

The Technology Stack and System Architecture of the University of Cologne Virtual Campus

Christian Willmes¹, Lorenzo Canals⁶, Christopher Ganser¹, Dirk Mennecke¹, Finn Luis Phillipps¹, Anne-Kathrin Pietsch², Tim Reichenau¹, Elisabeth Reuhl², Philip Schildkamp², Maximilian Stempel¹, Daniel Wickerroth⁴, Jan Wieners⁵, Øyvind Eide^{2,3}

¹Institute of Geography, University of Cologne, Germany - (cwillmes, cganser2, dmennec1, tim.reichenau)@uni-koeln.de

²Department for Digital Humanities, University of Cologne, Germany - (a.pietsch, philip.schildkamp, ereuhl2)@uni-koeln.de

³Center for Data and Simulation Science, University of Cologne, Germany - oeide@uni-koeln.de

⁴IT Center, University of Cologne, Germany - wickerroth@uni-koeln.de

⁵Center for Teacher Education, University of Cologne, Germany – jan.wieners@uni-koeln.de

⁶University Library, University of Cologne, Germany - canals@ub.uni-koeln.de

Keywords: 3D CityModel, Digital Twin, 3D Building Model, AR, VR, XR, GameEngine, indoor navigation, Modelling

Abstract

The here presented Virtual Campus project developed a 3D geodata modelling toolchain, and an open source software based technology stack for creating a Digital geoTwin of the University of Cologne (UoC). We also develop and present a reactjs based web application, a flutter and Unity based mobile app with AR features, as well as an UnrealEngine based VR application to access, browse and visualize the geoTwin data and models in the UoC CAVE 3D environment. The 3D models of the Universities Buildings will include indoor models based on 2D floorplans, which will facilitate indoor positioning in the mobile application of the project through geometry matching and simultaneous localization and mapping (SLAM) technology. For distribution and streaming the 3D building models, we utilize the OGC Standard 3D Tiles, for which we present a tool to georeference the glTF 3D models and assets accurately on the globe.

1. Introduction

In this contribution we present a primarily open source technology stack which is used to implement a 3D web GIS application, a Game Engine-based XR¹ mobile app, visualizations in an interactive and immersive virtual reality environment, and a 3D data modelling toolchain for a Digital Twin in which certain objects and processes at the University of Cologne are modelled,² including the Virtual Campus 3D models of university buildings.

1.1 Virtual Campus

The Virtual Campus project³ deals with the application of AR and VR in the context of the UoC. One of its aims is to create a georeferenced 3D model of the university. Our main objective is modelling a highly detailed 3D representation of the University Library (USB) building, which will be created, according to the CityGML 3.0 Standard (Kolbe et al., 2021) definition in Level of Detail (LoD) 3, as well as up to LoD3 for the indoor models and assets.

This detailed model of the USB and the UoC Campus will be used as basis to virtualize the services and offerings of the USB through augmented reality (AR) and virtual reality (VR) applications. AR primarily facilitates displaying additional location-based information about services and offers, while VR is employed for virtual tours and exhibitions. We are also 3D modelling other university buildings (excluding the university hos-

pital) in different Levels of Detail, which will be made available as a Digital Twin of selected aspects of the university campus and its infrastructure and will facilitate the scaffolding of further AR and VR applications. The 3D models are furthermore used by mobile devices to localize themselves within the interior of the modelled buildings, using a method based on Simultaneous Localization and Mapping (SLAM) for indoor positioning. In the Unity-based mobile app (see section 2.3), this is facilitated by the AR Foundation framework, which is a prerequisite for the correct display of AR content. This also provides the basis for indoor navigation in the first-person perspective, a functionality being implemented as a further development of the university's new interactive web map, which forms part of the CampusGIS2 project⁴ (see next section).

In addition to field mapping courses in Geography, in which detailed spatial information on the UoC campus is collected, we offer interdisciplinary MA level teaching for students from the Digital Humanities, Media Computer Science and Geography in which the content and methods of the Virtual Campus project are disseminated, and the students are invited to work in groups to further improve them.

1.2 Related Work

The Virtual Campus project is tightly interconnected with the CampusGIS2 project, that develops a Spatial Data Infrastructure (SDI), a Geographic Information System (GIS) of the University of Cologne (UoC) campus, and an interactive web map application, with a focus on inclusiveness and accessibility for handicapped and otherwise disadvantaged or underprivileged users. The most prominent feature is the routing engine, that allows accessible and barrier free navigation on the UoC campus. To facilitate this highly data intensive routing, the UoC

¹ Extended reality (XR) is a broad term that covers augmented reality (AR), virtual reality (VR), and mixed reality (MR).

