COLLABORATIVE VALIDATION OF USER-CONTRIBUTED DATA USING A GEOSPATIAL BLOCKCHAIN APPROACH: THE SIMILE CASE STUDY

J. R. Cedeno Jimenez [1,*], A. Folini [1], M. A. Brovelli [1]

[1]Department of Civil and Environmental Engineering, Politecnico di Milano, Italy
- (jesusrodrigo.cedeno, andrea.folini, maria.brovelli)@polimi.it

Commission IV, WG IV/4

KEY WORDS: Citizen Science, Blockchain, Geospatial, Smart Contracts, Environmental.

ABSTRACT:

Internet decentralization nowadays represents a critical topic to be addressed. It protects the users' privacy, promotes data ownership, eliminates single points of failure and data censorship. An element that has an important role in decentralization is blockchain technology. Although blockchain has revolutionised sectors like the financial one with Bitcoin, there are still some fields where it needs to be further developed. One of these is geospatial data sharing and citizen science, where features like decentralization, immutability and transparency are needed. This study focuses on the description of a decentralized application developed specifically for geospatial data-point sharing and validation. As an example, the Informative System for the Integrated Monitoring of Insubric Lakes and their Ecosystems (SIMILE) is used. This application is developed in the Velas blockchain infrastructure and implements a combination of a Discrete Global Grid System (DGGS) with smart contracts. Two types of smart contracts were created, a cell and a registry smart contract. The cell smart contracts are individual for each DGGS partition and contain the list of observations present in a specific area. The registry smart contracts keep track of all the DGGS cells added to the system. Currently, SIMILE observations are validated by public authorities, which requires time that is not always available. Therefore, a fully working prototype was developed to solve this. Here users can add and manage personal observations and validate the ones belonging to other users. This work demonstrates the feasibility of creating decentralized applications for geographical data validation as a citizen science solution.

1. INTRODUCTION

The Internet is currently the most important communication channel in the world (Karadsheh et al., 2022) and since its creation, it has evolved dramatically. According to some authors, Internet's transformation can be divided into three stages Web 1.0, Web 2.0 and Web 3.0 (Majid and Verma, 2018). Web 1.0 was born in 1989 with the invention of the World Wide Web (WWW) by Tim Berners (Riordan, 2022). This initial stage of the web was characterized by 'one-way' information sharing. Later, in 2006 Web 1.0 evolves to Web 2.0 where a bilateral communication model developed. Bilateral communication permitted users to interact with each other (Alabdulwahhab, 2018, Riordan, 2022), and nowadays it is mainly composed of an application layer that promotes this interaction (e.g. social networks). Most of these applications are based on a centralised approach. Centralization is based on the concept of having a single actor (e.g. a tech company) hosting all of the information of a data portal. This situation benefits the loss of democracy and promotes censorship. The reason is that a single institution or data owner can decide either to stop sharing the information or manipulate it to his convenience. Another problematic element in centralisation is that the user can be harmed by losing ownership of their data. As a consequence, users are at risk of being shut down by companies and losing their data (Alabdulwahhab, 2018).

Blockchain is the driving element of the Internet transformation to Web 3.0 and decentralisation (Vogel, 2015). The concept of a blockchain, also called distributed ledger, consists of a chain of blocks containing an established number of transactions. Each of the blocks is stored and validated by the participants of a network employing a consensus mechanism. For the blocks to be chained, each contains the hash value of the previous one. A hash value is a sequence of characters generated after processing a string with a mathematical operation. Although the resulting value is not unique, the probability of collision (i.e. two blocks having the same hash) is extremely low (in the order of 2^{-39} depending on the mathematical operation being used) (Nikolić and Biryukov, 2008). This means that, by chaining the elements with each other, the slightest modification in a transaction would change completely the hash number of its block. This in turn would modify the next block and all of the subsequent ones (Nofer et al., 2017). Not only the computational cost of a modification is extremely high but it would require that the majority of the participants agree to modify each of the blocks.

Given the mechanism described in the previous paragraph, using blockchain technology as a data-sharing instrument makes the infrastructure decentralized, immutable and transparent. Decentralization is achieved by distributing the complete ledger among peers belonging to the network. Immutability is guaranteed on the one hand, by hashing and chaining the blocks and on the other hand through consensus between participants. Transparency is accomplished by making the network a public one, therefore anyone interested can review the ledger and track any transaction. Recently, data provenance has become of interest to the blockchain community and for some infrastructures is integrated as an additional security layer. In this case, geolocation data is used to track the geographical behaviours of the users when making transactions inside the blockchain.

