing the building information model and its individual parts, but converting a BIM model into a CityGML model requires manual processing. The more accurately it is desired to produce the model, the better the source data must be known and the more time it takes to convert the model.

The CityGML model generated from the IFC model for the city information model illustrated problems with positioning the model: in Figure 76, the red building was converted from the IFC model and its positioning in the x and y directions was excellent, while the height adjustment of the building was challenging. This red building seems to be a few meters higher than the corresponding construction which the CityGML model generated from the CAD model (see Figure 78).

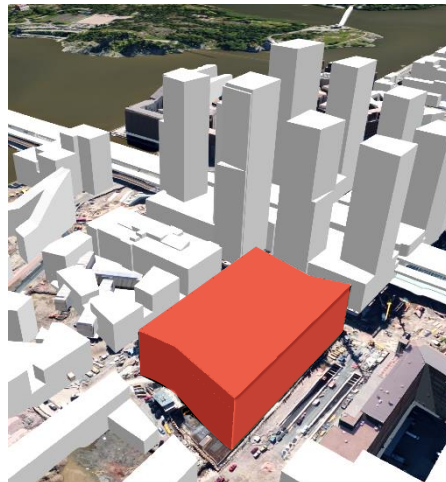


Figure 76. LoD1 CityGML building converted from IFC model



Figure 77. LoD2 CityGML building converted from CAD model

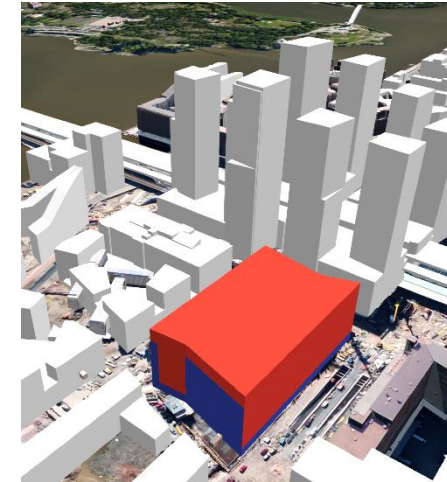


Figure 78. CityGML buildings converted from IFC model (red) and CAD model (blue)

## UAV Drone Experiment

City-wide aerial photography is not cost-effective to run every year, as most of the city remains mainly unchanged. However, emerging areas need to be brought into the city information model as quickly as possible, so new ways to upgrade the changing areas need to be tested.

In the pilot area of Kalasatama, a UAV (unmanned aerial vehicle) drone flight was conducted as a possible solution to the above problem. UAV scanning produces 10-200 points per square meter (traditional scanning has 0.5-40 points), and a UAV's terrain resolution is about 3 cm (traditional scanning resolution is 5-30 cm). Thus, this technique has the ability to produce more accurate source data. In this project, the suitability of this data for city information modelling and its functionality for maintenance in smaller areas, as well as for producing a mesh model, were tested.

The flight was carried out by GeoTrim Oy in the summer of 2018 on two days (July 4<sup>th</sup> and 11<sup>th</sup>). The drone used was a GeoDrone X4L quadcopter fitted with a YellowScan Surveyor laser scanner and a Sony RX1 camera (using the ETRS-GK25FIN/FIN2005N00 coordinate system). The flight altitude was 100 meters while capturing the aerial images and for the laser scanning it was 50-60 meters. The flights were carried out using crossed flight lines to get as many echoes of

**Observations:** A UAV flight is cost-effective for modelling small areas, and the laser scanned point cloud is even more accurate due to the cross-sectional lines compared to the traditional scanning from an airplane. Collecting data from the UAV and getting the data ready for further processing is relatively fast. The data collected was good enough to model high-quality buildings. Therefore, UAV laser scanning and photography were found to be well suited for maintaining the city information model and supporting existing modelling methods, especially as UAV technology is constantly evolving. However, the collected data for the production of the mesh model was insufficient. Running a flight with denser flight lines and better-targeted photography could provide better results.

the laser pulses from different vertical surfaces as possible. The result was a point cloud (with 86 million points) produced by laser scanning and a photogrammetric point cloud produced by photogrammetry (at 1.63 cm GSD, with 441 images, and 134 million points). The laser point cloud was classified using automated methods and the photogrammetric point cloud was successfully tested with manual classification for roofs, raised courtyards and cars.

