WEB3-BASED ROLE AND TOKEN DATA ACCESS: THE CASE OF BUILDING MATERIAL PASSPORTS

Jens J Hunhevicz[1,2], David F Bucher[1], Ranjith K Soman[3], Meliha Honic[1], Daniel M Hall[3] and Catherine De Wolf[1]
[1]ETH Zurich, Zurich, Switzerland
[2]EMPA, Dübendorf, Switzerland
[3]TU Delft, Delft, The Netherlands

Abstract

Long-term access to lifecycle data is key to a successful transition towards a circular built environment. However, the underlying technology often remains centralized and risks becoming inaccessible over time. In this paper, we investigate whether decentralized access methods using Web3, i.e. blockchain and decentralized data storage protocols, can help to mitigate this limitation. We implement Web3 data access mechanisms for a material passport with both a role-based and a token-based smart contract. Initial results suggest that Web3 offers a promising approach to data access over the lifecycle of a built asset - but only with careful design choices.

Introduction

A transition from a linear to a circular economy is urgently needed to reduce resource depletion and greenhouse gas emissions caused by the construction industry. Material passports, which are digital datasets containing valuable information about materials and products (Luscuere and Mulhall, 2018), have been identified as a key enabler of a circular economy (Çetin et al., 2021). However, the construction industry is characterized by long lifetimes, typically 30-100 years for residential buildings (Swisslife, 2017), and by the involvement of a large number of actors in an extremely fragmented supply chain (Sheffer, 2011). This raises the question of how to avoid loss of information or access to material passports throughout the life cycle of buildings. This is because Web2, the Internet as we know it today, relies heavily on centralized data access management. In such a system, the risk of data or access loss during the lifecycle of a built asset is high, as companies and server operators could go out of business, or actors could change and lose access to previous storage locations.

One possible way to address this could be Web3 - the vision to decentralize the internet (Ethereum, 2022). Web3 uses blockchain at its core to give data ownership back to the end user (Gajria, 2021). Combined with decentralized data storage protocols, the vision is to store data without relying on gatekeepers (Web3-Foundation, 2022). A key aspect of Web3 data ownership is the new forms of access control to data using decentralized mechanisms. It could enable access control to data independent of individual actors over long periods of time. Despite this potential, to our knowledge there is no research that investigates the feasibility and promise of Web3 access control for lifecycle data in the built environment.

Therefore, in this paper we provide an introduction to Web3-based data access mechanisms. We implement a prototype for material passport data to assess the feasibility and suitability of such an approach. In doing so, we demonstrate how blockchain and decentralized data protocols can be used together to control data access in the built environment.

In the background section, the paper reviews the difference between Web2 and Web3 access control, followed by the identification of possible Web3 data access mechanisms for implementations in the AEC industry. In addition, we briefly review existing research on blockchain, tokens, and decentralized data storage protocols in the AEC industry.

In the second part, we develop and test a prototype implementation for an exemplary use case of access to a material passport, with both an address- and token-based access mechanism using the *Stacks* blockchain and *Gaia* decentralized data storage.

Finally, we discuss our findings from the implementation, both in terms of technical aspects and possible implications of the proposed use case.

Background

Web2 vs. Web3 Access Control Stack

We based our work on the description of the Web3 technology stack by the Web3-Foundation (2022). We then simplified the technology stack to the components relevant to this paper, as shown in Figure 1 and described below.

Both Web2 and Web3 rely on the internet to connect the various components. On the client side, the user connects over the internet to a front-end using a browser.

While in Web2 the front-end communicates with a back-end server to authenticate users, in Web3 the front-end connects to a blockchain node, a computer that maintains a copy of the blockchain, to access the smart contracts with the access control logic. Smart contracts are scripts that run on the blockchain and can encode logic that interacts with transactions. The details of how blockchain and smart contracts work are beyond the scope of this paper, and can be found in Hunhevicz and Hall (2020). The key point is that while traditional back-end servers are owned and operated by intermediaries, a blockchain is a decentralized (i.e., not controlled by a single actor) network of computers that maintains a shared and transparent database. This database consists of transactions between different users of the blockchain. A transaction can be a payment between users or, as in the case relevant to this paper, a request for