² Modelling is used here in two different, but related meanings: 1) the creation of 3D replica of physical objects, and 2) a heuristic method of meaning making and communication in science, scholarship, and system development (Ciula et al., 2023).

³ <https://virtueller-campus.uni-koeln.de/>

⁴ <https://campusgis2.uni-koeln.de/>

campus is mapped in high detail by the project members and by students during Geography field mapping courses. The CampusGIS2 project is a follow up to the original CampusGIS project (Baaser, 2010), which was funded from 2005 to 2010, developing an interactive Web Map that already had routing and navigation functionality for wheelchairs, with accessibility in mind. As part of the previous CampusGIS project mentioned above, a complete LoD2 model of the UoC Campus, and several LoD3 models of selected UoC buildings were developed (Willmes et al., 2010). This project already included a web application based on the degree3 implementation of the OGC Web Perspective View Service (WPVS) for accessing and visualizing the first 3D model of the University of Cologne campus.

A fully georeferenced 3D model of the buildings and infrastructure of the University of Cologne campus can be called a 3D CityModel (Biljecki et al., 2015) or a Digital geoTwin (Lehner and Dorffner, 2020) or location enabled Digital Twin (Ellul et al., 2022). We build our location enabled 3D building models and assets on the basis of heterogeneous data sources and workflows and software tools. We integrate data from OpenStreetMap, which provides data models for 3D as well as indoor data (Tauscher et al., 2024), we derive 3D data from pointclouds and developing techniques, tools and workflows for doing so, and researching for new technologies that facilitate this, for example AI based approaches (Borrmann et al., 2024). We also contribute to indoor positioning and navigation functionalities, outside of Apple's and Google's proprietary solutions, where a lot of research and development is underway currently.

Most existing 3D CityModels and Digital Twins of universities are based on esri's ArcGIS and CityEngine technologies or other proprietary one-stop-shop solutions. In contrast, the Virtual Campus project aims at building a Digital Twin of selected aspects of the university campus based on open source software and a platform-independent open standards based technology stack as presented in this contribution.

2. Technology Stack

The software tools and libraries that are combined to implement the project, as well as its applications (the so-called *Technology Stack*), are presented here. The system architecture and data flow of the Virtual Campus infrastructure are depicted in figure 1. The Virtual Campus model of the UoC can be called a 3D-SDI (Basanow et al., 2008). Technically, this is a PostgreSQL⁵ database with several extensions, such as PostGIS⁶. The database or SDI consists of many heterogeneous datasources. There is basic information and data about the UoC like building names, addresses, people, lecture halls, institutes, departments and so on, and detailed spatial data on possible barriers and accessibility, like surface conditions, path or door widths, inclination of surfaces, to name a few. It furthermore contains basic geodata from public and open sources like GeoBasis.NRW⁷ or OpenStreetMap⁸, detailed information about the infrastructure, including indoor floor plans of all Campus buildings of the UoC, generously provided by the infrastructure management department, and of course the already mentioned 3D building models.

⁵ <https://www.postgresql.org/>

⁶ <https://postgis.net/>

⁷ <https://www.opengeodata.nrw.de/>

⁸ <https://www.openstreetmap.org>

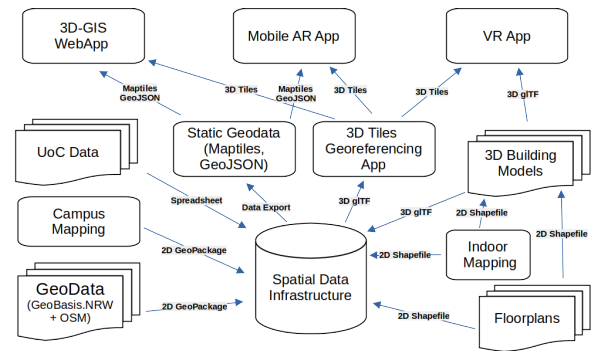


Figure 1. System architecture and data flow of the UoC Virtual Campus.

How this 3D SDI is designed and built, and how the data is acquired, generated, modelled and transformed for use in the various end-user applications of the project is described in detail in section 3. The Virtual Campus will have three main applications, the Web App (sec. 2.1), the mobile AR app (sec. 2.2) and the VR app. The main purpose of the VR app is its application in the UoC's immersive VR Environment CAVE (see section 2.3). The design and technology of these applications are described below.

2.1 Web Application

The web application is a CesiumJS⁹-based 3D view of the newly developed interactive map of the University of Cologne. The new CampusGIS2 map is a ReactJS¹⁰ and OpenLayers¹¹ based progressive web app (PWA), that provides a comprehensive geographic information system (GIS) about the buildings and infrastructure of the University of Cologne. The main focus is on accessibility and barrier-free navigation and any information that promotes inclusion and accessibility on the campus.