Both point clouds were compared to the city-wide laser point cloud produced in the summer of 2017, and it was found that the laser point cloud contained errors at altitudes, while the photogrammetric point cloud was very accurate with the altitude values. However, the laser scanned data became useful once the point cloud was corrected by re-registering the photogrammetric point cloud as a reference.

It was decided to model the buildings with laser scanned data, but it was decided to bring the colours into the model from a photogrammetric point cloud created from aerial photographs. The data and a MicroStation application were used to create a surface model (DSM) and terrain model (DTM) grids, which were introduced into the BRec application. The processing of the data into buildings is done in the same way as it is in traditional aerial photography and laser scanning.

**Source:**  
City of Helsinki, City Survey Services, 2018.

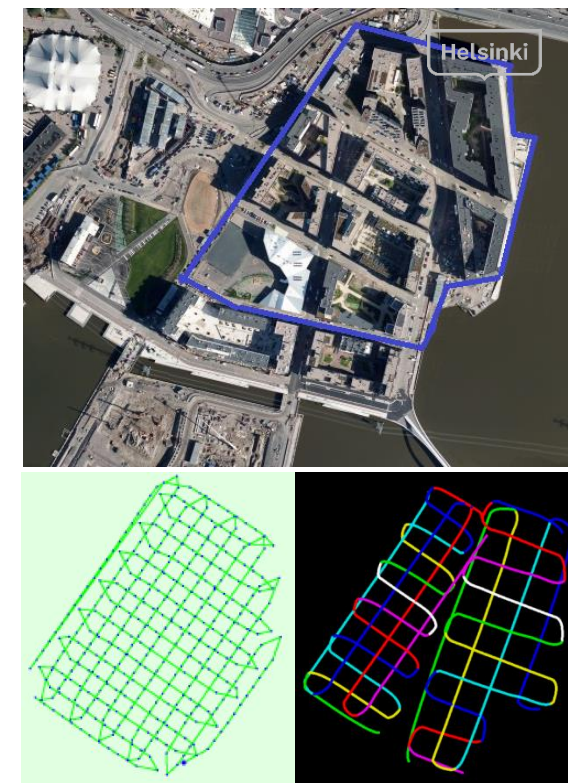


Figure 79. The pilot area (above), the crossing flight lines for UAV photography (bottom left) and laser scanning (bottom right)

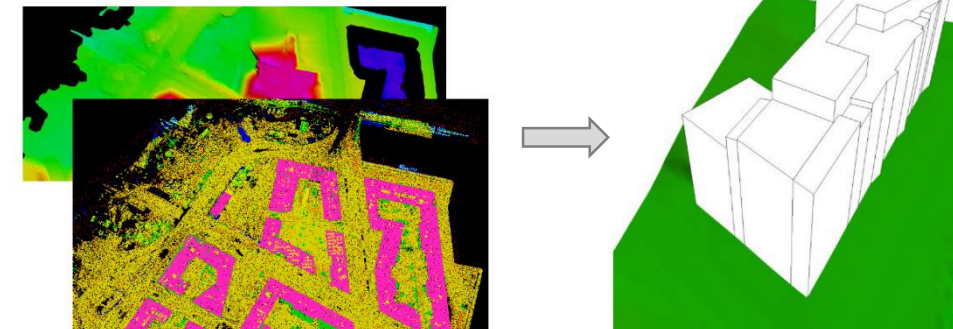


Figure 80. Buildings can be modelled with BRec application from the classified point clouds for the maintenance of city information model

# RESULTS: OBSERVED BENEFITS, EFFECTIVENESS AND DEVELOPMENT NEEDS

## Observed benefits

Synergy gains are achieved by carrying out the project in Kalasatama, where many other experimental projects are taking place at the same time.

The benefits from open data will accumulate over a longer period. We do not currently have exhaustive data on on-going projects that utilise open data.

Connecting the city model with the Smart Kalasatama urban development and related projects has already been useful at the initial stage.

Game engine streaming brings new developer groups for model data and creates new opportunities for its use.

The user experiences of and the results from the wind analysis tool are very positive, and the need for the tool has been recognised by many of the relevant parties already during the regional planning stage.

Wind simulation can also be used for predicting the spreading of harmful emissions and smoke gases from fires.

Point cloud data for small areas, collected by UAV flights, can be used for maintaining the city information model. This is a cost-effective method and allows rapid reaction to changes taking place in the city.

## Effectiveness

There has already been great interest in the on-going work from both Finnish and international operators: the linking of a city model with Smart urban development is new.