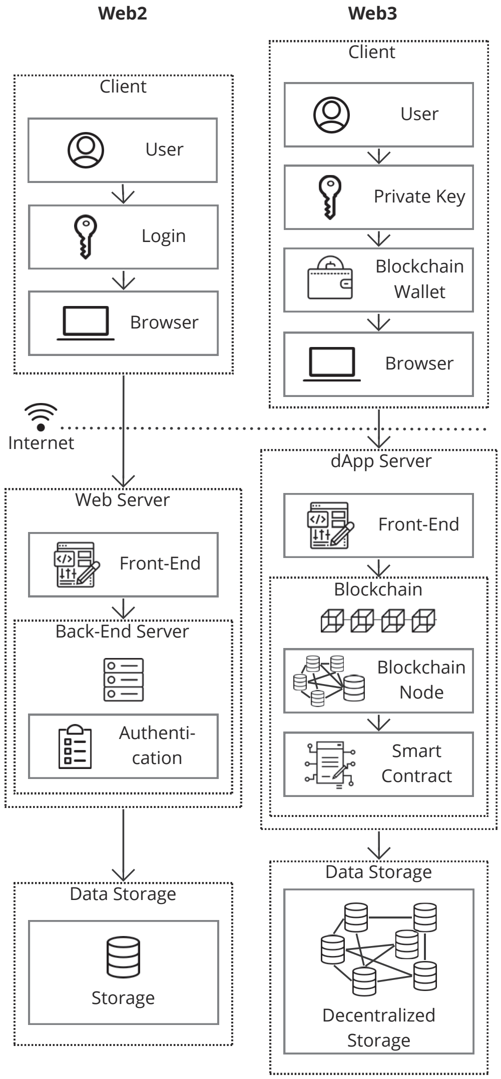


Figure 1: Data access: Web2 versus Web3 technology stack.

access to data. The consensus rules of the blockchain ensure that double spending and the removal of valid transactions are not possible. The end result is that users do not need to trust any intermediaries to facilitate access control, but can rely on the "trustless" smart contracts running on the blockchain. Trustless in the sense that they execute transaction logic transparently and as coded. Overall, the combination of a front-end that communicates with smart contracts on the blockchain is called a "dApp" - a decentralized application.

The difference in the backends used to authenticate users also has client-side implications. While Web2 requires users to enter valid credentials (usually email and password) that match those stored on the back-end server, Web3 requires the user to control a valid private key to facilitate transactions on the blockchain and interact with the smart contract. This private key is never shared with any other stakeholder or backend server and typically managed by a personal blockchain wallet connected to the browser, which can sign transactions on the user's behalf.

Finally, in Web2, the back end forwards the access request to the servers that store the data. Often these are data centers owned and operated by institutions (often referred to as "the cloud"). In contrast, the Web3 approach relies on so-called data distribution protocols to store data. They store data in a decentralised and sometimes distributed manner on networked servers that can be set up and connected to the network by any user at any time. The best known implementation is the Interplanetary File System (IPFS) (Benet, 2014). Although it is possible to store data on a blockchain, it is more efficient to store data in an off-chain store and only reference the unique identifiers (UIDs) on the blockchain. This way, there is no transaction fee for storing data, no waiting time for transactions to settle, and data can also be deleted or modified. Verification of data integrity is still possible through the unique UIDs.

Web3 Access Control

The main technology that facilitates access control is blockchain, which uses addresses to identify users. A user can prove ownership of an address by signing a transaction with the associated private key. In this paper, we discuss two Web3 access control approaches: role-based and token-based. They are described below and illustrated in Figure 2.

Role-based: With role-based access, a "role" can be assigned to an address by storing the connection in a hash table of the smart contract store, called a mapping. When a user calls the *accessData()* function with an address, *accessData()* checks whether the address has the correct role assigned to access the UID in the mapping (see Figure 2: Does AddressA hold RoleA?). Role-based access has the advantage of allowing fine-grained control over which addresses are allowed to access a UID. Also, access can be removed from an address by updating the role entry for that address. Also, access is not transferable between addresses. However, role-based access has a disadvantage in that the authorized user must add or remove individual addresses in the smart contract each time there is a change.

Token-based: In token-based access, the smart contract maps an "access token" to a UID and checks whether an address calling the *accessData()* function is the owner of the correct token (see Figure 2: Does AddressA own TokenA?). The two main types of tokens used on blockchains are either fungible (FT), meaning all tokens are the same, or non-fungible (NFT), meaning each token is unique. Both types of tokens can be used to hold access rights. A token is created on the blockchain with a smart contract as a value container that can be sent to other users. Unlike role-based access, token-based access makes access rights portable. A user can send the token to another address and the new address will have access rights. This makes access more flexible and scalable without having to manage access rights for each address in the smart contract. In addition, a user can prove access rights by holding the token