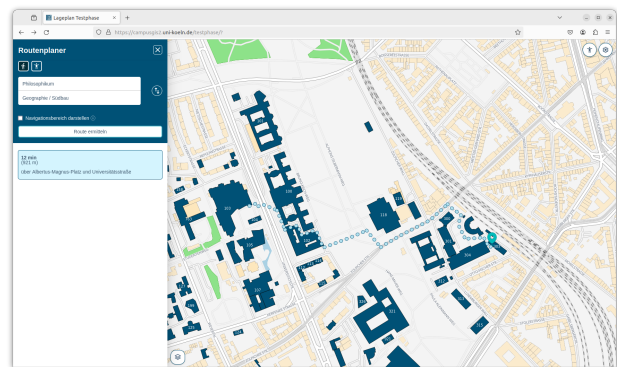


Figure 2. The CampusGIS2 web map application.

The web application was developed with the idea of not having any server-side logic or software dependencies other than a standard Apache web server provided and maintained by the IT Centre of the University of Cologne (ITCC). This decision was taken to guarantee sustainability of the application after the end of funding for its development. Thus, the website was designed to consist solely of static files (HTML, JavaScript, CSS, Routing graphs and media files such as pictures). All (geo)data used

⁹ <https://cesium.com/platform/cesiumjs/>

¹⁰ <https://react.dev/>

¹¹ <https://openlayers.org/>

by the application are given as static JSON, GeoJSON or vector and raster mapfiles files. The static data files are generated by export scripts from the CampusGIS2 SDI. These scripts are executed manually and the resulting static files can be transferred to the web space at any time.

Even the routing system operates entirely within the client's web browser, aligning with the project's overarching goal of minimizing server-side dependencies. The routing system is designed to empower users to navigate the University of Cologne campus efficiently and is tailored to their specific needs. By shifting the routing process to the client-side, we reduce the reliance on external servers, subsequently enhancing privacy, security, and minimizing request latency.

2.2 Mobile AR Application

The Virtual Campus project uses Flutter¹² for the development of its mobile applications. Flutter, a mobile app SDK from Google, enables the development of natively compiled applications for mobile, web, and desktop from a single codebase. It was also chosen for its efficiency in developing both iOS and Android applications without the need to maintain two separate codebases. The integration with Unity is facilitated through plugins, allowing Unity-based AR experiences to be embedded within Flutter applications.

Unity¹³ is used in combination with AR Foundation¹⁴ for the creation of AR functionalities within the mobile app. Unity is a widely used game development platform that supports the creation of 2D and 3D content, while AR Foundation provides a common API that abstracts the differences between ARCore¹⁵ (Android) and ARKit¹⁶ (iOS). It enables the development of AR applications for both platforms without platform-specific code. This setup enables the project to implement AR experiences such as interactive tours and information overlays on the virtual campus, utilizing features like plane detection, image tracking, and environmental understanding to enhance the user's interaction with the 3D model of the University of Cologne.

2.3 VR Application

The project also aims to develop VR applications, for example to interact with head mounted VR devices or for visualising the Virtual Campus in an immersive VR Environment (CAVE).

2.3.1 Immersive VR Environment (CAVE) The models created during the project will also be used for several purposes in the CAVE of the IT Center of the University of Cologne (ITCC). A CAVE is a room whose walls, ceiling and floor surround a viewer with projected, stereoscopic images (Cruz-Neira et al., 1992). Each surface measures 3x3 metres and is illuminated by a single laser projector with WQXGA resolution.

The software stack for the visualization of point clouds in the CAVE is entirely open source. We use the current LTS release of Ubuntu as operating system for the cluster. An open source software called COVISE (Rantzau et al., 1996) is used for rendering the virtual content. COVISE is an extendable distributed



Figure 3. The CAVE of the ITCC.

software environment to integrate simulations, postprocessing, and visualization functionalities in a seamless manner. The point clouds are specifically rendered using a plugin developed at the ITCC which employs the raytracing library Visionaray (Zellmann et al., 2017). In the case of point clouds one should call the algorithm used for rendering a ray caster instead of a ray tracer. A ray caster will not spawn secondary rays after the first hit. Instead the color information is computed directly from the first hit of each ray with the scene. This is possible because the illumination information is already contained in the point cloud, and doesn't have to be recreated by the render engine.

