



Research Article

Map Whiteboard Cloud Solution for Collaborative Editing of Geographic Information

Karel Charvát^{1*} , Stein Runar Bergheim², Raitis Bērziņš³, František Zdražil⁴, Dailis Langovskis³, Jan Vrobel⁵, Šárka Horáková⁶ 

¹WIRELESSINFO, Cholinska 1048/19, 784 01 Litovel, Czech Republic

²Asplan Viak A/S, Kystveien 14, 4800 Arendal, Norway

³Baltic Open Solutions Center, Peldu iela 7, Jelgava, LV3002, Latvia

⁴Lesprojekt-sluzby Ltd, Martinov 197, 277 13 Zaryby, Czech Republic

⁵Forest Management Institute, Nabrezni 1326, 250 01 Brandys nad Labem, Czech Republic

⁶Czech Centre of Science and Society, Radlicka 663/28, 150 00 Praha 5, Czech Republic

Email: charvat@wirelessinfo.cz

Received: 27 April 2021; **Revised:** 15 July 2021; **Accepted:** 15 July 2021

Abstract: For the purpose of exploiting the potential of cloud connectivity in geographical information systems, the Map Whiteboard technology introduced in this article does for web mapping what Google Docs does for word processing; create a shared user interface where multiple parties collaboratively can develop maps and map data while seeing each other work in realtime. To develop the Map Whiteboard concept, we have applied a methodology whereby we have collected technical and functional requirements through a series of hackathons, implemented a prototype in several stages, and subjected this to rigorous testing in a lab environment and with selected users from relevant environments at intermediate scale. The work has resulted in a fully functional prototype that exploits WebSockets via a cloud service to reflect map and data changes between multiple connected clients. The technology has a demonstrated potential for use in a wide range of web GIS applications, something that is facilitated by the interfaces already implemented towards mainstream mapping frameworks like OpenLayers and QGIS-two of the most popular frameworks for Web GIS solutions. Further development and testing are required before operationalization in mission-critical environments. In conclusion, the Map Whiteboard concept offers a starting point for exploiting cloud connectivity within GIS to facilitate the digitalization of common processes within the government and private sector. The technology is ready for early adopters and welcomes the contribution of interested parties.

Keywords: cloud computing, visualisation, interactivity, collaboration, WebSockets, digitalization

Abbreviations

Term	Description
API	Application Programming Interface
ArcGIS	Complete cloud GIS mapping software
CAD	Computer-Aided Design
CARIS	Computer Aided Resource Information System

Copyright ©2021 Karel Charvát, et al.

DOI: <https://doi.org/10.37256/ccds.222021897>

This is an open-access article distributed under a CC BY license

(Creative Commons Attribution 4.0 International License)

<https://creativecommons.org/licenses/by/4.0/>

CGIS	Canada Geographic Information System
CPU	Central Processing Unit
CSW	OGC Catalogue interface
EDA	Event-Driven Architectures
ERDAS	Earth Resource Data Analysis System
ESRI	Environmental Systems Research Institute
FAIR	Findable, Accessible, Interoperable, and Reusable
GeoJSON	Open standard format designed for representing simple geographical features
GIS	Geographic Information System
HTTP	Hypertext Transfer Protocol
INSPIRE	Infrastructure for Spatial Information in Europe
KLM	Keyhole Markup Language
Leaflet	leading open-source JavaScript library for mobile-friendly interactive maps
MIDAS	Mapping Display and Analysis System
MOSS	Map Overlay and Statistical System
NASA	National Aeronautics and Space Administration
NRCan	Natural Resources Canada
OGC	Open Geospatial Consortium
OpenLayers	High-performance, feature-packed library for creating interactive maps on the web
QGIS	Free and Open Source Geographic Information System
RAM	Random Access Memory
SaaS	Software as a Service
SWE	Sensor Web Enablement
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service Interface Standard
WPS	Web Processing Service
YAMHub	YetAnotherMonolithicHub

1. Introduction

The concept of a “Map Whiteboard”, introduced in this article, was originally conceived as a “Google Docs for Maps”. To understand why that is useful and how it adds value to existing processes, we start by describing three user stories. Each story describes how the concept fits into an existing process, augments it, and improves the quality, timeliness, or costs relative to the present situation.

1.1 User story 1: Map Whiteboard as a stand-in replacement for physical planning meetings

The County Council is developing a new spatial plan for the future development of commerce, industry in Montgomery County. This work is carried out in close cooperation with state sectoral authorities for road infrastructure, railways, airports, seaports, and the environment. Furthermore, the work involves the major urbanities of Greenhill, New Town, and the rural district of Mountpleasant. The work is carried out as a combination of one-on-one meetings between the county’s planning department and each of the other stakeholders as well as joint meetings with many stakeholders present. Of late, following the pandemic, meetings are increasingly held online. In meetings, the different parties come together and unite over their interests and perspectives related to a shared map of the county and its built-up areas. The map is sometimes printed, other times shared via a video projector for all to see. Each party brings their information to the table and indicates important considerations and issues to be avoided by pointing to the map. Annotations are made either by a GIS operator present or by drawing on the map.

By utilizing Map Whiteboard technology, Montgomery County will enable all parties to share a common map view

where everybody can add their information from their corporate GIS systems and where everybody will have a cursor that is visible to all the others. That way each stakeholder can indicate to the other issues. Furthermore, all stakeholders can draw points, lines, and polygons on the map and assign attributes and visual styles to them. This enables the resolution of issues in the meetings that otherwise would have taken numerous rounds between the stakeholders to get right. The same technology will be used when processing planning permit applications from businesses and residents of Montgomery County, speeding up the process to get a permit, thus facilitating sustainable economic growth.

The “planning” user story was developed during the VGI & Citizens’ Observatories INSPIRE Hackathon held in Barcelona, Spain, in 2016.

1.2 User story 2: Agricultural advisor interacting with farmer

The YAMHub (YetAnotherMonolithicHub) project has developed a state-of-the-art technical platform that is capable of harnessing Earth observation (EO) and remote sensing data from a plethora of different sources for agricultural decision making on a very detailed level. YAMHub intends the tool to be used by Farmer Brown.

But, in its eagerness to provide a sophisticated tool, YAMHub partners have overlooked the fact that Farmer Brown does not hold a double Ph.D. in data sciences, does not possess a supercomputer, and would be using this tool at the very maximum of 10 times a year.

Thus, YAMHub relies on man-in-the-middle services provided by regional and local agricultural advisors who will use the platform on a regular basis as they each serve a multitude of farmers and agricultural businesses. Williams AgriConsult is one such agricultural advisory business.

They use YAMHub to conduct high-resolution spatial analysis for their client farms-but when the analysis is done, they require to discuss the results and how to apply them to the specific farm together with the farmer. By using the Map Whiteboard technology, Williams AgriConsult, and Farmer Brown collaboratively improve the input data for the model, improving its accuracy and utilizing its output on each subsequent run.

The “agriculture” user story was developed during the INSPIRE Hackathon organised as part of the INSPIRE Conference in Antwerp, Belgium, in 2018.

1.3 User story 3: Volunteer mapper interacting with subject matter or local expertise

The planning department of Hovězí Municipality is developing a plan to transform local agricultural production from live-stock to super-crops, seeking to boost the human-nutrition output while reducing the intensity of the land use.

For this, Hovězí requires ad-hoc data collection about current conditions and historically grown crops in the region from elderly people with good local knowledge, resident in the area. By using Map Whiteboard technology, municipal planner Jan Novak publishes a map where people can volunteer the necessary information by means of drawing on a map and specifying attributes in an electronic form. The web address is published in the local newspaper and distributed via an information campaign.

Adam Svoboda is a retired farmer with good local knowledge but a suspicious relationship to modern technology. With the help of his grandson, he is able to get online and has called Jan Novak on Skype. They are both looking at the same map and with the aid of his grandson, Adam is able to show Jan the exact location of relevant parcels and landscape features that are relevant.

The “volunteer mapping” user story was developed during the Open Spring INSPIRE Hackathon, organised virtually in the spring of 2021.

2. Technical baseline

The Map Whiteboard concept builds on well-established and stable principles of GIS and geographical information. To understand the context of our proposed solution, it is useful to step back and revisit the technical baseline and historical development that underpins our application. For the benefit of readers who may not be familiar with the terminology of geomatics, we introduce the fundamental concepts and terms in this section.

Geographic information can be defined as:

- information about places on the Earth's surface
- knowledge about where something is
- knowledge about what is at a given location [1]

A geographic information system (GIS) is a computer-based tool supporting the management, visualisation, and analysis of spatial data, as defined by the OECD Glossary of Statistical Terms. Till 1960, maps were mapped in paper form. The first spatial analysis mentioned in literature was a density map of cholera death created by the French geographer Charles Picquet in a report on the cholera outbreak in 1832 [2]. A similar principle was adopted by John Snow in London, incidentally also for monitoring a cholera epidemic, as described by Tate (2018) in Geospatial World.