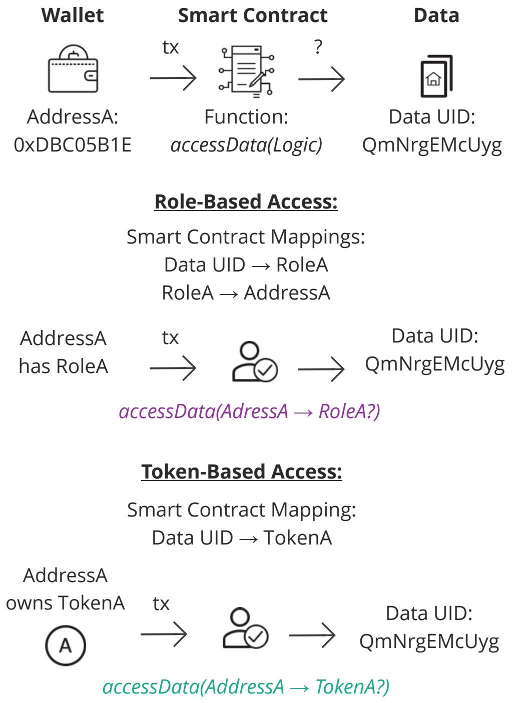


Figure 2: The logic of role-based (purple) and token-based (green) Web3 data access. The colors correspond with the respective workflows in Figure 4.

without having to perform a transaction. However, a disadvantage of token-based access is that it is more difficult to control access at the individual address level, because it is not possible to enforce the removal of a token from an address.

Blockchain, Decentralized Data Storage Protocols, and Tokens in the AEC industry

Blockchain is gaining traction in the construction informatics research domain. Proposed applications range from blockchain as an assurance layer, new forms of economic mechanisms such as incentives and markets, all the way to new forms of decision-making and ownership (Hunhevicz et al., 2022). Nevertheless, there is to our knowledge no work solely focusing on blockchain data access mechanisms through roles and tokens.

However, there is early work exploring the combination of blockchain with decentralized data storage protocols for construction data management. Tao et al. (2021) proposes the use of blockchain with IPFS for BIM-based collaborative design. Also, Hamledari and Fischer (2021) store field reality capture data in IPFS to keep data off the blockchain and only link the CIDs with the smart contracts. Das et al. (2022) propose a blockchain-based document management system, also mentioning IPFS as a potential storage location. Finally, Darabseh and Martins (2021) highlights the potentially increased reliability of data exchange and immutability of files when using decentralized file systems.

In addition to payments, proposed use cases for tokens in the built environment include tokenized lien rights (Hamledari and Fischer, 2021), ownership of design topologies (Dounas et al., 2021), incentivizing high-quality data sets (Hunhevicz et al., 2020), and token-based investment mechanisms (Tezel et al., 2021; Tian et al., 2020).

Implementation

The Use Case

We implement Web3 data access for the exemplary use case of lifecycle access to material passport data. For demonstration purposes, we use data snippets from a material passport developed in a previous study for a residential building (Honic et al., 2019). The building consists of five above-ground floors with a gross floor area of $\approx 3780m^2$ and a gross volume of $\approx 12000m^3$. The material passport contains information about the exterior walls, roof, slabs and windows, as well as the accumulation of these elements at the building level.

To enable a circular material and element usage strategy, a stakeholder should have access to the material passport at the end of a building's life cycle. In Web2, there is a risk that only parts of the data will be accessible at the end (see Figure 3). This is because Web2 data access is centralized while the construction is fragmented. Since CDEs are likely to be managed by a number of companies that generate and store the data, there is no guarantee that the CDE will still be running or that the data will still be available at the end of its lifecycle. There is also a high probability that credentials will be lost or stakeholders will disappear.

In contrast, Web3 can decentralize both access and storage (see Figure 3). Although stakeholders use their own keys to access the data, access control is managed by a blockchain smart contract. Therefore, even if individual private keys are lost and nodes cease to operate, the blockchain is likely to persist. With the existence of the blockchain, all the smart contracts and access rules would still be available to give the new stakeholder access to the material passport. Also, since the data is stored in a decentralized data store with redundancy to a server failure, the dApp could still provide access and limit the loss of lifecycle data.