Because COVISE is almost entirely implemented in C++ it is not easy for end users to implement custom interactions and rendering algorithms. Therefore the software stack used to run the CAVE currently shifts towards using game engines, which will allow interested end users to create their own interactive content and visualizations. To this end the ITCC employs an Unreal Engine plugin developed by the RWTH Aachen, which facilitates the implementation of applications based on the Unreal Engine for multi-projection environments such as the CAVE (Krüger et al., 2024).

3. Data

The project developed an open source toolchain for creating and modelling the Digital Twin for the Virtual Campus Project. On the basis of georeferenced 2D data, i.e., architectural building plans and floor plans, as well as administrative geodata, Blender was used to create three dimensional models.

3.1 GIS Data

The Virtual Campus models are based on 2D floorplans and building footprints, as well as on the comprehensive geodatabase developed in the aforementioned sister project CampusGIS2. This geodatabase includes a highly detailed 2D map of the University of Cologne campus and its facilities, based on public (GeoBasis.NRW) and open (OSM) geodata sources. It is augmented by data gathered in Geography field mapping campaign courses, in which high precision geoinformation about barriers or surface properties and many other objects and features of interest for implementing a user friendly and accessible campus routing system were mapped.

¹² <https://flutter.dev/>

¹³ <https://unity.com/>

¹⁴ <https://unity.com/unity/features/arfoundation>

¹⁵ <https://developers.google.com/ar>

¹⁶ <https://developer.apple.com/augmented-reality/arkit/>

3.2 3D Data

The 3D building models are stored as gITF files. Since gITF is using WebGL, it is ideal for web-based 3D applications. Furthermore, the human-readable JSON format of its files is useful, as it makes automated modification of the metadata stored in the gITF file easier to implement. By embedding the 3D models using 3D Tiles version 1.1, it is possible to directly reference a gITF file as the content of a tile. Furthermore, it is possible to define metadata for each tile. Leveraging this has two main benefits for us. Firstly, to identify specific buildings, floors or even rooms of a building based on click events initiated by a user, in order to subsequently request more information on the clicked object from our backend. And secondly, to hide, show or style certain objects based on their metadata.

3.3 3D Tiles georeferencing app

We utilize 3D Tiles for streaming the 3D assets/models to the according applications. 3D Tiles is designed for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds (Cozzi and Lilley, 2023).

The 3D models and assets are handled in gITF format and are given in a local non-georeferenced coordinate system. gITF stands for Graphics Library Transmission Format or GL Transmission Format and was formerly known as WebGL Transmissions Format or WebGL TF (Khronos Inc., 2021).

In order to evaluate and visualize the placement of 3D building models and to gain a deeper understanding of 3D Tiles and their potential applications, a web GIS application was developed as part of the project. This application creates a main tileset that represents the whole campus and for each added building a child tile is created. In order to ensure that each building is identifiable within a web application, the building ID is included as metadata in the child tile. This is an automated process based on a predefined naming convention for the files. The main tileset is placed in the center of the campus and the position of the buildings is calculated relative to this point.

In the graphical user interface (GUI) – see fig. 4 – the user can select one or more .gITF or .glb (Graphics Library Binary) files. By default, a bounding volume is calculated for each building, and the buildings will appear at the center of the campus, as they initially lack a transform property. In the app, the user can specify the geographical positioning by defining the longitude and latitude for each building. Additionally, orientation and elevation details are adjustable, including height, heading, pitch, and roll. Furthermore, the interface includes buttons representing the cardinal directions, providing an alternative method for specifying positions.

All changes to these parameters trigger a recalculation of the *transform* property of the corresponding child tile and an update of the tileset resulting in a direct visual feedback. 'The transform property is a 4x4 affine transformation matrix, stored in column-major order, that transforms from the tile's local coordinate system to the parent tile's coordinate system—or the tileset's coordinate system in the case of the root tile' (Cozzi and Lilley, 2023).

As we want to show and hide individual floors in future 3D models, these can also be uploaded individually in the application. Based on a naming convention of the files, the floors

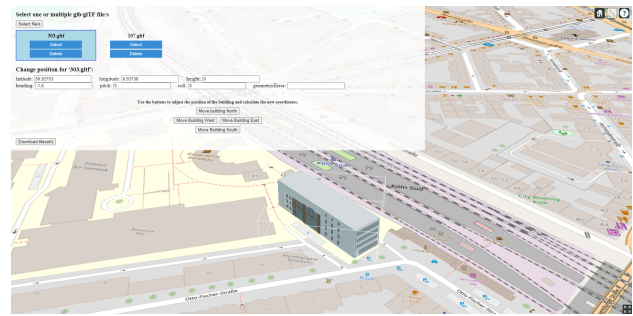


Figure 4. Screenshot of the 3D Tiles georeferencing app.

belonging to a building are automatically combined in a tile. Each floor is added as a child of the building's tile and metadata is added to ensure that each floor is identifiable.