The first instance of a GIS, in the sense of a computer application, is considered to be the Canada Geographic Information System (CGIS) [3], which dates back to 1960. After 1970 a number of GIS platforms were developed. Among the first GIS platforms, the Map Overlay and Statistical System (MOSS) [4] is worthy of mention. This was developed in the second half of the 1970s and was one of the first Open Source GIS software products. At the same time, started the development of another very popular Open Source GIS software, GRASS [5], that has survived and is in active use to this day. Major commercial platforms launched in the early 1980s include the “Arc” product line from Environmental Systems Research Institute (ESRI) [6]. Other solutions that date back to this era grew out of the ever closing gap between GIS and Computer-Aided Design (CAD); Microstation was developed commonly by Intergraph and Bentley Systems Incorporated [7]. The contender who created the first desktop GIS was the MapInfo Corporation with its Mapping Display and Analysis System (MIDAS) [8], later renamed MapInfo [9]. Other tools that shaped the GIS technology from that time include ERDAS (Earth Resource Data Analysis System) [10] and CARIS (Computer Aided Resource Information System) [11]. With the development of the multi-tasking operating system, MS Windows many GIS platforms migrated to this environment [12].

The 1990s brought two important events, which have been influencing GIS development until the present time. The first was the creation of the World Wide Web at the start of the decade. In 1994 the first Web Map Viewer followed, the Xerox PARC MapViewer (1994), and shortly after that, the first GIS WebServer MapServer.

It was the move of GIS onto the Web that constituted a real “revolution” in the world of GI. It rapidly increased the volume of the data and opened GIS technology and new applications to large communities. New applications were developed and the number of users rapidly grew. Web GIS also provides for user-driven context and creates a demand for collaborative solutions. Over the past two decades, analytical GIS has been brought onto the Web, too. That brought with it more requirements on standardization, improving discovery of data and demand on semantics and Linked Data [13].

The second important event of the 1990s was the foundation of the Open Geospatial Consortium (OGC), as described by McKee (2021) in OGC History (Detailed). The need for the OGC was triggered by the proliferation of different solutions for publishing data on the web. Publishing data on the Web required better mechanisms for sharing data across systems and professional domains as well as standardized data models. The Open Geospatial Consortium (OGC) is an international consortium of more than 500 businesses, government agencies, research organizations, and universities driven to make geospatial (location) information and services FAIR—Findable, Accessible, Interoperable, and Reusable (OGC, 2021; available at www.ogc.org). It presently connects over 500 organizations worldwide cooperating on the definition of geospatial standards. The most frequently used OGC standard is the Web Map Service Interface Standard (WMS), which provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. This standard is now commonly used all over the world [14]. Among other commonly used OGC standards, it is necessary to mention the Web Feature Service (WFS), Web Coverage Service (WCS), Sensor Web Enablement (SWE), OGC Catalogue Service (CSW), and Web Processing Service (WPS). The OGC has also adopted leading de facto web standards for geoinformation, including Google KML (formerly Keyhole Markup Language).

The OGC came to adopt leading de-facto web standards for geoinformation, including Google KML. This open and easily parsable format opened GI information to large communities [15]. In recent years the OGC has focused on high-performance lightweight APIs [16]. Adjacent to OGC standardization, GeoJSON has become widely popular on account of its ease of use and relatively compact nature. GeoJSON is not OGC standard but is accepted by the OGC community [17]. For the Map Whiteboard, it was important to use both GeoJSON and the OGC standard known as Web Map Context [18], which inspired our present ideas on how to share maps in addition to the data that maps are made

from.

The next shift in the development of geographic information (GI) came with the transition to cloud environments and the development of publicly available data repositories. Early solutions that left a major mark on the sector include NASA WorldWind and Google Maps/Google Earth.

Now GI Cloud services are offered by large cloud providers such as Amazon-Amazon Web Services (AWS) or Microsoft-Microsoft Azure Functions for the management of geospatial big data, and also by major GI providers like ESRI ArcGIS Online. In Europe, Data and Information Access Services (DIAS platforms) are being developed to support easier access to data from the European Earth Observation Programme Copernicus and to enable new services built on top of them. Parallel to this, there also exists a vast number of smaller cloud solutions provided by minor data providers. Parallely to this there also exist a vast number of small clouds provided by smaller data providers [19].

As was stated, new Web and Cloud Technologies moved GI closer to citizens and opened possibilities for collecting data from and by local communities. Voluntary geographic information (VGI) is a concept that describes born-digital spatial information collected/created by individuals [20]. Some examples of this phenomenon are WikiMapia, OpenStreetMap, and Google Map Maker [21] and as a specific concern within online or web credibility [22].

Voluntary data collection is an important part of data processing. To this end the term Neogeography (New Age Geography) was introduced, combining geotagged data with a map interface [22-24].

Last year, the World Wide Web Consortium (W3C) together with Natural Resources Canada (NRCan) organised a joint W3C-OGC Workshop on Maps for the Web, which presented a wide range of information and opinions on the topic of standardising maps for the Web.

The first idea of WhiteBoard was conceived during the first Barcelona INSPIRE Hackathon, where the goal was to develop “Google Docs for Maps”.

3. An introduction to Map Whiteboard technology

The vision of the Map Whiteboard innovation was born out of a sequence of large-scale collaborative writing efforts using Google Docs [25].

As opposed to traditional offline word processing tools, Google Docs allows multiple people to edit the same document [26]-at the same time-allowing all connected clients to see changes made to the document in real-time by synchronizing all changes between all connected clients via the server.

The ability to work on a shared body of text, avoiding the necessity to integrate fragments from multiple source documents and with multiple styles removed many obstacles [27] associated with traditional document editing. The Map Whiteboard technology seeks to do the same for the traditional use of GIS tools.

The overall vision for the technology is that a Map Whiteboard will be to GIS what Google Docs is to word processing.

3.1 Analysis of traditional GIS workflows

The traditional workflows for GIS (as shown in Figure 1, below) have grown out of digitizing manual processes that predated the modern computer [28], making them more efficient and scalable, but not in a manner that is disruptive or that exploits the inherent potential of connected and networked technology, nor the capabilities of cloud services.

Looking at the white boxes in Figure 1, we may see that GIS makes it possible for expert users to create maps, visualise geospatial data, and conduct advanced geometrical, topological, and graph analysis on desktop computers. GIS content created in professional tools can also be published on the Internet and shared with others with relative ease.

The dominating enabling technology for online sharing and exchange of spatial data is a suite of standards developed by the OGC, particularly the CSW, WMS, and WFS standards for service discovery, raster and vector data access, respectively.

The WMS and WFS-protocols define a set of operations that enable decentralized clients to access data from a GIS server without manually downloading data to their computers. That has helped break down some of the silos formed by proprietary GIS software suites and has greatly facilitated inter-organization, multi-purpose sharing of geographical information.

That sharing concept is a critical precursor to the Map Whiteboard described in this article.