The Prototype

To test the possibility of building a decentralized Web3 access control for the described material passport, we implement a prototype using the Stacks blockchain (Stacks-Network, 2023b) and the Gaia decentralized storage protocol (Stacks-Network, 2023a). Stacks is a public permissionless blockchain that uses the Bitcoin blockchain (Nakamoto, 2008) for its security. It can compile smart contracts that integrate with the Bitcoin transaction ledger (Clarity, 2023). Gaia is the data storage protocol built for the Stacks ecosystem. It allows a user to choose and control a storage location, while a so-called Gaia hub authenticates writes based on an authorized Stacks address. The

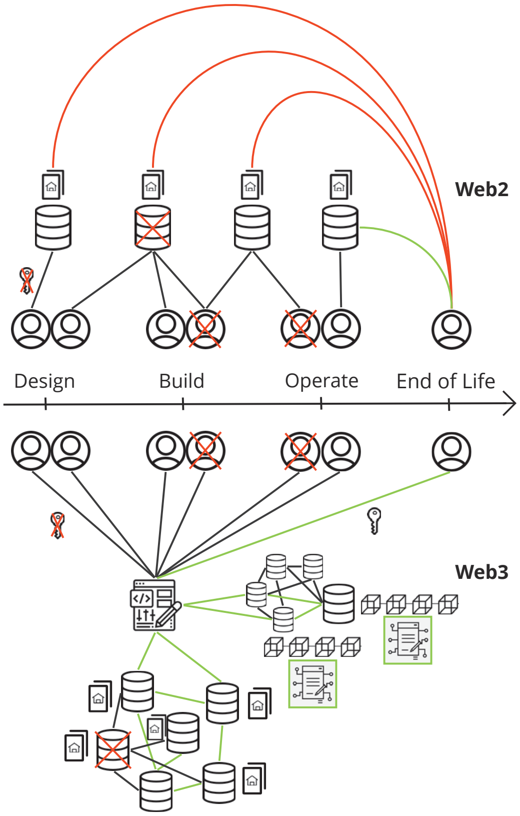


Figure 3: The exemplary use case of material passport access throughout the building lifecycle. The hypothesis is that the risk of information loss could be reduced with Web3 compared to Web2. Component descriptions in Figure 1.

dApp that coordinates access control to Gaia via Stacks is built using React (2023). The interaction between the different components of the technology stack is illustrated in Figure 4. The code and documentation of the prototype is available on GitHub[1].

We first describe how a user can store and access the material passport with Gaia (see Figure 4, gray dots). We have developed a dApp to store data in Gaia on behalf of the user. Users connect their Hiro wallet (Hiro, 2023a) to the dApp. The Gaia Hub performs authentication and allows writes if the authentication is successful with a valid Stacks account. Authentication with the selected account (via Hiro) passes the URL of the user's Gaia hub to the dApp. For example, we uploaded the material passport file to the selected Gaia hub after authenticating with our Stacks account in the dApp (see Figure 5). This makes the file accessible under a unique URI (in the case of Gaia: a Gaia URL). If the file is not made public during upload, the material passport will be encrypted. In this case, only our account can decrypt the file after successful authentication in the dApp.

We can authorize the dApp to decrypt data stored with Gaia on our behalf in order to share data with other users. Figure 6 shows the material passport file for which we have enabled sharing with the dApp in the implemented prototype. Note that the dApp will only decrypt data for another user if the access control checks are successful. Since authentication is done using a Stacks account, the dApp needs to know which addresses are allowed to access the data. We can define the rules using smart contracts to decentralize the access logic. The user can interact with the smart contracts through the dApp using the sharing control page in the prototype (see Figure 7). Each button in the sharing control page triggers a smart contract function that must be signed with the authenticated account. The next two sections describe the role-based and token-based access control mechanisms implemented in the prototype.

Role-Based Material Passport Access

The role-based smart contract implements the role-based access logic described earlier in this paper and shown in purple in Figures 2 and 4. The smart contract was coded in Clarity (2023), a scripting language developed for the Stacks ecosystem. The contract code is available in the GitHub repository[2] and has been deployed to the Stacks testnet (Hiro, 2023b).

The contract creates two roles: the *data-owner* role and the *data-access* role. The owner of the data can claim the owner role by navigating to the access control page of the file and executing the *add-data-owner* function. This will associate the owner address with the Gaia URL in a first mapping. The data owner can then register other addresses for the data access role by calling the *add-data-accessor* function, connecting other addresses to the Gaia URL in a second mapping. Anyone with a data access role can access the Gaia URL. In addition, data access roles can be removed by the data owner by calling the *remove-data-accessors* function.

Token-Based Material Passport Access

The token-based smart contract implements the token-based access logic described earlier in this paper and shown in green in Figures 2 and 4. It is available in the GitHub repository[3] and deployed on the Stacks testnet (Hiro, 2023b).