* Corresponding author

Blockchain can help an ecosystem to regulate the transactions inside it through consensus and eliminates the need for a central institution that takes unilateral decisions. Therefore, blockchain has revolutionized some sectors, such as the financial industry with cryptocurrencies (Zarrin et al., 2021) with Bitcoin (Nakamoto, 2008). In 2014 the Ethereum platform (Buterin, 2014) remodeled the blockchain by implementing smart contracts. These allow the execution of general-purpose computations by using Solidity, a Turing-complete programming language (Lone and Naaz, 2021). However, there is still a lot of work to do in other areas. Such is the case of geospatial data sharing, where citizen science can help the communities to participate in the digital representation of the Earth. Crowdsourcing and volunteered geographic information (VGI) can be demonstrated successful. An example of this is Wikipedia, a free encyclopedia with more than 79 million registered users. Another case is OpenStreetMap (OSM), the largest existing geospatial database with more than 5 million contributors (Brovelli et al., 2020). For this reason, citizen science can benefit from the features of blockchain technology.

This work focuses on the development of a blockchain crowdsourcing infrastructure to be used by the Informative System for the Integrated Monitoring of Insubric Lakes and their Ecosystems (SIMILE). SIMILE is a cross-border Italian-Swiss project to improve the collaboration between public administrations and stakeholders for the management of the Insubria lakes (Lugano, Como, and Maggiore) and their ecosystems, as well as monitoring water resources quality (Brovelli et al., 2019, Bratic et al., 2022). One of the main sources of data in SIMILE is collected with a Citizen Science approach where the data is obtained from normal citizens through their smartphones. The observations of this type include data about water quality, meteorological parameters and multimedia files such as images (Carrion et al., 2020). The data can be currently validated by the public authorities managing the platform (Biraghi et al., 2021, Vavassori et al., 2021) but this requires time which is not always available to technicians. Therefore this development tries to solve this problem, by implementing a decentralized application where users can register and validate data through a voting mechanism. This will help to reach a consensus on the quality of the observations.

2. STATE OF THE ART

2.1 Geospatial blockchain

Up to the moment of this writing, only a few examples of decentralized applications attending to the needs of geographical citizen science have been developed. The best example is FOAM Map. The main purpose of FOAM is to build a consensus-driven map that uses the Ethereum blockchain as a platform. The objective of using the blockchain is to provide cryptographic utility tokens as an incentive to perform the computational work and verification of the network. Tokens in turn empower users to build a decentralized permissionless network (Foamspace-Corp, 2018).

FOAM Map is specifically designed to solve the problems of location encoding (currently there are no standards for location encoding in smart contracts), user experience (an interface capable of representing geospatial data in decentralized applications) and location verification (currently there is no decentralized trusted location verification service). To solve the first problem, FOAM employs the Crypto Spatial Coordinates standard, which links an address in the Ethereum network with a specific geographic location. For the user experience, FOAM developed the Spatial Index and Visualizer, a general-purpose front-end visual blockchain explorer. Finally, for location verification, a verification mechanism enables the users to verify and determine proofs of location by votes to reach a consensus (Foamspace-Corp, 2018).

Another example of a geospatial blockchain application is D-GIS, a blockchain infrastructure designed to share geospatial information. It was developed to permit scientists working with geospatial data to share their work securely. The same as FOAM, D-GIS is based on the Ethereum network and it promotes ownership rights and consensus among participants. The concept behind D-GIS is to promote competition by using a ranking mechanism. Tokens are assigned to participants depending on the size of their contribution. Authors can also vote on the work of other participants, the power of their vote depends on their own ranking. Voting penalizes on its own ranking to avoid the domination of a single user (Leka et al., 2019).

Both of these projects contribute to the development of a geospatial blockchain and promote both decentralization and citizen science. Although D-GIS has been developed conceptually, FOAM is the only one of the two that has been fully deployed. On the one hand, FOAM has extended its functionalities to implement the “Trust Zone”, a program focused on distributing radio devices that pride feedback to improve FOAM Location services (Foamspace-Corp, 2022). On the other hand, D-GIS has no recent updates in its repositories and no application has been publicly published or promoted. Up to this moment, Ethereum is a network whose costs for transactions are very high compared to other networks. This is a relevant topic that will be addressed in this work.