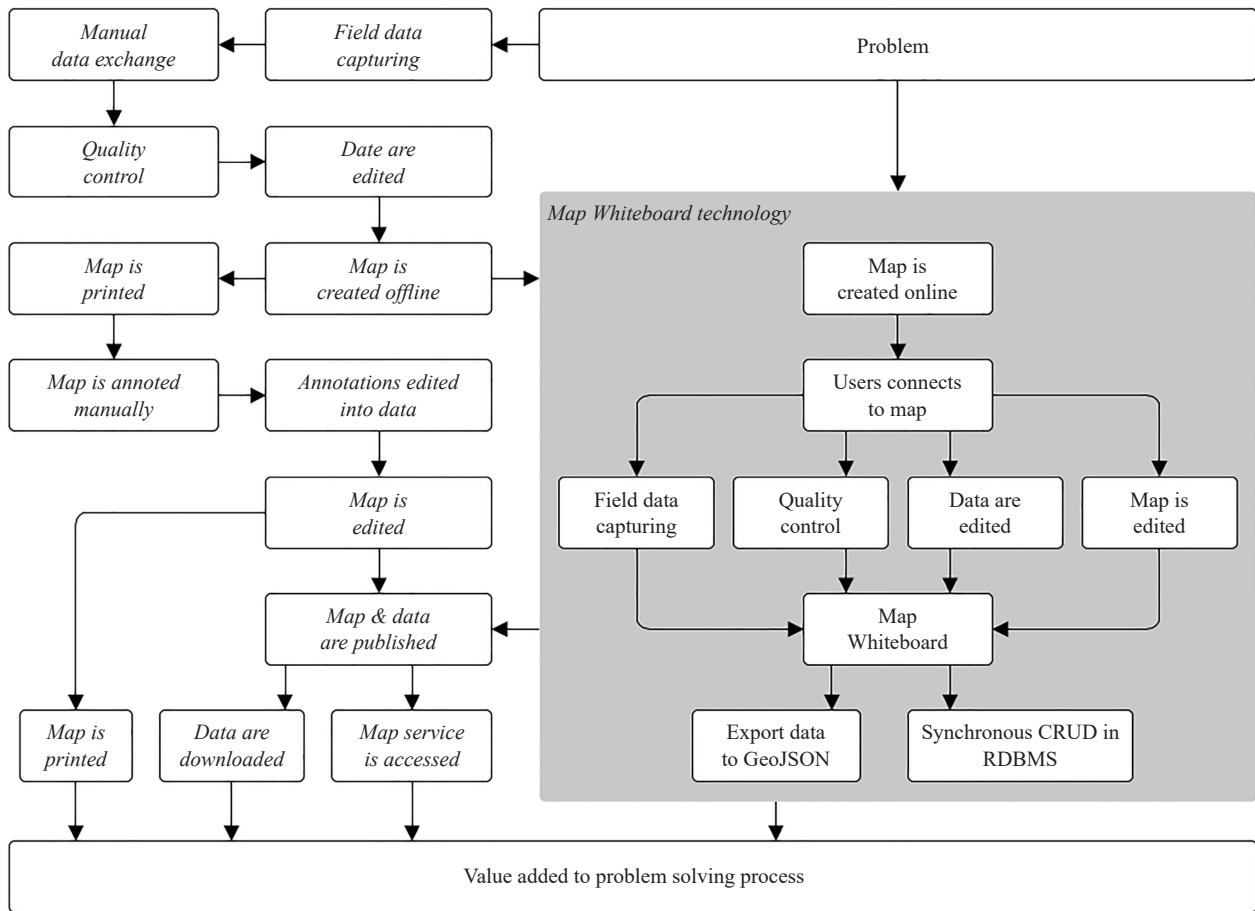


Figure 1. Traditional GIS-workflow aligned with envisioned Map Whiteboard workflow for professional use

With the increasing adoption of GIS by end-users in non-traditional domains [29], there has been a shift from desktop to web-based applications, not only for publishing information but also for professional use. That is the second factor that is of importance to the Map Whiteboard concept.

In the present situation, where people are working with “thin clients” that communicate towards the same set of shared services [30], the potential for collaboration is significantly expanded compared to the setting when everybody worked locally on their computers.

Professional Web GIS solutions are typically integrated into web map portals that have highly heterogeneous user interfaces [31]. The capabilities of the software emulate well-known concepts from desktop GIS.

Collaborative data creation has become increasingly common and has given rise to concepts like user-generated content [32] (UGC) and VGI [20]. The interaction process whereby users contribute data is typically based on users working individually towards standard data models using a web form and a map surface.

The proliferation of mobile smart devices with GIS and positioning capabilities has accelerated the uptake of collaborative data creation in the domain of geospatial information [33].

Still, manual process steps and printing [34] are yet common in government sectors where expert GIS-users act as a human interface between digital mapping technology and non-technical subject matter experts within domains such as policy development, planning, and analysis.

3.2 Proposal for alternate Map Whiteboard-based GIS-workflow

The Map Whiteboard seeks to upgrade the manual and linear GIS-workflows that have been inherited from the era when desktop software dominated the field. The technological concept builds on state-of-the-art, as described here, and seeks to exploit some of the capabilities native to web applications, such as the Web Sockets protocol that enables continuous and instantaneous message exchange between multiple connected clients.

Our experiments (shown in the grey rectangle of Figure 1) demonstrate the technical feasibility and the potential value creation of the technology within land-use planning, agriculture, forestry, and transport planning, and field data capturing.

An additional dimension to the Map Whiteboard technology is the importance of sharing rather than only sharing data. Spatial data are subject to (mis)interpretation; maps are the means to express these interpretations in a manner suitable for communicating purpose, intent and shared understanding.

The Map Whiteboard is not limited to editing a shared dataset. The technology shares the entire map, presenting the data in the context of any background information that may be useful.

It allows multiple clients to see the identical map interface and simultaneously draw on it, emulating the widespread use case of sitting around a meeting table looking at a map together, annotating it, pointing at “things”, and proposing changes.

To realize the vision of the Map Whiteboard technology, it is necessary to make certain behaviour of GIS applications interoperable across instances and platforms. The interoperability is limited to the smallest common denominator between different platforms in terms of the types of business objects that are present in each application, what functions they expose and which events they emit.

Figure 2 shows an abstraction of the way mapping frameworks work in terms of discrete object types, properties, and events that are necessary to realize the Map Whiteboard technology. The below diagram holds for major frameworks for web development such as OpenLayers [35], and Leaflet [36], as well as desktop frameworks like ArcGIS [37] and QGIS [38]. A thorough study of other mainstream frameworks has not been conducted.

GIS applications typically dedicate the majority of the user interface to a Map control that displays map content. Maps are composed of one or more layers of “drawings” that are drawn or “stacked” on top of each other in a specific order. A layer can be either a pre-rendered raster image (or set of images) or it can be mathematical vector geometries, drawn by the client application.

A vector layer consists of vector features. A feature typically has a geometry and one or more descriptive properties called feature attributes.

Maps have a set of associated controls that can either execute an action when you click on them or that require some interaction with the map in the form of drawing, selecting, dragging, or zooming by means of a human interface device such as a mouse, stylus or touch-sensitive surface.

All the objects emit events that make it possible to track the changing state of the map and propagate changes to connected clients so that these may synchronize the user interface among each other.

Each client must be able to pan and move independently on the map. All other users must however see the cursors of other connected users to emulate “pointing”, as people would do, standing over a meeting room table.

Clients must be able to draw on the map and separate what they draw into discrete layers with diverging data models. Users must be able to modify each other’s drawn features and to edit the attributes associated with each feature.

It must be possible to add information to the map, such as adding a data layer to use as background maps or supplementary information to the topic under discussion. Finally, it must be possible to take data “out of” the Map Whiteboard and bring it into ordinary GIS workflows such as professional editing, visualization, and analysis.

To describe and represent the state of a map to the level indicated in Figure 2, we employ a standardized notation called a Map Composition that is an extension or application profile of the OGC OWS Context [39] standard that allows the embedding of user markup and other features omitted from the underlying standard. A map consists of layers, which in turn consists of raster tiles, or vector features which in turn consist of geometries which finally consist of points. The various elements that must be recorded and tracked to enable the propagation of an identical map to multiple connected clients are shown in Figure 2.