The contract coordinates the creation (so-called *minting*) of two NFTs: the Ownership NFT and the Access NFT. The two NFTs are implemented based on the SIP-009 NFT Clarity standard and customized with two token smart contracts. The data owner can mint the Ownership NFT by executing the function *mint-ownership-nft*. Each Ownership NFT has a unique ID. The contract associates the ID of the minted Ownership NFT with the Gaia URL in a first mapping, because the contract must be able to associate the correct NFT ID with the Gaia URL. The owner of the Ownership NFT is then allowed to mint Access NFTs that can be transferred to other addresses by executing the

[1] `https://github.com/dabuchera/web3-access`

[2] `https://github.com/dabuchera/web3-access/blob/master/contracts/contracts/rolesAccess.clar`

[3] `https://github.com/dabuchera/web3-access/blob/master/contracts/contracts/tokenAccess.clar`

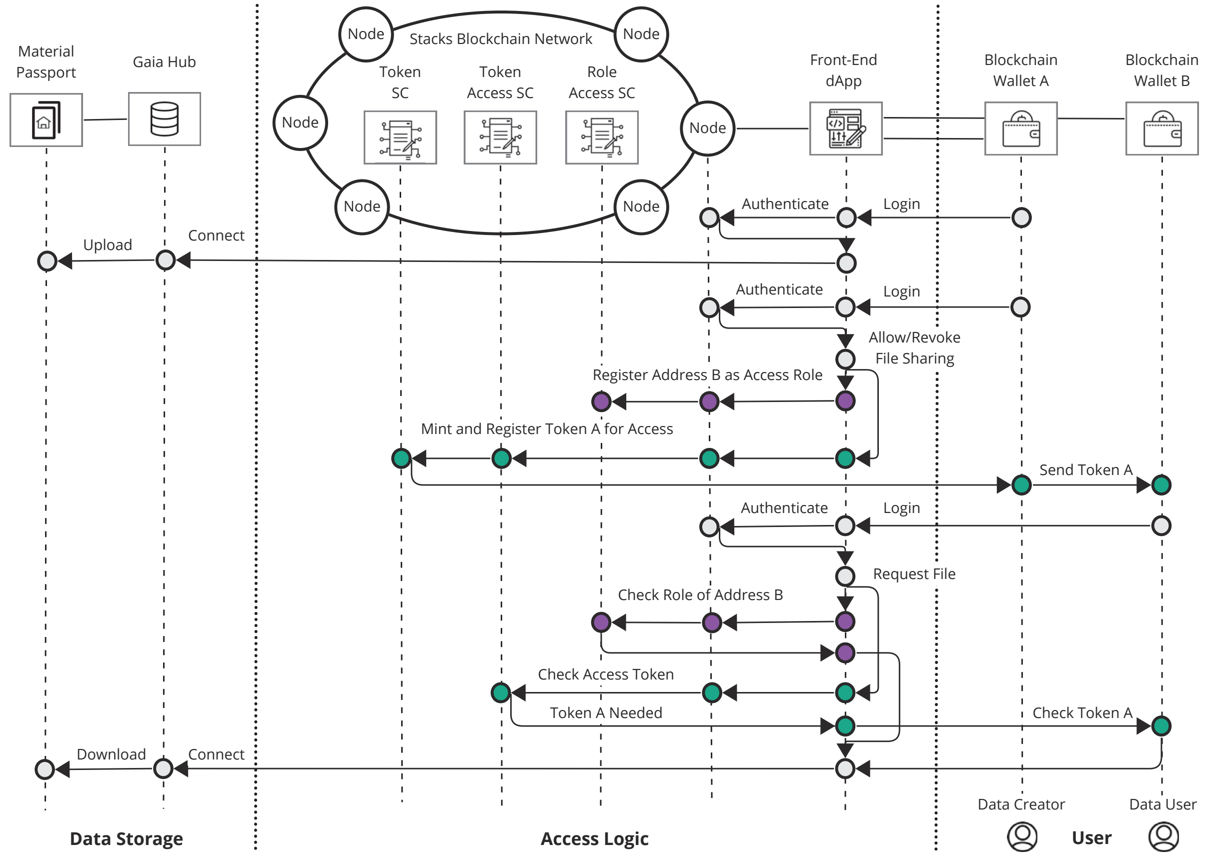


Figure 4: Implemented technology stack and workflow for role-based access (purple) and/or token-based access (green).

transfer function. The contract links the unique IDs of the access-NFTs minted for this Gaia-URL in a second mapping. In this way, the contract keeps track of which tokens have access rights to the file. Although it is not possible to remove tokens from other addresses, it is possible to deactivate access for all minted access-NFTs by calling the *change-access-nft-activation* function.