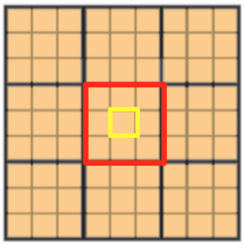
2.2 InterPlanetary File System

Given the architecture described in section 1, resources for storing data on the blockchain are limited (Xu et al., 2017). This represents a high cost both from a monetary and from a time perspective, compared to other systems like cloud storage. Therefore, storage of files or data not belonging to the transaction information should be avoided. For this reason, decentralized applications need a solution to keep data safely. Since a centralized storage system would compromise the benefits provided by the blockchain, developers started to implement decentralized storage platforms for the purpose. One of the most widespread is the InterPlanetary File System (IPFS) (Benet, 2014).

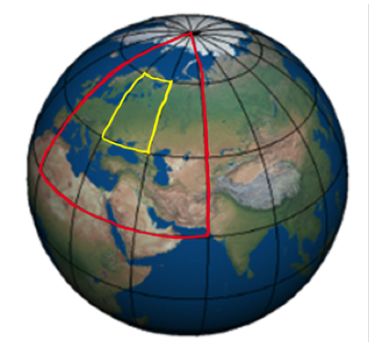
The IPFS is a peer-to-peer (P2P) file system which combines successful ideas from architectures such as BitTorrent (Inc., 2022) and Git (Git-SFConservancy, 2022). Once a file is uploaded to the IPFS, it is duplicated and saved locally by multiple peers that will return it on request. The stored files are permanent in the system as long as duplicates of it exist. To guarantee the reliability of the network and avoid loss of data, the nodes are incentivized to continuously share and store the files. Each file in the IPFS is indexed with the multihash of its content, which can be used by users to retrieve it. A multihash is similar to the hash concept explained in section 1, but with the inclusion of additional data that can make it more informative (Bolger, 2019). Therefore, a solution to store large files in decentralized applications development is to save only its hash on the blockchain, and the content on the IPFS.

2.3 Discrete Global Gridding Systems

The objective of geocoding is to associate a feature, an event or an observation to a specific location on the Earth's surface. For such a purpose, hierarchical gridding systems are used to partition an area (in this case the Earth) into cells. An option to do this is to use conventional spatial reference systems such as WGS84 or Web Mercator. These systems use projected spherical Cartesian coordinate axes to create rectangular planar grids. However, when projecting spherical coordinates to a plane some limitations exist. The first is that in planar grids (figure 1(a)), the shape of the parent cell (red) is equal to the shape of the child (yellow). Instead for spherical systems (figure 1(b)), the shape of the parent cells (red) differ from that of the child cell (yellow). In this representation, the yellow area has a different geometry from the red one. This would not happen in a planar system, where both parent and child areas have the same geometries. This in turn affects the transformation from spherical to planar coordinates, where the geometrical properties are modified. This means that either the original shape or area are preserved, but never both (Purss, 2017).



(a) Planar gridding



(b) Spherical gridding

Figure 1. Geocoding for planar and spherical systems. Copyright © 2017 Open Geospatial Consortium, Inc. All Rights Reserved. http://www.ogc.org/ogc/document

To solve this difficulty, one possibility is to use Discrete Global Grid Systems (DGGSs). DGGSs are area-preserving reference systems, that hierarchically partition the Earth into cells (e.g. hexagons or triangles) (Sahr et al., 2003). These maintain the accuracy and precision of spatial data at all scales. Since cells in DGGSs have homogeneous geometries and spatial properties, each cell has an equal probability to contribute to an analysis. DGGSs are designed to be hierarchical gridding systems. Here, the initial resolution can be increased iteratively by reducing the size of the cells by means of a chosen mathematical method. As a result, the system will have finer resolution child cells derived from the original (parent cells). The method for creating the tessellations and refining the cell resolution depends on the chosen DGGS (Purss, 2017).

The OGC is an organization in charge of creating geospatial standards with the objective of making information Findable, Accessible, Interoperable and Reusable (FAIR). Their standards are characterised by being free, public and open. For this reason, the OGC has developed DGGS standards to encourage the interoperability of the systems. The OGC defines that the DGGSs must emphasise the three following aspects (Purss, 2017):

1. equal-area cells: hierarchical equal-area tessellations (cells) which at multiple resolutions have an equal probability of contributing to the analysis;
2. full Earth coverage: the DGGS must be defined over the entire surface of the Earth;
3. functions and operations for computations: A DGGS' capacity to identify and conduct algebraic operations on the data assigned to it.