The 3D Tiles georeferencing application is deployed within a Docker¹⁷ container and utilises the ReactJS framework and CesiumJS package.

3.4 3D Modelling in Blender

For creating the 3D building models, a semi-automated Blender based workflow was developed, using the Blender-GIS¹⁸ add-on to load the georeferenced floor plans. Three GIS layers are created for each floor plan, namely, *an outline of the floor*, *the rooms on that floor*, and *door lines*. Using the Blender scripting console, the GIS layers for each level are established and doors are cut out of the walls (see figure 5). The resulting 3D floor models are stacked on top of each other. The stairways, the elevators, and the exterior and facades of the buildings are modeled manually.



Figure 5. Screenshot of a 3D extruded floorplan in Blender.

Thanks to the seamless integration of scripting in Blender, this process can be easily automated for ordinary buildings. However, more complex buildings, such as buildings with lecture halls that extend over several floors, will require manual work. The same applies to adding stairs and roofs to the model.

3.5 Indoor Objects from Point Clouds

In the high detail 3D model of the USB, georeferenced indoor features are integrated into the library's rooms via a 3D to 2D pointcloud to vector data workflow. Firstly, this is done for all bookshelves in the areas that users frequent. The goal of adding bookshelves to the model is to perform so called "Weg-zum-Buch" navigation queries which direct users to requested bookshelves. Point cloud data collected with a hand-

¹⁷ <https://www.docker.com/>

¹⁸ <https://github.com/domlysz/BlenderGIS>

held SLAM LiDAR scanner¹⁹ provides the data basis for the georeferencing and modelling of the bookshelves. Every scan covers several rooms on one floor focusing on walls and on capturing the bookshelves from all sides. In CloudCompare²⁰ the pointcloud is segmented and the spatial information of the x- and y-coordinates is transferred into a 2D rasterfile. By employing QGIS' georeferencing tool the raster is georeferenced onto the library's room polygons stored in the CampusGIS2 geodatabase. This is done through matching the positions of wall corners as can be seen for the reading hall (see figure 7 a.)

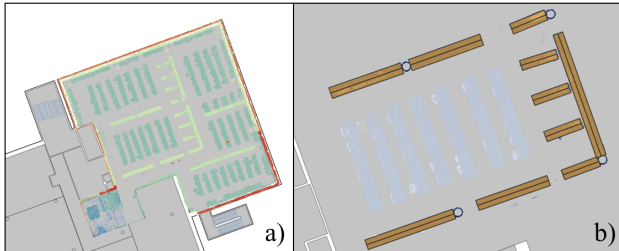


Figure 6. a.) Georeferenced raster of the USB reading hall. b.) Vector digitized bookshelves.

Once the raster is accurately georeferenced to the background polygon the position of the bookshelves within rooms can be used to digitize the bookshelves as vector data. Polygons are drawn over the raster image with very little accuracy loss (see figure 7 b.)). After the integration into the geodatabase the bookshelves can become endpoints for indoor navigation as part of the "Weg-zum-Buch" (way to the book) feature.

Additionally, the pointcloud-data can be used to model simple representations of freestanding objects within the buildings. For this process the pointcloud is exported as a .lts-file and imported into Blender, where it can be further cleaned up by deleting certain groups of vertices. Aside from far-out vertices and artifacts this also includes the ceiling of the current floor, since this makes it easier to view the insides of the individual rooms. In the next step primitive shapes such as cubes and cylinders are placed inside the rooms, and moved into the groups of vertices protruding from the ground level of the pointcloud. These primitives can then be additionally scaled and rotated as well as their individual vertices moved in the Edit-Mode in order to cover the pointcloud completely.

The shapes created in this way profit from the established accuracy of the LiDAR-Scanner and can be combined with the building-models mentioned in the previous subsection to create even more accurate collision-data. It may also be possible to use these primitives, in combination with photographic references, to create detailed models of indoor objects without the need for extensive measuring.