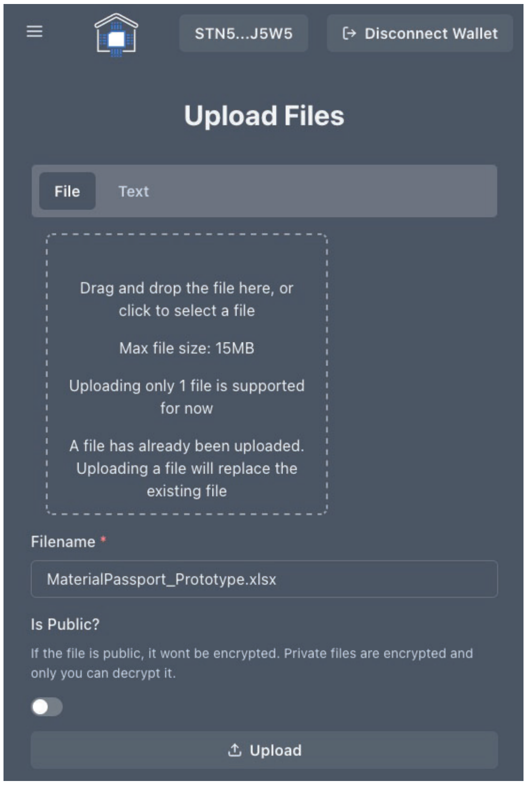
Discussion

Technical Implementation

Our implementation demonstrated that managing data access with Web3 is possible as a proof of concept. The interplay between different Web3 components, such as blockchain smart contracts to manage access control via blockchain addresses along with decentralized data protocols to store files, worked well for this use case. We believe that it is worthwhile for blockchain research in the built environment to further explore the interplay between different technological Web3 components. The obvious advantages over using blockchain alone are a reduced on-chain footprint and therefore reduced transaction costs, the ability to delete files since only the identifier is stored on the blockchain, and the ability to keep files private off-chain while using public blockchains for unrestricted access to the application.

However, there are many open research questions and technical challenges that need to be addressed. A potential disadvantage is that if the off-chain database is no longer available, the established identifiers alone will not help to access the data. Therefore, a challenge is to choose the right blockchain and decentralized data storage among the many options for a Web3 data management use cases. The most common options include Ethereum (Buterin, 2014) together with either Swarm (2021) or IPFS (Benet, 2014). Our decision to implement the prototype using Stacks and Gaia worked well for this prototype, as Gaia allows the user to control the location of storage, and therefore whether or not files should be publicly accessible. This feature formed the basis of our approach to controlling access to files. In IPFS, once a content identifier (CID - the UID in IPFS) is known, anyone can access the data. However, IPFS seems to provide better data distribution from the start (within Gaia the user needs to take care of data distribution), a feature that is also very important for resilient data management throughout the lifecycle. And it may be possible to develop similar privacy control features for IPFS as well.

The second challenge is to design the system to maximize decentralization and avoid points of failure. An early lesson from our implementation is that Web3 technologies do not automatically guarantee decentralization, although



Upload Files

Drag and drop the file here, or click to select a file

Max file size: 15MB

Uploading only 1 file is supported for now

A file has already been uploaded. Uploading a file will replace the existing file

Filename *

MaterialPassport_Prototype.xlsx

Is Public?

If the file is public, it wont be encrypted. Private files are encrypted and only you can decrypt it.

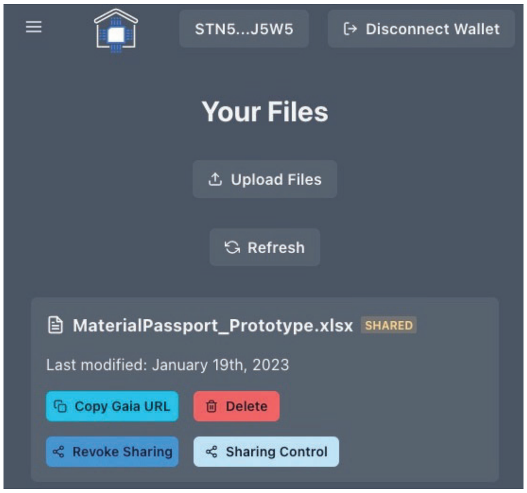
Figure 5: The upload page of the prototype allows to store files and text in Gaia. Authentication is done with a connected wallet that controls the Stacks address.

they have the capability to do so. Just as Web2 systems can take steps to reduce the risk of access and data loss with proper system design, Web3 systems must be well designed to maximize decentralization and eliminate potential single points of failure. For this reason, further research is needed to investigate how best to construct the dApp and Gaia storage network.

Finally, we have not considered in this prototype how to combine Gaia file storage with semantic web and linked data concepts. The linked data concepts may further allow decentralization of data and inference from multiple heterogeneous data stores. This could extend the system to accommodate bots and autonomous agents to improve operations on the data. Therefore, it is possible to extend our prototype to enable machine readability for more convenience in querying and accessing data.