In the following section, the S2 DGGS will be briefly described. This was the selected geocoding system for this work. It is an open-source library developed by Google that offers processing functionalities and a grid with a fine-grained resolution.

Each DGGS is characterized by features that distinguish each system from the other. The most relevant characteristics to be discussed are:

1. cell shape: the shape that the cells have at all hierarchical levels. This affects the efficiency of the indexing algorithm and the degree of complexity of the network;
2. indexing system: this is the system that the geocoding algorithm uses to generate the cell addresses. According to OGC standards, four general indexing systems currently exist: hierarchy-based, space-filling curve based, coordinate and encoded address schemas (Purss, 2017);
3. tesselation method: the procedure used internally by the DGGS to perform the cell refinement to increase or decrease the system's resolution;
4. textual representation: this is the method used to generate the string that will represent the geocoded location.

2.3.1 S2 is a DGGS library developed and used to improve the robustness and performance of planar geometry libraries. This system works with spherical projections to reduce the area and geometry distortion. The point mapping assumes the Earth as being a perfect sphere. This consideration achieves a maximum distortion of 0.56%, with no singularities or discontinuities. The algorithm uses a spherical and not an ellipsoidal projection to have faster computations, in fact, processing under this projection is orders of magnitude faster than other DGGSs that use ellipsoidal projections (S2Geometry, 2022a).

The S2 cells defined in this framework don't use planar bounds to define the top-level hierarchy but instead project the six faces of a cube into the unit sphere. Hierarchical tessellations are then performed by dividing each cell into four child cells, using spherical projections (figure 2). To assign an Identification number to each cell, the S2 algorithm uses the fractal space-filling curve, where six Hilbert curves (one per face of the projected cube) are linked together. Regarding the textual representation, the S2 DGGS uses the S2CellId naming algorithm(S2Geometry, 2022b).

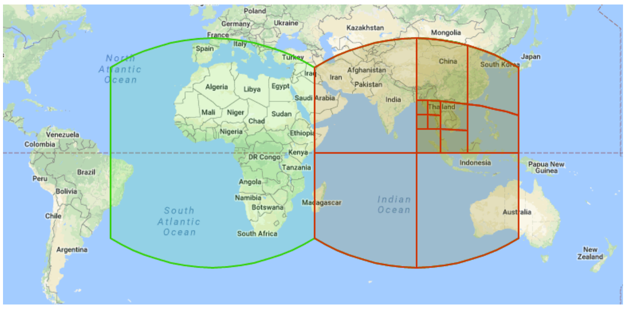


Figure 2. S2 DGGS visual representation. (S2Geometry, 2022b)

3. BLOCKCHAIN ARCHITECTURE

The main principle of the architecture implemented in this work is to map discrete portions of the Earth's surface (DGGS cells) to addresses in the blockchain that can store data relative to that cell. This design is inspired by the framework Crypto Spatial introduced in (Daho, 2020), but we applied different modifications to make it more suitable for the open data sharing use case. As mentioned in section 2.3 the chosen DGGS was S2 and the system implementation is developed in the Solidity programming language. This allows it to be used on every blockchain that supports the Ethereum Virtual Machine (EVM) (Buterin, 2014) and guarantees extended flexibility. Moreover, this choice is justified by the expanded ecosystem that Ethereum offers. The Smart Contracts are completely open-source and developed with a focus on the reusability of the components for other applications in the same field.

3.1 Smart Contracts design

The two main parts of the architecture are the Cell Smart Contracts and the Registry Smart Contracts. For each of the parts, we developed a generic abstract contract (CellContract and RegistryContract) that is suitable in different use cases and a more specific contract that expands the basic implementation (SimileCell and SimileRegistry).

The Registry Contracts are characterized by a resolution of the DGGS and are used to keep track of all the DGGS cells added to the system and their relative contract address. The Cell Contracts are instead used to maintain a list of 32 bytes hashes, that represent the files inserted in the system. These hashes are used to retrieve the complete files that are stored in a distributed file system, such as IPFS.

Moreover, in the cell contracts, it is possible to store additional data relative to each hash. In particular in the case of SimileCell Contracts:

- the address of the owner of the file
- the votes for the file
- the address of the voter for each vote

The cells contain functions to add, remove and update hashes, and vote existing observations. The removal and update of a hash are only allowed to the owner of a file.