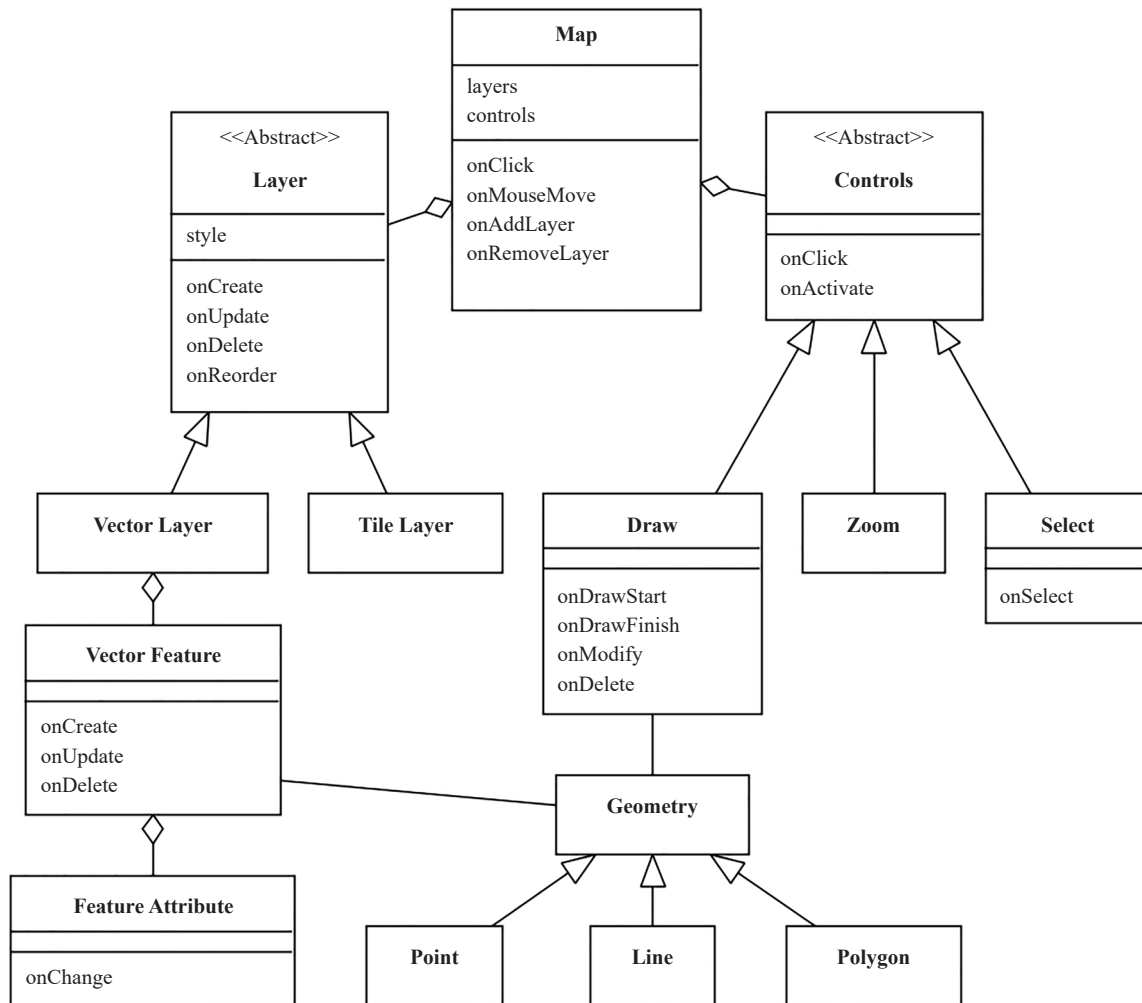


Figure 2. Abstraction of common features GIS applications that are used by the Map Whiteboard technology

4. Materials and methods

The following sections elaborate on the working methodology that has been employed in the design, development, and validation of the Map Whiteboard technology.

4.1 Overall architecture

The software architecture of the Map Whiteboard, as expressed in Figure 3 consists of three discrete parts that are integrated via a set of standard instructions and messages as defined by a Web Sockets API and a RESTful API. These three are:

1. Cloud service implementation: the overall system including data storage, business logic, and web server components.
2. Web clients integration: a library that integrated with common web mapping tools and that enable the efficient exchange of messages/API-requests.
3. Desktop clients integration: a library that integrates with mainstream desktop tools, demonstrating the applicability of the Map Whiteboard in a mixed web/desktop, occasional/professional GIS-user.

In the following sections, the content of each of the three parts is described under its headline.

4.2 Server implementation

To realize the goals described above, it is necessary with a highly efficient data storage component that is capable of high combined read-write workloads.

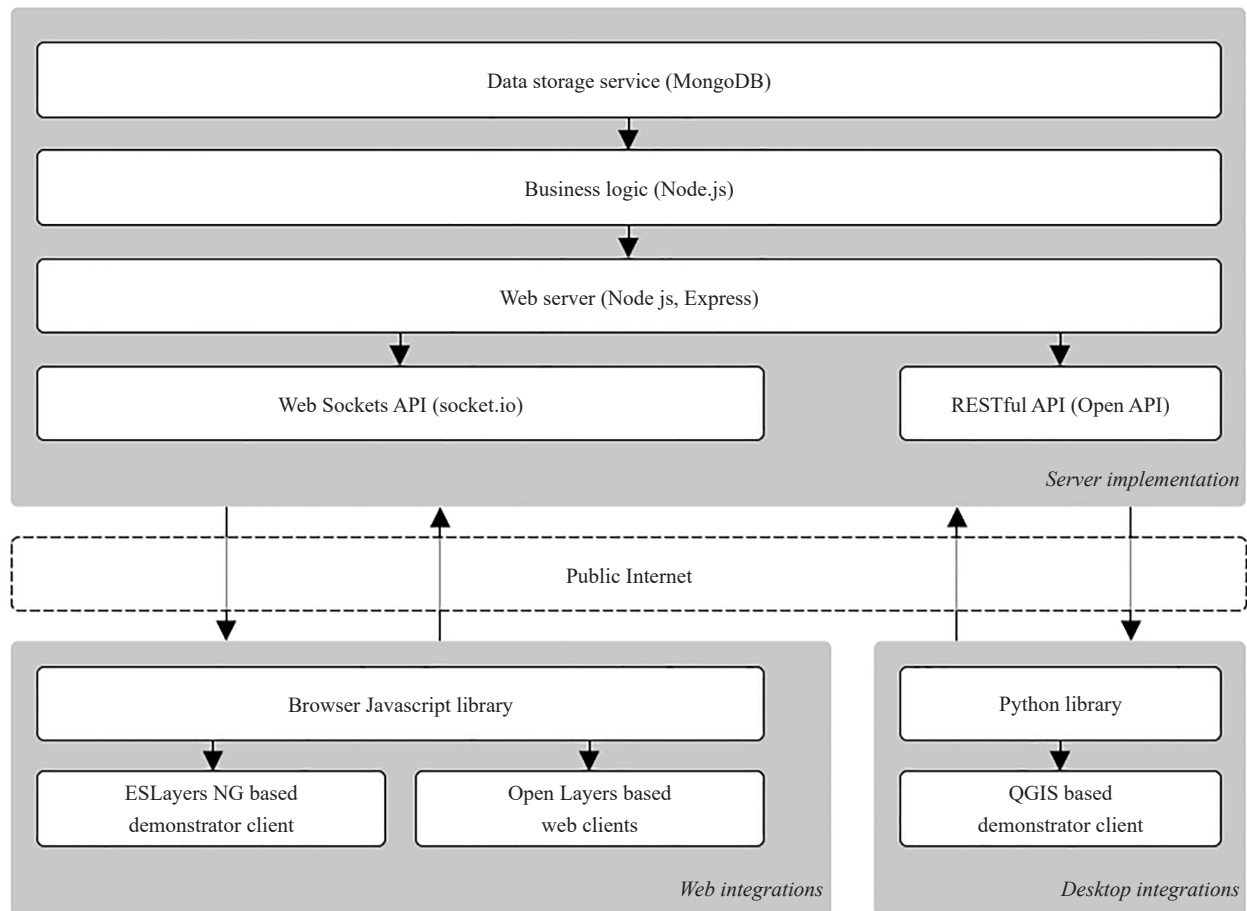


Figure 3. Overall architecture of Map Whiteboard technology

For this purpose, a traditional relational database management system (RDBMS) is not suitable as the table-based structure requires indices to be rebuilt on each insert, causing excessive load on CPU and disk when used by many simultaneous users.

The Map Whiteboard therefore instead utilizes a scalable document store, electing to use JSON-based MongoDB as the back-end for the service. By combining MongoDB and the popular JSON-based standard for geographical information GeoJSON, an end-to-end integration between data-storage and client layer is achieved without the need to modify or alter data structures.

The business logic is implemented in JavaScript using Node.js as the execution environment and the Express library as a server implementation on top of Node.js' native HTTP component.

The use of JavaScript as an implementation language served two purposes.

1. Ability to work natively with JSON objects.
2. Ability to use the most popular implementations of Web Sockets on the client and server-side; Socket.io.

Communication with clients relies on two different protocols that operate in parallel:

1. A WebSockets [40] message API for high-frequency, low-latency message exchange over a persistent open connection. This is used for the exchange of cursor positions, feature creation, updates, and deletions as well as changes

to map content, such as the addition of layers, etc.

2. A RESTful API that is used for low-frequency and high volume transfers that might otherwise interfere with the continuous operation of the Web Sockets message queues. This is used for administrative operations such as uploading maps, assigning user rights, etc.

All the components of the cloud server implementation are bundled into a software suite that runs behind a web server that is accessible to clients via HTTPS from any Internet-connected location.

4.3 Web client integrations

Clients that utilize the Map Whiteboard must implement a set of instructions that pass between their map components and the Map Whiteboard server. There is a very large number of potential integrations that can be made, but to start with a technology that has substantial market adoption, the Map Whiteboard at the outset targets integrations with any OpenLayers-based map client.

To this end, a browser-based JavaScript library has been developed that adds Map Whiteboard functionality to existing instances of OpenLayers maps. The library implements a set of default tools and controls that may be shown inside the maps, on par with the built-in zoom buttons, as well as an extended set of API methods that allow client applications to activate/deactivate/invoke the functions from their user interface controls.

The library attaches itself to the event model of OpenLayers as illustrated in Figure 1, and uses feature creation events and pointer movement map events to propagate messages to connected clients via web sockets.