Implemented Lifecycle Data Management

The prototype demonstrated how Web3 can change the lifecycle access management for material passports, allowing for unrestricted and pseudonymous access to files. In addition, the demonstrated role and token-based access implementation showed that it is possible to build a decentralized access logic using smart contracts. The investigated implementation offers promising features for long-term lifecycle data access in the built environment. The mechanisms could ensure flexible and robust data access structures over long periods of time.



Your Files

MaterialPassport_Prototype.xlsx

Last modified: January 19th, 2023

Figure 6: The sample Material Passport file owned by the connected address. Access sharing with the dApp on behalf of the user is enabled.

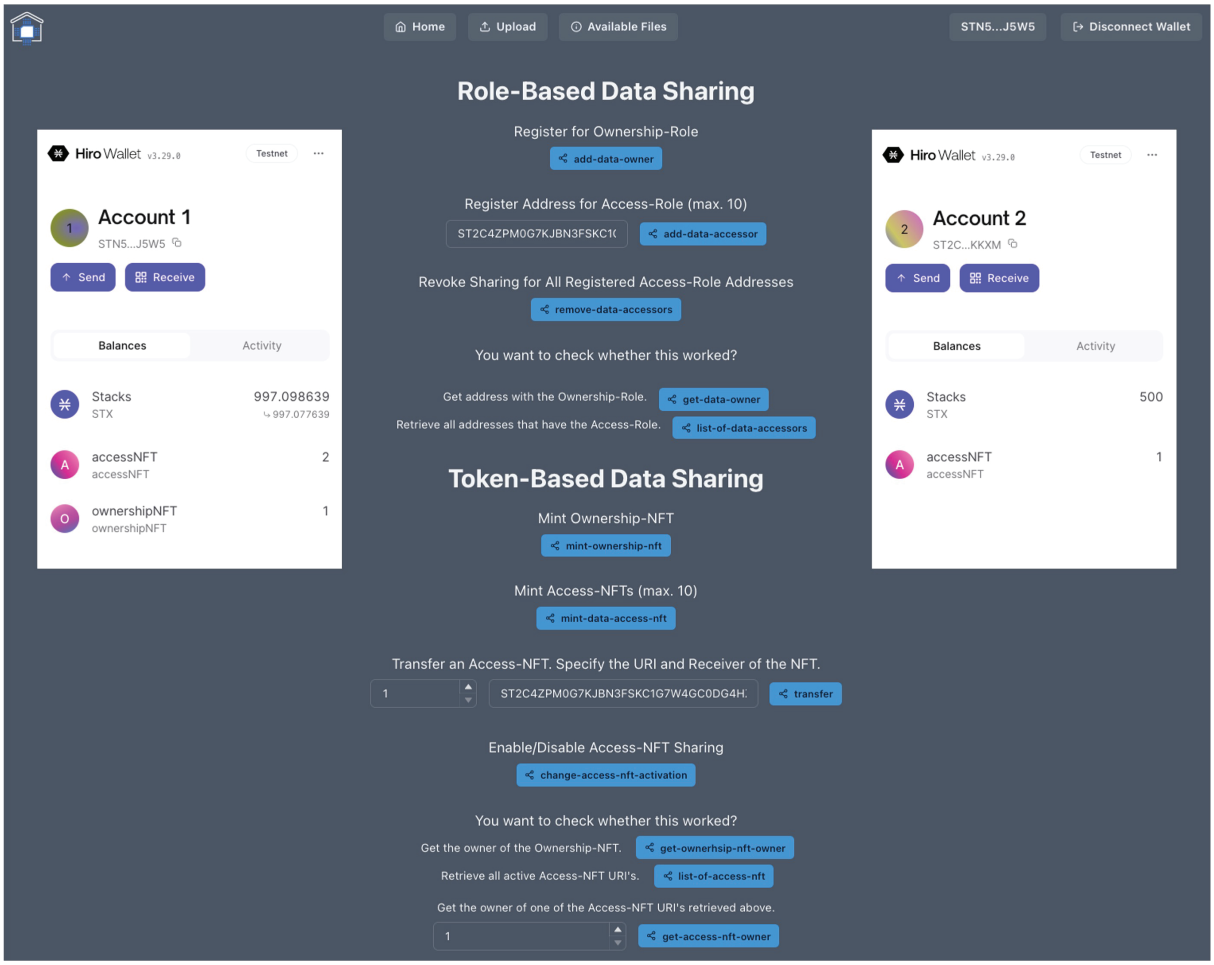
We consider token-based access to be more interesting than role-based access for the use case investigated. The main advantage is that it can be more scalable and flexible over long periods of time. A well-designed mechanism for minting new access tokens could avoid the deadlocks of role-based access distribution. This seems particularly appropriate for the fragmented AEC industry, where stakeholders are likely to change over the lifecycle of a building. Furthermore, after a one-time setup, as many tokens as specified could be minted without assigning a role to each stakeholder.