All the data structures used in both contracts follow the storage patterns introduced in (Hitchens, 2017), which allow direct access to all data. These patterns involve some redundancy in the data but guarantee consistency and remove loops, which are too costly in Solidity, especially for dynamic arrays.

Figure 3 and 4 show in detail the class diagram of all the contracts.

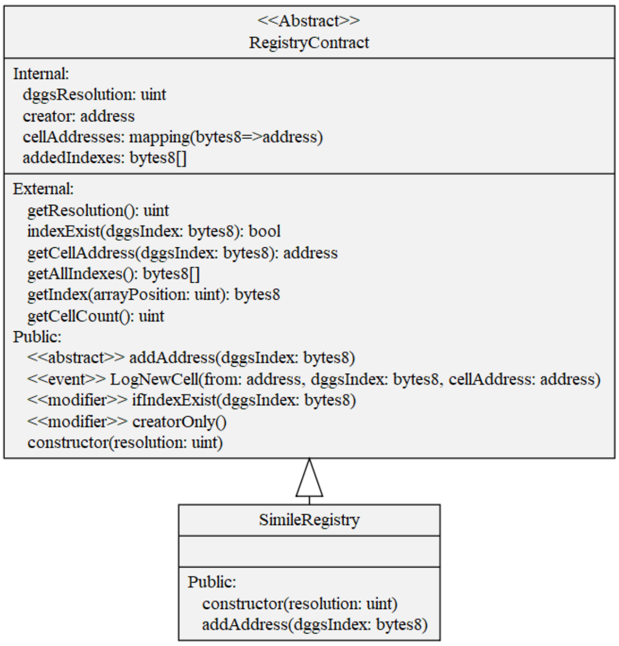


Figure 3. SimileRegistry class diagram.

3.2 Contracts events

Each function that modifies the state of a contract emits an Event with the information of the operation, such as the address that executed it and on which data. The Events are useful for two different reasons:

- an outside system can listen for them and use them to update the information that it has on the state of the contract without querying the entire state again (which can be a long operation);
- to keep track of the deletion or update on the hashes of the system. This is particularly useful in the case of updates, as the old hashes are not stored in the contract anymore, but through the events it is possible to retrieve the files from the distributed storage system, effectively creating a history of updates without storing them all.

3.3 Contracts flexibility

While the proposed architecture is implemented specifically for an application where users have control over their files and each file can be voted by other users, the contracts are designed to allow their usage, with few code changes, in similar use cases with different requirements.

Firstly, we implemented function modifiers to allow changes to the state only by the creator of the contract, enabling the use in a more controlled environment where a single entity (such as a server) performs all actions on behalf of the users.

Secondly, the cell contracts are suitable to store any kind of data relative to each file hash instead of the user votes.