A more detailed description of the OpenLayers integration as well as the built-in demonstrator client is provided later in this article.

4.4 Desktop client integrations

The Map Whiteboard technology aims to add-value to professional workflows. Therefore, a key part of the initial integration is to prove that the concept operates across platforms. That means not only cross-web clients but also between a professional desktop GIS client and a lightweight web client.

As with web applications, there is a large body of existing GIS applications that are widely used in government and by the private sector. For the proof of concept to gain traction among potential adopters, the project has opted to target QGIS.

QGIS has emerged to become a credible alternative to proprietary GIS suites and offers robust implementations of the most common functions and capabilities found in suites like ESRI's ArcGIS portfolio, MapInfo, or Autodesk's AutoCAD Map 3D, to name but a few.

QGIS features a customizable application engine and user interface via a Python API that provides access to map events, map content, styling, as specified in Figure 2 above.

A more detailed description of the QGIS integration is provided later in this article.

4.5 Cloud service implementation

The Map Whiteboard technology is not based on direct peer-to-peer communication but relies on clients to be connected to a shared server instance that collects and distributes messages among them. The principal envisioned service delivery mechanism is Software as a Service.

The normal operation of the Map Whiteboard causes a high frequency of small messages, mainly updated vector features and users' cursor locations. While the size of each message is negligible, the accumulated volume and frequency cause a significant CPU and RAM load on the server-side. This means that the use of the technology will incur costs on some level, which in turn will require a business-balanced model.

We have considered three different scenarios:

1. A single centralized cloud service on the model of "Google Drive" where the Map Whiteboard is offered from a single, scalable universal end-point.
2. A centralized service with the option of private clouds [41] or on-premise solutions that can be hosted in a private cloud or any data centre for organizations and corporations that wish to host their solutions.
3. Solely private clouds or on-premise solutions.

The first option would offer the greatest flexibility as it would require no effort on the part of clients wishing to utilize the technology. On the other hand, as has been demonstrated, there would be substantial costs associated with running the centralized service and it remains to be analyzed whether a business model can be put in place that secures sufficient revenue to guarantee continuous operation.

The second option offers greater flexibility as it combines a central service offering with the possibility for parties who for security, sensitivity, competitiveness, or other reasons wish to host their Map Whiteboard server. It also opens up the opportunity to offer the server technology as a product for installation in private clouds, something that might bring access to corporate revenue streams.

The third option is the one that instantly is cheapest to realize, but the lack of a centralized service would constitute an obstacle to small-and intermediate-scale integrations and thus be a limiting factor for widespread adoption. As a foundation for the work, it has been decided to move on with option two, not excluding one or three until the completion of technical validations.

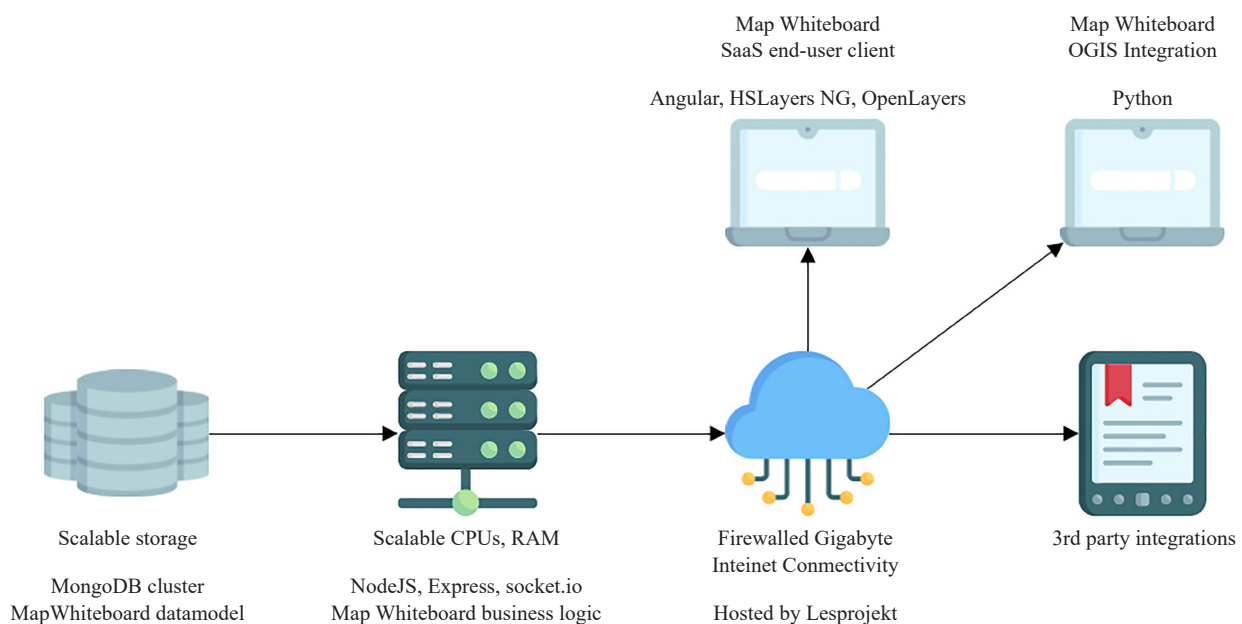


Figure 4. Principal components of Map Whiteboard cloud infrastructure

As part of the development and validation efforts, the service has been installed in a cloud environment operated by Lesprojekt-sluzby in the Czech Republic, as shown in Figure 4. This relies on a storage layer consisting of MongoDB installed in one node and a NodeJS server application built on top of Express and socket.io in another. These sit behind a firewall that is open for HTTPS and WebSocket communication and connected to a Gigabit Internet connection.

The entire server installation is Dockerized and runs on a Ubuntu Linux image. It is horizontally scalable in terms of the storage layer, but the business logic is as of yet monolithic.

On top of the infrastructure, a self-service administration application has been created that allows users to create and publish maps as well as assign user rights to them to determine who shall be able to access which map based on a user and role mechanism.

4.6 API definition

Besides software, the core of the Map Whiteboard technology is a composite API consisting of a set of structured WebSocket messages and RESTful web service requests/responses. This section describes the principle of operation for these APIs.

The system uses a mix of REST services through HTTPS for simple requests which don't need to be propagated to connected clients such as listing and modifying maps, layers, or downloading and uploading layer data. Websockets on the other hand are used for high performance, low-overhead, and latency data transmission which flows both ways from client to server and from server to multiple clients such as cursor movement, geographical feature synchronization (See Figure 5), and modifications to individual layer definitions such as styling, order, title, and other properties. To circumvent circular loops where data is sent to the server, propagated back to the client of origin and then sent again to the server, we use a collection of feature IDs/layer IDs to ignore on the next update on the client-side feature collection (vector layer).

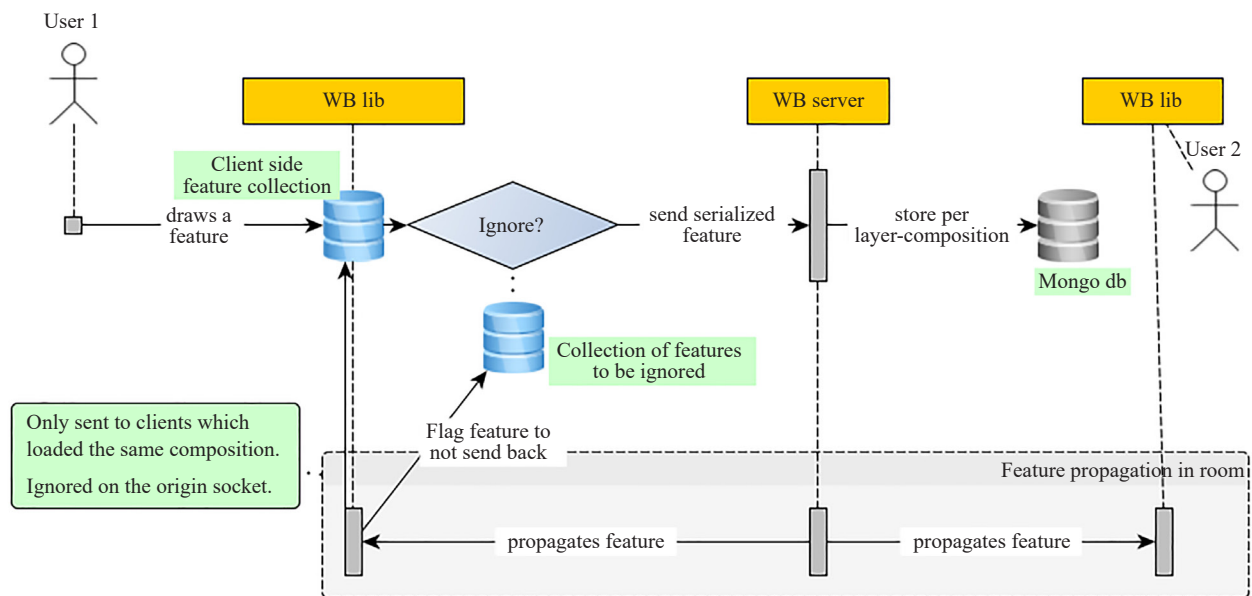


Figure 5. Principle for synchronization of features between multiple clients connected a shared Map Whiteboard instance

The REST API is documented using the OpenAPI standard and contains CRUD operations for:

- maps (compositions)
 - metadata
 - user privileges for specific maps
 - layers
 - features
 - styles
 - order
- user and session management

Some of the operations are initially called through a REST endpoint, but upon execution on the server are propagated to the connected clients to synchronize the map state as soon as possible and not wait for the user to reload the page. This is enabled by mapping individual socket connections to user sessions used by REST queries with the help of session tokens.