Because there is a demand for it, the report is translated into English: the challenges of urban development are global, and open cooperation justified.

The work that has been carried out had a direct impact on the content of the draft for Helsinki's digitalisation programme, and the digital twin is one of the programme's key projects.

The effectiveness of the project should be evaluated over the long term: practical reports on digital twin experiments carried out at a city model level have not been available so far.

The effects of open data and an open CityGML database dump are long-term. The use of the data in training, in particular, generates iterative benefits.

The expansion of the digital twin operating model to city level through practical cases could create a totally new operating culture.

In the best case scenario, all parties involved in the value chain of urban planning will consider and innovate new uses for a digital twin in their own operations.

## Development needs

The modelling technologies used in both the CityGML and reality mesh models would benefit from simpler applications and operating chains. Also, model maintenance technologies, standardisation and the seamless integration of the BIM and GIS environments require further international development for years to come.

At the national level, training on the CityGML city information model should be offered by universities as part of their degree programmes. It is not possible to develop internationally successful products or service concepts without solid basic knowledge.

The team hopes that this report will promote the overall digital development of urban planning and maintenance, which will produce cumulative benefits, starting now.

# SUMMARY OF LINKS

## Tools

3DCityDB <https://www.3dcitydb.org>  
ANSYS Discovery Live <https://www.ansys.com/products/3d-design/ansys-discovery-live>  
BIM inside ANSYS <https://www.CADfem.de/produkte/CADfem-ansys-extensions/bim-inside-ansys.html>  
BuildingReconstruction <https://www.virtualcitysystems.de/en/products/buildingreconstruction>  
ContextCapture <https://www.bentley.com/en/products/brands/contextcapture>  
FME <https://www.safe.com/>  
Grasshopper <https://www.grasshopper3d.com/>  
Ladybug Tools <https://www.ladybug.tools>  
MicroStation <https://www.bentley.com/en/products/brands/microstation>  
OpenCities Planner <https://cityplanneronline.com/site/>  
pgAdmin <https://www.pgadmin.org/>  
PostGIS <https://postgis.net/>  
PostgreSQL <https://www.postgresql.org/>  
Rhinoceros <https://www.rhino3d.com/>  
Simplebim <http://www.datacubist.com/>  
SketchUp <https://www.trimble.com/Buildings/SketchUp>  
TerraModeler <http://www.terrasolid.com/products/terramodelerpage.php>  
TerraScan <http://www.terrasolid.com/products/terrascanpage.php>  
virtualcitySUITE <https://www.virtualcitysystems.de/en/products>

## Open Data

Helsinki

Downloadable City Model Data:

[https://hri.fi/data/en\\_GB/dataset/helsingin-3d-kaupunkimalli](https://hri.fi/data/en_GB/dataset/helsingin-3d-kaupunkimalli)

City Information Model of Helsinki

<https://kartta.hel.fi/3d>

Viewable Reality Mesh Model of Kalasatama

<https://kartta.hel.fi/3d/mesh/kalasatama>

Umbrified Helsinki 3D Data

<https://info.umbra3d.com/request-access-to-umbrified-helsinki>

Platform for Smart City Development of Kalasatama

<https://www.cityplanneronline.com/helsinki/kalasatama>

Platform for Smart Kalasatama's Projects and Innovations

<https://www.cityplanneronline.com/helsinki/fiksukalasatama>

## Videos

Kalasatama's Reality Mesh Model:

[https://youtu.be/c4SoR2SME\\_4](https://youtu.be/c4SoR2SME_4)

Umbra Composit

<https://youtu.be/eQpXqcTiGQk>

Wind Flow Simulation

[https://youtu.be/Zynv4a\\_TIIU](https://youtu.be/Zynv4a_TIIU)

Solar Hour Analysis

<https://youtu.be/3AhcdA900dQ>

# THE PROJECT TEAM



**Enni Airaksinen**  
3D Specialist, Helsinki 3D+, City Executive Office, City of Helsinki



**Maija Bergström**  
Community Manager, Smart Kalasatama, Forum Virium



**Hannu Heinonen**  
IT Specialist, Urban Environment Division, City of Helsinki



**Kari Kaisla**  
Project Manager, Helsinki 3D+, City Executive Office, City of Helsinki



**Kristiina Lahti**  
Coordinator, Helsinki 3D+, City Executive Office, City of Helsinki



**Jarmo Suomisto**  
Project Manager, Helsinki 3D+, City Executive Office, City of Helsinki