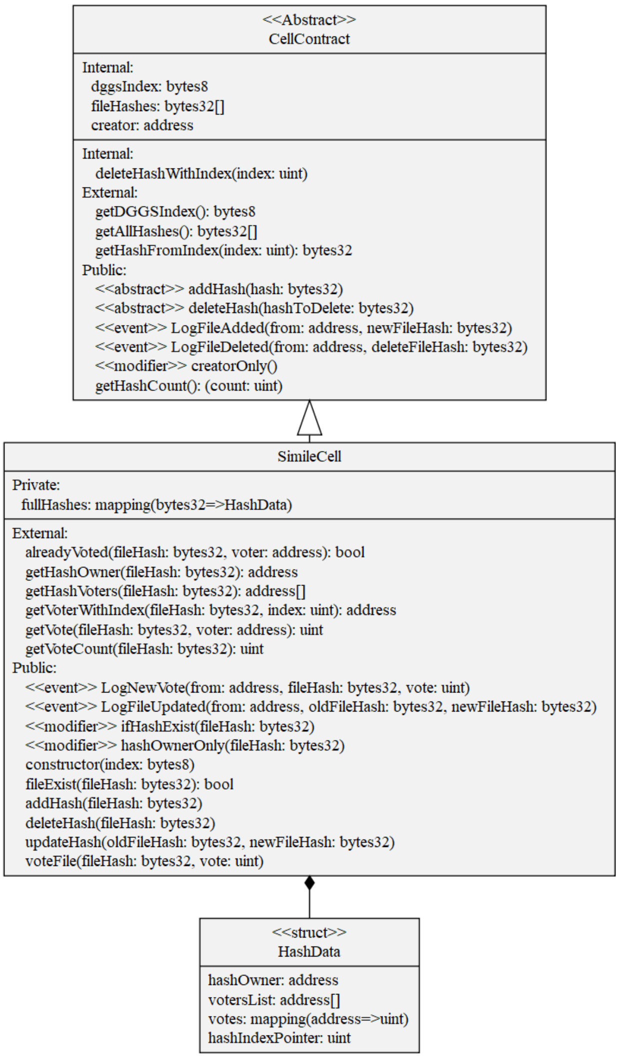


Figure 4. SimileCell class diagram.

Finally, the same contracts could be used in applications where multiple DGGS resolutions are needed, since that would only require the deployment of more registries, once per resolution.

4. WEB DECENTRALIZED APPLICATION PROTOTYPE

Along with the blockchain architecture, we developed a prototype web application that interacts with all the components previously described and demonstrates how it is applicable to a citizen science context. The use-case for this work was the SIMILE project. The complete implementation of the prototype is available in the project GitHub repository (https://github.com/rodrigocedeno/simile-blockchain)

4.1 Implementation overview

The contracts for the prototype are deployed on Velas (VELAS, 2021), a blockchain system with a strong focus on high transaction speed and low cost of fees compared to other blockchains (e.g. Ethereum, Cardano, Solana).

The prototype is used to simulate the environment of the SIMILE system without reimplementing the entire application. All the observations are represented as GeoJSON files, which are used to store information in the SIMILE system. Different types of files such as different formats or multimedia files could be easily stored. Since all the observations refer to a point and not surfaces of different areas, the system uses a single registry with a fixed resolution.

To interact with the web app, a user is required to provide its EVM wallet address, and the relative private key to perform transactions. In a real application, the wallet and the key should be managed by a wallet manager, such as Metamask(MetaMask and ConsenSys Formation, 2022).

4.2 Application functionalities

The application is split into three main sections:

- Add new observations
- Validate observations
- Manage personal observations

In the first section, a user can add an observation to the specified coordinates, that are converted to the corresponding S2 index. If the index is not already present in the registry, the system deploys a new cell contract for that location. Since the cells are only deployed when needed, we avoid adding useless contracts to the blockchain and to the registry. The file is instead encoded into its SHA-256 hash and is added to the cell. Once the transactions are successfully executed, the entire file is uploaded permanently to the IPFS and can be quickly retrieved using its multihash. The flow of this operation and the interactions among the components are shown in figure 5.

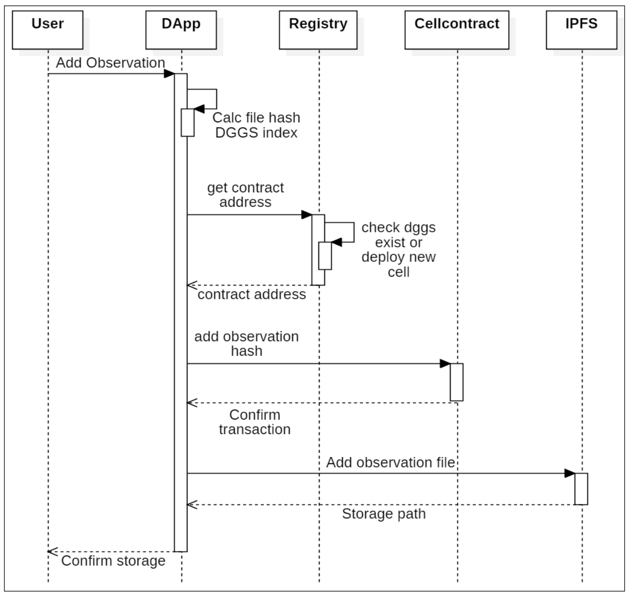


Figure 5. Add observation sequence diagram.

The second and third sections show respectively the list of all the observations and the ones added by the current user. To guarantee simple integration with the mapping interface, all the file hashes are connected with their coordinates or S2 index. To reduce the time the system needs to show all the observations, the entire file and its votes are loaded only after it is selected in the interface. In the validation section (figure 6), once an observation is selected, the user has the possibility to send his positive or negative vote. In the personal section, instead, he can delete the observation from the system, or update it by uploading the new file.