4.7 HSLayers NG integration

As the initial user interface, which we select for implementation of Map Whiteboard is HS Layers NG [42]. There were two reasons, the solution as Open Source is implemented by us and also HSLayers NG support work with Map composition [43].

Since Openlayers library provides the only barebone user interface and focuses more on the map functionality

itself, we needed to rely on other systems to provide:

- layer management;
- picking features on Vector and WMS layers;
- styling of layers and individual features;
- interface to add external data sources;
- printing and sharing;
- operate drawing tools etc.

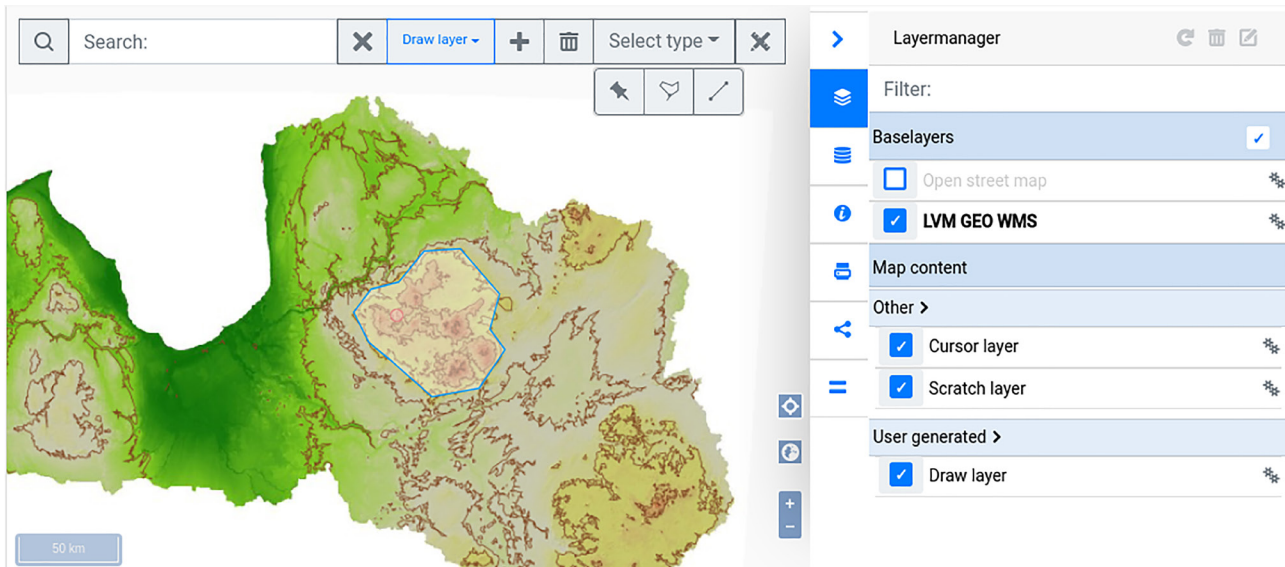


Figure 6. Screenshot from the standard Map Whiteboard client offered as part of SaaS self-service application

All of these needs fit well with the open-source Hslayers-ng client-side library which provides the user interface and extra functionality around the OpenLayers map (Figure 6). It is built on the Angular 11 framework and depends on OpenLayers version 6.5.0. The open-source library handles tasks which the Whiteboard library plans to solve in the future, but does not do it just yet, such as serialization and parsing layer types other than Vector layers i.e WMS, ArcGIS REST, WMTS, WFS. Vector layer features, such as points, lines, polygons, can be drawn on the map using the draw toolbar, located on the top left corner of the map or using the draw panel, found among the list of other available Hslayers-ng UI panels. Features can be described, updated, or deleted using an attribute editor and later downloaded through the UI provided by Hslayers-ng. Layer management is controlled from Hslayers-ng UI panel-Layermanager, supporting many functionalities, such as setting layer rendering order, changing each layer visibility, editing each layer properties, etc.

Integration with the Whiteboard library is done through calling methods in the library itself i.e. *loadMap*, *activate*, *deactivate*, and also through event call-backs, *manipulation* functions, such as *parseLayer*, *parseLayerStyle*. Later, the processing of a layer or style definition is done in two steps, first the Whiteboard library tries to do its best to parse a serialized description and create OpenLayers objects itself. Then it gives control to the external application to do further result processing, incorporated into the map in the form of new layers or layer styles. This enables separation of concerns and loose coupling of the Whiteboard library which deals with synchronization of map content, Hslayers-ng which provides the user interface for GIS functionality and the container application providing main administrative functionality and binding the different parts together.

To be able to work with a collaborative map using the Whiteboard library, the Openlayers map object reference, initialized on the Hslayers-ng library side, gets passed over to the Whiteboard library. This is done to make sure that both libraries are referencing and working with the same map object. In case that the map has not been created, but the

Whiteboard library client side is already loaded, the user can still create a new composition, and modify it, because the library also works without initialized Openlayers map object.

The map composition format, used in this solution, was first implemented in the Hslayers-ng library to store map state between page reloads and later extended with extra parameters such as layer style definitions, ways of storing map features directly in composition definition.

Hslayers-ng library includes a server application providing proxy functionality to be able to serve data from other domains and hide sensitive information such as API keys used to query these data sources and web services.

4.8 QGIS client implementation

The Map Whiteboard solution is not limited just to the web environment but can be integrated even with desktop GIS. Since QGIS is one of the most popular open-source geographic softwares, the first integration was naturally aimed at it [44].

QGIS is extensible by Python or C++ based plugins which the end-user can install through the native plugin library or from the ZIP file. Python language has been used to implement the Map Whiteboard tools and is currently available in an experimental version named WhiteBoard in the QGIS plugin library.

The plugin communicates with the Map Whiteboard Server via the Socket. IO library which is not a part of the QGIS core installation but is included with the plugin so no additional setup is needed from the user. WebSockets API of the Map Whiteboard Server is implemented and the plugin currently supports the following features:

- creating new users;
- creating and deleting new maps;
- set permissions to map per user;
- share map by link;
- load map shared by link;
- add and remove map features from QGIS map canvas in real-time;
- add and remove layers from QGIS map canvas in real-time;
- support the cursor layer of another user.

All of this core functionality has been implemented and tested, so the user is now able to work with the same set of maps regardless of the environment (desktop or web). The refresh rate of displaying changes is under 100 ms.