3.6 Context Information and Bookshelf ID's

Relevant context information, such as those named in the AR tours (regarding orientation, use of certain places and devices, special accessibility requirements like keys etc.; see section 4.2)

¹⁹ Emesent Hovermap ST. The scanners "LiDAR accuracy" is specified by the manufacturer as 30mm and that generally matches our experience. Even though our approach only generates an uncolored pointcloud, the capturing process is done significantly faster and easier, compared to other popular approaches. <https://emesent.com/hovermap-series/>

²⁰ <https://cloudcompare.org/>

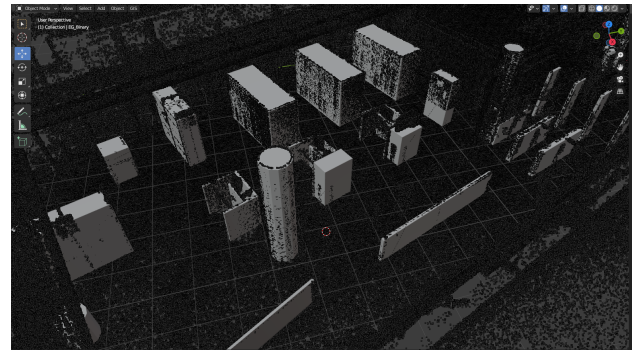


Figure 7. Screenshot of a room in the pointcloud filled with meshes derived from the pointcloud.

were sometimes not previously available online or held only by staff of the library. Therefore, they were gathered mostly through personal inquiries, research, and from scripts of guided tours. Information was first collected and organized in spreadsheets and are integrated into the projects central database. The mentioned "Weg-zum-Buch" navigation is using the location of the bookshelf as the endpoint, instead of the book's exact location. It is more efficient to take a whole bookshelf and store their range of book title signatures. The signatures here are strictly numerical and can be easily found by end users, once the correct shelf is found. Therefore, a new ID naming scheme for the shelves was introduced, composed of the ISIL signature of the library (International Standard Identifier for Libraries)²¹, the abbreviation of the library's section and a number, e. g. DE38-SAB-001. Within the library's OPAC system, each book title's JSON file has now the corresponding shelf ID stored. The ID of the bookshelf, together with the georeferenced polygon feature, will then enable the localization of and the navigation to endpoints.

4. Results and Applications

Based upon the aforementioned technology stack, we are in the process of implementing multiple software solutions and end-user applications within the Virtual Campus project. This section will present our development efforts and goals.

4.1 Web Application

The ReactJS- and OpenLayers-based interactive web map for the University of Cologne provides a CesiumJS based 3D view of the Virtual Campus and its 3D modelled buildings (see figure 8).

This app provides an interface to visually and spatially browse through the Virtual Campus 3D model. Because the 3D model is fully georeferenced it can be contextualized with any other geodata available for the UoC campus. Context information about the 3D models are accessible and displayed through clicking on 3D building models or simply through the base map that provides the spatial context.

4.2 Mobile Application with AR Functionality

The Android and iOS mobile applications focus on AR-services for navigation and for displaying information on the campus

²¹ <https://sigel.staatsbibliothek-berlin.de/suche?isil=DE-38>

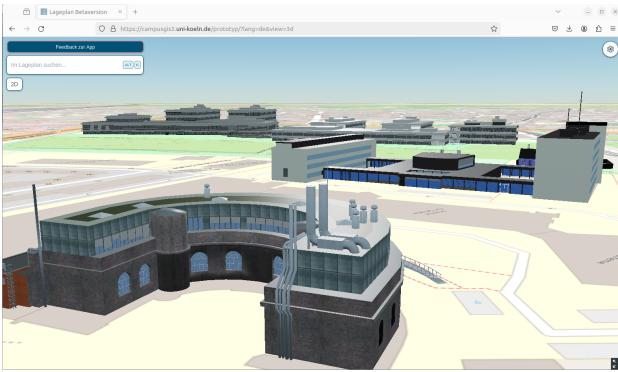


Figure 8. Screenshot of the 3D view of the new University of Cologne map webapp.

(what is where?) and on services (what can I do here? How does this work?). It includes features for locating books and classrooms, as well as navigating the campus. The applications feature a 2D campus map displaying points of interest with accessibility details, hours of operation, as illustrated in Figure 9. They also offer an Augmented Reality (AR) option for enhanced real-world navigation, shown in Figure 10 as a prototype of AR outdoor navigation. In addition, 3D Tiles are utilized for AR tours to streamline the georeferencing of assets, avoiding the need for multiple approaches to similar technological challenges.

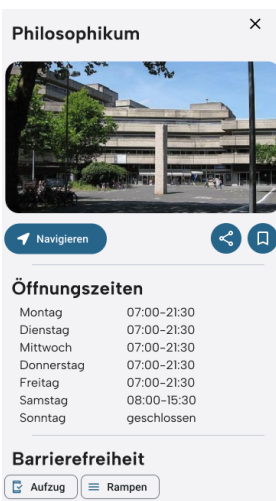


Figure 9. Point of Interest: Building information



Figure 10. AR-View: Outdoor navigation

The AR app includes two types of AR tours for getting to know the USB building and the library's various services, allowing members and visitors of the university alike to learn independently and interactively as they move through the building. First, a guided tour of key locations within the library helps new library users to become familiar with the layout of the building and how to search, locate, check out, and return books and other media. During the tour, users learn where to find freely accessible or previously ordered media, where to find self-issue and return machines, and where to find library staff for help with specific questions.