Figure 6. SIMILE app observations validation section.

5. TRANSACTIONS FEES DATA

To prove the economic sustainability of a similar system in a real-world scenario, in table 1 we report the cost of each transaction that is possible to perform in our prototype. All the values refer to the first operation of each kind, which is slightly more expensive due to memory allocation of new data structures. Along with the fees of the transactions on Velas blockchain, we report the ones on Ethereum to make a comparison with the most widespread platform for decentralized applications.

Since the fee of a transaction is not fixed but is dependent on the cryptocurrency value, and on gas cost as well in the case of Ethereum, we took fixed values on 03/05/2022 and used them for all the estimations. In particular, the values used are:

- 1 VLX = €0.1723
- 1 ETH = €2708.62
- ETH Gas Price = 60 gwei

6. CONCLUSIONS

This work demonstrates the feasibility to develop a geospatial blockchain application to be used as a citizen science tool.

Operation	Velas fee	Ethereum fee
Registry deployment	€0.001305	€369.63
Cell deployment	€0.00085	€240.95
Add observation hash	€0.000046	€13.55
Add observation hash to new index (cell deployment + add hash)	€0.000896	€254.50
Update observation hash	€0.000036	€12.70
Delete observation hash	€0.000015	€4.74
Vote observation	€0.000047	€14.35

Table 1. Transaction fees for Velas and Ethereum.

Given the high costs of using the Ethereum network as a basis for the implementation, other options were explored and the Velas infrastructure was chosen. The reason was that Velas focuses on transaction speed and low fees, making it ideal for a crowdsourcing implementation. The backend infrastructure consists of a connection with the blockchain network, this could be successfully coupled with a Nuxt frontend application with client-side rendering (CSR). The IPFS proved to be a reliable decentralized storing system which can be connected to the blockchain. Utilizing this tool enables a decentralised application to store large files that otherwise would not be possible to store in the blockchain.

Future developments will focus both on improving the current prototype and on implementing the same blockchain architecture with different degrees of decentralization and types of data (i.e. not only data points). A significant improvement of the application would consist in the implementation of a reward system. As demonstrated in (Lotfian et al., 2020), extrinsic motivation such as a monetary reward motivates users' participation in crowdsourcing applications. Examples of this can be seen in other blockchain applications (Protocol Labs, 2017), where a utility token promotes the citizens' participation.

7. ACKNOWLEDGMENTS

The paper presents the research carried out within SIMILE project (informative System for the Integrated Monitoring of Insubric Lakes and their Ecosystems, ID: 523544), which is funded by the European Commission in the frame of Interreg Italy-Switzerland 2014-2021 program.

REFERENCES

Alabdulwahhab, F. A., 2018. Web 3.0: the decentralized web blockchain networks and protocol innovation. *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, IEEE, 1–4.

Benet, J., 2014. IPFS - Content Addressed, Versioned, P2P File System. *CoRR*, abs/1407.3561. http://arxiv.org/abs/1407.3561.

Biraghi, C. A., Lotfian, M., Carrion, D., Brovelli, M. A., 2021. AI IN SUPPORT TO WATER QUALITY MONITORING. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2021, 167–174. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B4-2021/167/2021/.

Bolger, K., 2019. How Geoblockchain Could Curb 'Conflict Diamond' Trade.

Bratic, G., Carrion, D., Cannata, M., Rogora, M., Strigaro, D., Brovelli, M. A., 2022. LAKE WATER QUALITY MONITORING TOOLS. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2022, 599–606. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B4-2022/599/2022/.

Brovelli, M. A., Cannata, M., Rogora, M., 2019. SIMILE, A GEOSPATIAL ENABLER OF THE MONITORING OF SUSTAINABLE DEVELOPMENT GOAL 6 (ENSURE AVAILABILITY AND SUSTAINABILITY OF WATER FOR ALL). *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W20, 3–10.

Brovelli, M. A., Ponti, M., Schade, S., Solís, P., 2020. Citizen Science in Support of Digital Earth. H. Guo, M. F. Goodchild, A. Annoni (eds), *Manual of Digital Earth*, Springer Singapore, Singapore, 593–622.

Buterin, V., 2014. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. 36.

Carrion, D., Pessina, E., Biraghi, C. A., Bratic, G., 2020. CROWDSOURCING WATER QUALITY WITH THE SIMILE APP. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2020, 245–251. https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B4-2020/245/2020/.