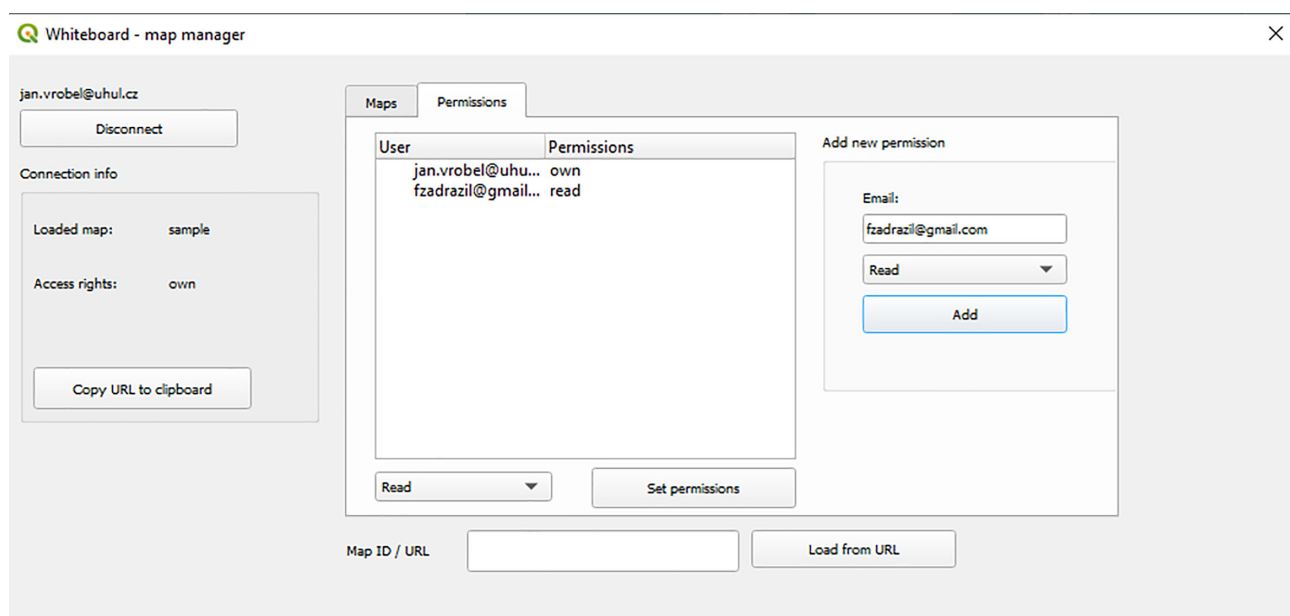


Figure 7. Screenshot of map manager dialogue implemented in QGIS as part of Map Whiteboard technology

The Map Whiteboard QGIS plugin provides a user interface (Figure 7) based on the Python QT library. After connecting to the server, a list of existing maps is displayed (maps owned by the user or maps shared with him). The GUI provides components for setting user permissions for the maps owned by the current user. The existing map can also be loaded from its URL generated by other clients.

After successfully loading the map the user can move to QGIS canvas and use it as usual. There are layers from the selected map automatically loaded and also a cursor layer that shows the mouse position of other users in real-time. From this point on the user can add or delete layers in the same manner as in the standard QGIS project workflow. The plugin then operates event-based and propagates all the occurring changes automatically to the Map Whiteboard Server and modifies the map in the server storage.

The concept of Temporary Scratch Layers of QGIS is used for locally storing the content of editable vector layers of the Map Whiteboard maps. Those scratch layers are only in-memory objects, meaning that they are not saved on the disk and will be discarded when QGIS is closed.

5. Results

To date, the effort has resulted in a functional prototype at a technical readiness level (TRL) of 5 to 6 published at <https://dih.bosc.lv/whiteboard-demo/admin/>.

The technological results include:

1. An API specification defining RESTful requests/responses and WebSocket messages.
2. A Map Whiteboard server.
3. A Map Whiteboard SaaS self-service application with a built-in map client.
4. A Map Whiteboard client-side JavaScript library for integrations with OpenLayers.
5. A Map Whiteboard plugin for integration with QGIS.

The technology has been validated in relevant environments through guided user trials both in controlled settings and in decentralized environments where test users have been working independently and without guidance.

The concept has proven to be workable from a technical point of view. By combining low overhead technologies in transmission and storage such as WebSockets, NodeJS, and MongoDB we obtained a roundtrip measurement of ~250 milliseconds for sending a modification request of one point in a reasonably small polygon (36 vertices). This includes also the time waiting for acknowledgement from the server. For cursor movements, the latency of ~60 ms was even lower, because it didn't involve acknowledgement packets and storage.

We saw no considerable delay in cursor synchronization speeds in a close-to real-life scenario involving 35 users connected to a single map. The bottleneck for complex maps involving multiple imagery layers turned out to be the loading of aforementioned map tiles through a centralized proxy which will need to be addressed by caching.

A transaction and queuing mechanism needed to be developed to handle simultaneous reading and writing Map Composition files and common layer/feature records at the same time from different clients.

6. Conclusions

The Map Whiteboard technology has been realized and demonstrated from to the level of a working prototype on Technical Readiness Level (TRL) 5 “Component and/or breadboard validation in the relevant environment”.

User feedback from demonstrations and interest shown by clients and cooperating partners in our existing markets justifies continued efforts to advance to TRL 6 “System/subsystem model or prototype demonstration in a relevant environment.” This requires the integration of the Map Whiteboard concept into an existing application that supports a real-world business case. Several relevant environments are under consideration in Norway, Latvia, and the Czech Republic. Use cases for the Map Whiteboard technology are also identified in the H2020 PoliRural project.

There are several known issues that need further work before the concept is mature enough for being used in mission-critical solutions, these include:

6.1 Technical and functional issues

1. The technology must be proven to scale to a larger number of concurrent users and must be tested in environments with multiple groups of users working on multiple maps.
2. The code must be extended to include unit tests. The codebase is extensive and both on the client-side and on the server-side there are too many classes, methods, and properties to keep track of them manually.
3. The WebSocket API must be made self-documenting in a manner similar to Open API, possibly using the AsyncAPI standard. AsyncAPI is an open-source initiative that seeks to do for Event-Driven Architectures (EDA) what Open API does for RESTful APIs.
4. An automated benchmark test for end-to-end testing of the system must be implemented with coverage of 80+%.
5. A login mechanism should be implemented to permit the use of social logins and auto-creation of account profiles based on these when the users authenticate using the third-party provider.
6. Conflict resolution when multiple users are editing the same geometry simultaneously: it is necessary to identify a mechanism that either preserves and fuses all changes or a locking mechanism that prevents two users from editing the same geometry simultaneously.

6.2 Operational and business model issues

1. Integration with existing communication tools: The Map Whiteboard is well-positioned to work as a complement to online conferencing tools such as Zoom, Teams, GoToMeeting, etc. Specific integrations should be explored.
2. One or more business models must be elaborated that can secure the sustainability of the Map Whiteboard technology in an operational context. There must be sufficient revenue to support the development and maintenance of the code base and enough added value to justify the investment from the perspective of customers. Customers may be individuals accessing the SaaS service or they may be corporations/organizations that wish to host an instance of the Map Whiteboard server technology in a private cloud.

Thus far, the concept has been brought forward through a series of consecutive challenges at Inspire Hackathons. The Map Whiteboard most recently won the first prize at the Open Spring INSPIRE Hackathon in June 2021 with a two-part challenge that elaborated two of the open issues, namely (1) the elaboration of a business plan and (2) the definition of a framework for scalability and testing of the technology.

Work is also supported by several EU Horizon 2020 projects, including PoliRural-Future Oriented Collaborative Policy Development for Rural Areas and People, DEMETER-Empowering Farmers, STARGATE-Resilient Farming by Adaptive Microclimate Management, and SmartAgriHubs-Unleashing the Innovation Potential for the Digital Transformation of the European Agrifood Sector, all of which have a vested interest in the application of the technology as part of envisioned workflows and service delivery concepts.

Author contribution

The concept of the published research, methodology and validation of experiments, check of the results, writing and editing the main part of the original draft was executed by Karel Charvát and Runar Bergheim; Runar Bergheim also supervised the entire research and compilation of the article; Raitis Bērziņš, František Zdražil, Dailis Langovskis, Jan Vrobel provided technology studies and contributed to the research text writing; Šárka Horáková contributed to the research text writing, editing, and translation. All authors have read and agreed on the published version of the manuscript.

Acknowledgements

The paper was prepared on the basis of materials collected by all authors. Work is also contributed with support from the EU projects.

For functional requirements and support in implementing the prototypes:

- PoliRural-Future-Oriented Collaborative Policy Development for Rural Areas and People (Grant agreement ID: 818496)
 - DEMETER-EMPOWERING FARMERS (Grant agreement ID: 674973)
 - STARGATE-reSilienT fARminG by Adaptive microclimaTe managEment (Grant agreement ID: 818187)
- For the hackathons that have been used as a vessel to carry forward the work:
- Plan4all z.s. (Identification number in the Czech legislation: 03698416)
 - SmartAgriHubs-Connecting the dots to unleash the innovation potential for the digital transformation of the European agri-food sector (Grant agreement ID: 818182)