With a special focus on accessibility, these context information will be presented in multiple languages, simple language and various multimedia types, such as audio files, GIF's and videos,

depending on each point of interest. Second, a gamified tour of the USB provides similar information in an alternative, more playful way in order to appeal to users with different preferences. In this tour, a mascot character guides users through the building, for example encouraging them to solve a quiz. Gamification in AR offers even greater interactivity and is therefore, in some scenarios, suited to teach the sometimes complex interrelations of rooms, bookshelves and the organization of the library.

The contextual and cultural information presented in the AR app is modelled in accordance with CIDOC CRM (Bekiari et al., 2024), an ontology for cultural heritage developed by the international museum federation ICOM.²² This enables us to integrate seamlessly into the app virtual recreations of historical conditions of buildings as well as historical and cultural aspects of the campus environment more generally, as they will be developed, e.g., in student projects. We are also experimenting with the use of the CIDOC CRM extension CRMact (Velios, 2022) for planning future events; in our case, to make prototypes for future tours. This connects to the reconstruction and annotation aspects of the CAVE-Kompakkt integration mentioned below.

4.3 Immersive VR Experiences (CAVE)

In contrast to other VR-devices, most notably head-mounted displays, a CAVE allows for small groups to experience the virtual content together, while social behavioural patterns, such as establishing eye contact and turn taking, remain mostly intact. This makes the CAVE suitable for scenarios such as teaching small groups of students (Benölken et al., 2021), presenting scientific findings to collaborators and project evaluations.

In the project Virtual Campus the CAVE is used predominantly for teaching and collaborative data review task.

Building on top of the aforementioned Unreal Engine plugin developed by the RWTH Aachen (see 2.3.1), a direct connection to the Kompakkt database (Eide et al., 2020), is created. Kompakkt²³ is an open-source multi-modal object repository, 3D Viewer and 3D Annotation System run by the Institute for Digital Humanities for linking 3D objects to multimedia content and for gathering information through annotations in 3D space more generally. It enables users to share, explore, and collaboratively annotate objects in standard modern web browsers. The software development efforts at the ITCC aim at creating an additional frontend for the database, which will make bringing custom 3D models with their annotations into the CAVE as easy as uploading the files using the web interface.

With the 3D models uploaded into Kompakkt's object repository, visualizing them in the CAVE is easy and can be used e.g. for data review tasks as well as guided tours through the campus. It can also be used to show the history and constant change of the university's buildings. Especially the library has been remodelled heavily over the decades, making it an interesting subject for the virtual recreation of its current and historic conditions.

5. Discussion and Outlook

The here presented technology stack for the Virtual Campus of the University of Cologne is mostly based on Open Standards

²² <https://icom.museum/>

²³ <https://kompakkt.de/>

and Open Source Software, the only exception is the application of Unity based GameEngine technology, which may be replaced by an open source alternative in the future, if development resources within the project allow. This sustainable and open source technology stack for building location enabled Digital Twins (Ellul et al., 2022), is the main contribution from our research and development efforts presented here. Our findings and the technology stack can be used by the community, other universities, municipalities or government agencies, as well as in any kind of 3D CityModel project setting out to develop applications based on and for domain specific 3D CityModels and Digital Twins.

The main outcomes, products or results of this project are the described end user applications (web app, mobile app, VR app) and the Virtual Campus 3D model of the UoC Campus, which will be published in line with the FAIR principles as Open Data for reuse in future projects and applications. If circumstances allow we may also contribute some of our data to the Open Street Map 3D and indoor data efforts.

Interdisciplinary courses for students of Geography, Digital Humanities and Media Computer Science are organised by project participants at the University of Cologne. We aim to develop and publish open educational resources, making our approach to 3D Geoinformation modelling, Digital Twinning and the development of applications making use of the Virtual Campus 3D models accessible beyond the institutional setting in Cologne.