Daho, A., 2020. CRYPTO-SPATIAL: AN OPEN STANDARDS SMART CONTRACTS LIBRARY FOR BUILDING GEOSPATIALLY ENABLED DECENTRALIZED APPLICATIONS ON THE ETHEREUM BLOCKCHAIN. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B4-2020, 421-426.

Foamspace-Corp, 2018. FOAM Whitepaper.

Foamspace-Corp, 2022. FOAM Trust Zone. https://foam.space/.

Git-SFConservancy, 2022. Git. https://git-scm.com/.

Hitchens, R., 2017. Solidity CRUD. https://medium.com/robhitchens/solidity-crud-part-1-824ffa69509a.

Inc., B., 2022. BitTorrent — Il client Torrent più popolare al mondo. https://www.bittorrent.com/it/.

Karadsheh, L., Alryalat, H., Alqatawna, J., Alhawari, S. F., Jarrah, M. A. A., 2022. The Impact of Social Engineer Attack Phases on Improved Security Countermeasures: Social Engineer Involvement as Mediating Variable. *International Journal of Digital Crime and Forensics (IJDCF)*, 14(1), 1–26.

Leka, E., Lamani, L., Selimi, B., Deçolli, E., 2019. Design and Implementation of Smart Contract: A use case for geo-spatial data sharing. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1565–1570.

Lone, A. H., Naaz, R., 2021. Applicability of Blockchain Smart Contracts in Securing Internet and IoT: A Systematic Literature Review. *Computer Science Review*, 39, 100360.

Lotfian, M., Ingensand, J., Brovelli, M. A., 2020. A Framework for Classifying Participant Motivation that Considers the Typology of Citizen Science Projects. *ISPRS International Journal of Geo-Information*, 9(12). https://www.mdpi.com/2220-9964/9/12/704.

Majid, I., Verma, S., 2018. Web 2.0 Technological Tools in Teaching and Learning. *International Journal of Education*, 10, 6.

MetaMask, ConsenSys Formation, 2022. The crypto wallet for Defi, Web3 Dapps and NFTs | MetaMask. metamask.io.html.

Nakamoto, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. 9.

Nikolić, I., Biryukov, A., 2008. Collisions for Step-Reduced SHA-256. K. Nyberg (ed.), *Fast Software Encryption*, 5086, Springer Berlin Heidelberg, Berlin, Heidelberg, 1–15.

Nofer, M., Gomber, P., Hinz, O., Schiereck, D., 2017. Blockchain. *Business & Information Systems Engineering*, 59(3), 183–187.

Protocol Labs, 2017. Filecoin: A Decentralized Storage Network. https://filecoin.io/filecoin.pdf.

Purss, M., 2017. Topic 21: Discrete Global Grid Systems Abstract Specification. https://docs.opengeospatial.org/as/15-104r5/15-104r5.html#11.

Riordan, M., 2022. Going Global: The World the Web Has Wrought. *Physics World*, 35(1), 41.

S2Geometry, 2022a. About S2Geometry. http://s2geometry.io/about/overview.html.

S2Geometry, 2022b. S2 Cells. http://s2geometry.io/devguide/s2cell_hierarchy.html.

Sahr, K., White, D., Kimerling, A. J., 2003. Geodesic Discrete Global Grid Systems. *Cartography and Geographic Information Science*, 30(2), 121–134.

Vavassori, A., Biraghi, C. A., Bratic, G., Carrion, D., Zamboni, G., Brovelli, M. A., 2021. Citizen Science Tools for Lake Monitoring in the Framework of the United Nations Sustainable Development Goals: The Project SIMILE. *GI_Forum*, 1, 179–186. https://hw.oeaw.ac.at?arp=0x003c9b72.

VELAS, 2021. Velas Whitepaper v1.0. https://velas.com/pdf/whitepaper.pdf.

Vogel, N., 2015. The great decentralization: How web 3.0 will weaken copyrights. *J. Marshall Rev. Intell. Prop. L.*, 15, 136.

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., Rimba, P., 2017. A taxonomy of blockchain-based systems for architecture design. *2017 IEEE international conference on software architecture (ICSA)*, IEEE, 243–252.

Zarrin, J., Wen Phang, H., Babu Saheer, L., Zarrin, B., 2021. Blockchain for Decentralization of Internet: Prospects, Trends, and Challenges. *Cluster Computing*, 24(4), 2841–2866.