Conflicts of interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- [1] Lü G, Batty M, Strobl J, Lin H, Zhu AX, Chen M. Reflections and speculations on the progress in Geographic Information Systems (GIS): a geographic perspective. *International Journal of Geographical Information Science*, 2016; 33(2), 346-367.
- [2] Koch T. *Cartographies of Disease: Maps, Mapping, and Medicine*. Redlands, CA: Esri Press; 2017.
- [3] Harvey F, Kwan MP, Pavlovskaya M. Introduction: critical GIS. *Cartographica: The International Journal for Geographic Information and Geovisualization*. 2005; 40(4): 1-4.
- [4] Veenendaal B., Brovelli M.A., Li S. Review of web mapping: eras, trends and directions. *ISPRS International Journal of Geo-Information*. 2017; 6(10): 317.
- [5] Neteler M. Bowman MH, Landa M, Metz M. GRASS GIS: A multi-purpose open source GIS. *Environmental Modelling & Software*. 2015; 68: 70-85.
- [6] Longley PA, Goodchild MF, Maguire DJ, Rhind DW. *Geographic Information Science and Systems*. Hoboken, New Jersey: Wiley; 2015.
- [7] Chang KT. Geographic information system. *International Encyclopedia of Geography: People, the Earth, Environment and Technology*. John Wiley & Sons, Ltd.; 2016. p. 1-10. Available from: <https://doi.org/10.1002/9781118786352.wbieg0152> [Accessed 15th March 2021].
- [8] Goodchild MF. Reimagining the history of GIS. *Annals of GIS*. 2018; 24(1), 1-8. Available from: <https://doi.org/10.1080/19475683.2018.1424737>
- [9] Wilson JP, Fotheringham AS. *The Handbook of Geographic Information Science*. Wiley; 2017.
- [10] Dobesová Z. Cognition of graphical notation for processing data in ERDAS IMAGINE. *ISPRS International Journal of Geo-Information*. 2021; 10(7): 486.
- [11] International Hydrographic Organization & Open Geospatial Consortium. (2020). Development of Spatial Data Infrastructures (SDI) for Marine Data Management: IHO-OGC Marine SDI Concept Development Study. Monaco: IHO. Available from: https://portal.opengeospatial.org/files/?artifact_id=88037.
- [12] Limpouch A, Charvát K. Quo vadis GIS: From GIS to GIMS and Open GIS. *International Conference on Current Trends in Theory and Practice of Computer Science*. Berlin, Heidelberg: Springer; 1995. p. 334-349.
- [13] Kuhn W, Ballatore A. Designing a language for spatial computing. In: Krisp JM. (ed.) *Computing with Spatial Trajectories*. Cham, Switzerland: Springer; 2015. p. 309-326.
- [14] Blower JD, Masó J, Díaz D, Roberts CJ, Griffiths GH, Lewis JP, et al. Communicating thematic data quality with web map services. *ISPRS International Journal of Geo-Information*. 2015; 4(4): 1965-1981.
- [15] Huang CY, Chang H. GeoWeb crawler: An extensible and scalable web crawling framework for discovering geospatial web resources. *ISPRS International Journal of Geo-Information*. 2016; 5(8): 136.
- [16] Schleidt K, O'grady M, Grellet S, Feliachi A, Van Der Schaaf H. ELFIE-The OGC Environmental Linked Features Interoperability Experiment. *International Symposium on Environmental Software Systems*. Cham: Springer; 2020. p. 188-193.
- [17] Liang C, Li X. The Design of the lightweight WebGIS based on GeoJSON by using HTML5 canvas. *Computer*

science and Application. 2012; 2(4): 189-196.

- [18] Charvat K, Vohnout P, Sredl M, Kafka S, Mildorf T, De Bono A, et al. Enabling efficient discovery of and access to spatial data services. *International Journal of Advanced Computer Science and Applications*. 2013; 3(3): 28-31.
- [19] Řezník T, Lukas V, Charvát K, Krivánek Z, Kepka M, Herman L, et al. Disaster risk reduction in agriculture through geospatial (Big) data processing. *ISPRS International Journal of Geo-Information*. 2017; 6(8): 238.
- [20] Zhang G, Zhu AX. 2018. The representativeness and spatial bias of volunteered geographic information: a review. *Annals of GIS*. 2018; 24(3): 151-162.
- [21] Basiri A. Crowdsourced geospatial data quality: challenges and future directions. *International Journal of Geographical Information Science*. 2019; 33(3): 457-476.
- [22] Panek J, Netek R. Collaborative Mapping and digital participation: a tool for Local Empowerment in Developing countries. *Information*. 2019; 10(8): 255.
- [23] Harris T, Lafone F. Toward an informal Spatial Data Infrastructure: Voluntary Geographic Information, Neogeography, and the role of citizen sensors. In: Čerba O, Čerbová K. (eds.) *SDI, Communities, and Social Media*. Prague, Czech Republic; 2012. p. 8-21.
- [24] Charvát K, Čerba O, Kafka Š, Mildorf T, Vohnout P. *International Symposium on Environmental Software Systems*. Berlin, Heidelberg: Springer; 2013. p. 1-10.
- [25] Albahri A, Al-Emran M, Shaalan K. Collaborative writing with Google Docs: a systematic review. *Education and Information Technologies*. 2019; 24(3): 2973-3000.
- [26] Liu SHJ, Lan YJ. Social constructivist approach to web-based EFL learning: Collaboration, motivation, and perception on the use of Google docs. *Journal of Educational Technology & Society*. 2016; 19(1): 171-186.
- [27] Jung YW, Lim YK, Kim MS. Possibilities and limitations of online document tools for design collaboration: The case of Google Docs. In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. USA: Association for Computing Machinery; 2017. p. 1096-1108.
- [28] Fu P. *Getting to Know Web GIS*. 3rd ed. Redlands, CA: Esri Press; 2018.
- [29] Morse WC, Kutner R, Kulczycki A, Wilson TD. Using Public Participation Geographic Information Systems (PPGIS) for coastal vulnerability mapping: Exploring stakeholder values and sea-level rise in the U.S. Gulf Coast. *Sustainability*. 2020; 12(17): 6711.
- [30] Morse WC, Kutner R, Kulczycki A, Wilson TD. Using Public Participation Geographic Information Systems (PPGIS) to assess a wide variety of landscape values and their vulnerability: Coastal vulnerability mapping on the U.S. Gulf Coast. *Sustainability*. 2020; 12(17): 6711.
- [31] Agrawal S, Gupta RD. Web GIS and its architecture: a review. *Arabian Journal of Geosciences*. 2017; 10(23): 1-13.
- [32] Giuffrida N, Le Pira M, Inturri G, Ignaccolo M. Mapping with stakeholders: An overview of public participatory GIS and VGI in transport decision-making. *ISPRS International Journal of Geo-Information*. 2019; 8(4): 198.
- [33] Melumad S, Inman JJ, Pham MT. Selectively emotional: How smartphone use changes user-generated content. *Journal of Marketing Research*. 2019; 56(2): 259-275.
- [34] Gao S, Mai G. Mobile GIS and Location-Based Services. In: Huang B, Cova TJ, and Tsou MH. (eds.) *Comprehensive Geographic Information Systems*. Oxford, UK: Elsevier; 2018. p. 384-397.
- [35] Karlsson M, Stöckel F. *The Myth of the Paperless Office: Is there a key amongst the clouds?* Umeå universitet, Institutionen för informatik; 2017. p. 26.
- [36] Hazzard E. *Openlayers 2.10 Beginner's Guide*. UK: Packt Publishing Limited; 2011.
- [37] Gratier T, Spencer P, Hazzard E. *OpenLayers 3: Beginner's Guide*. UK: Packt Publishing; 2015.
- [38] Law M, Collins A. *Getting to Know ArcGIS Pro*. Redlands (CA): Esri Press; 2019.
- [39] Menke K, Smith R Jr, Pirelli L, Van Hoesen J. *Mastering QGIS : Go Beyond the Basics and Unleash the Full Power of QGIS with Practical, Step-By-Step Examples*. UK: Packt Publishing Limited; 2016.
- [40] Gonçalves P, Brackin R. *OGC OWS Context GeoJSON Encoding Standard. Version 1.0*. Wayland, MA: Open Geospatial Consortium; 2017.
- [41] Pautasso C. *RESTful Web Services: Principles, Patterns, Emerging Technologies. Web Services Foundations*. New York: Springer; 2014. p. 31-51.
- [42] Fette I, Melnikov A. *The websocket protocol*. Internet Engineering Task Force; 2011. Available from: <https://www.greenbytes.de/tech/webdav/rfc6455.pdf> [Accessed 15th March 2021].
- [43] Salah EE, Ahmed AS, Hazem ME. Overview of using private cloud model with GIS. *International Journal of Electronics and Information Engineering*. 2017; 7(2): 68-78.
- [44] Popelka S, Herman L, Řezník T, Pařilová M, Jedlička K, Bouchal J, et al. User evaluation of map-based visual analytic tools. *ISPRS International Journal of Geo-Information*. 2019; 8(8): 363.
- [45] Charvat K, Čerba O, Kozuch D, Splichal M. Geospatial data based environment in INSPIRE4Youth. *Procedia*

Computer Science. 2017; 104: 183-189.

- [46] Barik RK, Das PK, Lenka RK. Development and implementation of SOA based SDI model for tourism information infrastructure management web services. *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence) IEEE*. 2016. p. 748-753.