References

- Baaser, U., 2010. Das CampusGIS der Universität zu Köln - webgestützte Geodatendienste für raumbezogene Anwendungen. PhD thesis, Universität zu Köln, <https://kups.uni-koeln.de/3244/>.
- Basanow, J., Neis, P., Neubauer, S., Schilling, A., Zipf, A., 2008. *Towards 3D Spatial Data Infrastructures (3D-SDI) based on open standards — experiences, results and future issues*. Springer Berlin Heidelberg, Berlin, Heidelberg, 65–86. DOI: <https://doi.org/10.1007/978-3-540-72135-2>.
- Bekiari, C., Bruseker, G., Canning, E., Doerr, M., Michon, P., Ore, C.-E., Stead, S., Velios, A., 2024. *Definition of the CIDOC Conceptual Reference Model. Produced by the CIDOC CRM Special Interest Group. Version 7.1.3*. CIDOC, [Heraklion].
- Benölken, P., Bresges, A., Wickerth, D., Genz, F., 2021. Vr based explorations of the rhine riverbed. *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 291–296.
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889. <https://www.mdpi.com/2220-9964/4/4/2842>.
- Borrmann, A., Biswanath, M., Braun, A., Chen, Z., Cremers, D., Heeramaglore, M., Hoegner, L., Mehranfar, M., Kolbe, T. H., Petzold, F., Rueda, A., Solonets, S., Zhu, X. X., 2024. Artificial intelligence for the automated creation of multi-scale digital twins of the built world—ai4twinning. T. H. Kolbe, A. Donaubaue, C. Beil (eds), *Recent Advances in 3D Geoinformation Science*, Springer Nature Switzerland, Cham, 233–247.
- Ciula, A., Eide, Ø., Marras, C., Sahle, P., 2023. *Modelling between digital and humanities: thinking in practice*. Open Book Publishers, Cambridge.
- Cozzi, P., Lilley, S., 2023. 3D Tiles Specification. Technical report, Open Geospatial Consortium. <http://www.opengis.net/doc/cs/3D-Tiles/1.1>.
- Cruz-Neira, C., Sandin, D., Defant, T., Kenyon, R., Hart, J., 1992. The cave-audio visual experience virtual environment. *Communications of The ACM - CACM*.
- Eide, Ø., Wieners, J., Schubert, Z., Türkoglu, E., 2020. The intangibility of tangible objects: re-telling artefact stories through spatial multimedia annotations and 3d objects. *ICOM Kyoto 2019, 25th ICOM General Conference: Museums as Cultural Hubs: The Future of Tradition*, Zenodo.
- Ellul, C., Stoter, J., Bucher, B., 2022. Location-Enabled Digital Twins – Understanding the role of NMCAS in a European context. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W2-2022, 53–60. <https://isprs-annals.copernicus.org/articles/X-4-W2-2022/53/2022/>.
- Khronos Inc., 2021. glTF™ 2.0 Specification. <https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html>.
- Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdorf, C., Heazel, C., 2021. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. Technical report, Open Geospatial Consortium (OGC). <https://docs.ogc.org/is/20-010/20-010.html>.
- Krüger, M., Gilbert, D., Kuhlen, T. W., Gerrits, T., 2024. Game Engines for Immersive Visualization: Using Unreal Engine Beyond Entertainment. *PRESENCE: Virtual and Augmented Reality*, 1-25. https://doi.org/10.1162/pres_a.00416.
- Lehner, H., Dorffner, L., 2020. Digital geoTwin Vienna: Towards a Digital Twin City as Geodata Hub. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 63–75. <https://doi.org/10.1007/s41064-020-00101-4>.
- Rantza, D., Lang, U., Lang, R., Nebel, H., Wierse, A., Rühle, R., 1996. Collaborative and interactive visualization in a distributed high performance software environment. M. Chen, P. Townsend, J. A. Vince (eds), *High Performance Computing for Computer Graphics and Visualisation*, Springer London, London, 207–216.
- Tauscher, H., Heigener, D., Krishnakumar, S., Graichen, T., Schmidt, R., Richter, J., 2024. Ifc georeferencing for osm. T. H. Kolbe, A. Donaubaue, C. Beil (eds), *Recent Advances in 3D Geoinformation Science*, Springer Nature Switzerland, Cham, 193–209.
- Velios, A., 2022. *Definition of CRMact. An extension of CIDOC-CRM to support activity plans. Proposal for approval by the CIDOC CRM-SIG. Version 0.2*. CIDOC, [Heraklion].
- Willmes, C., Baaser, U., Volland, K., Bareth, G., 2010. Internet based distribution and visualization of a 3D model of the University of Cologne Campus. *3rd ISDE Digital Earth Summit, Nessebar, Bulgaria*. https://cwillmes.de/publications/CampusGIS-3D_Willmes_DigitalEarth2010.pdf.

Zellmann, S., Wickerth, D., Lang, U., 2017. Visionaray: A cross-platform ray tracing template library. *2017 IEEE 10th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, 1–8.