Combined with data ownership, we can imagine the evolution of the dApp towards a decentralized marketplace where data access can be sold and traded. This could be an additional incentive to participate and share data over the lifecycle of a built asset (Hunhevicz et al., 2020).

However, the adoption of a Web3 data management approach obviously depends on the increasing adoption of Web3 tools for creating secure smart contracts and dApps, as well as the knowledge of how to handle blockchain wallets and private keys to interact with the dApps and smart contracts.

Moreover, the technical implementation must be well designed to provide the promised decentralization, especially the mechanisms for registering ownership of data, the role and token mechanisms, and in general the private key management. If roles responsible for sharing access lose the private key and there is no way to register new participants, the decentralization and distribution of the application is useless or even worse to recover than in Web2, because there is no system administrator who can recover keys. How best to design access mechanisms, e.g. a combination of role-based and token-based access using different token types, should be the subject of further studies.

Finally, more research is needed to investigate the industry requirements for data and data access management over the



Role-Based Data Sharing

Register for Ownership-Role

Register Address for Access-Role (max. 10)

ST2C4ZPM0G7KJBN3FSKC1(

Revoke Sharing for All Registered Access-Role Addresses

You want to check whether this worked?

Get address with the Ownership-Role.

Retrieve all addresses that have the Access-Role.

Token-Based Data Sharing

Mint Ownership-NFT

Mint Access-NFTs (max. 10)

Transfer an Access-NFT. Specify the URI and Receiver of the NFT.

1

ST2C4ZPM0G7KJBN3FSKC1G7W4GC0DG4H:

Enable/Disable Access-NFT Sharing

You want to check whether this worked?

Get the owner of the Ownership-NFT.

Retrieve all active Access-NFT URI's.

Get the owner of one of the Access-NFT URI's retrieved above.

1

Figure 7: The sharing control page for the owner of the Material Passport ("Account 1"). It allows interaction with the smart contract functions by signing transactions with the Hiro wallet. "Account 1" owns the minted Ownership NFT and two minted Access NFTs that have not yet been sent to other addresses. Account 2's wallet owns one Access NFT that was transferred from Account 1. Therefore, "Account 2" can also access the material passport by connecting to the dApp.

lifecycle of the built asset. This would likely influence the design of both the technology stack and the access logic implemented. For example, if open access to data is acceptable, a public protocol such as IFPS combined with an open token-based access mechanism would probably be most appropriate. However, it seems unlikely that stakeholders would share all data openly. In this case, the prototype implemented with Gaia and Stacks is likely to provide better access control.

Conclusions

This conference paper outlined the possible differences between new Web3 access mechanisms and current Web2 approaches. We successfully implemented an exemplary Web3 prototype for the use case of lifecycle data access to a material passport.

After this study, we consider Web3 access control for lifecycle data management as one of the key use cases for technologies like blockchain and decentralized data protocols in the built environment. The many stakeholders involved in the built environment over very long periods of time make a decentralized Web3 approach to data access management interesting. In particular, token-based access mechanisms seem to offer flexible and scalable ways to manage data access.

While the prototype was successful, we discussed the many remaining technical and system design challenges. A key conclusion was that the degree of decentralization and robustness promised by a Web3 application is not automatically guaranteed and depends strongly on the implementation of the system. In particular, the design of the decentralized data storage protocol, the dApp that manages access, and the precise logic for transferring access rights require further research and careful implementation.

However, if well designed, the potential and impact of Web3 data access control for data management in the built environment is promising. It can serve as a foundation for more advanced use cases, such as decentralized data marketplaces for building information, leading to a better database for the transition to a circular built environment.

References

Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System.

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

Çetin, S., De Wolf, C., and Bocken, N. (2021). Circular Digital Built Environment: An Emerging Framework. Sustainability, 13(11):6348.

Clarity (2023). Clarity Smart Contract Language. https://clarity-lang.org.

Darabseh, M. and Martins, J. P. (2021). The Expected Outcomes of Implementing a Distributed File System in the Construction Industry. In Sustainability and Automation in Smart Constructions, Advances in Science, Technology & Innovation, pages 237–242. Springer International Publishing.

Das, M., Tao, X., Liu, Y., and Cheng, J. C. (2022). A blockchain-based integrated document management framework for construction applications. Automation in Construction, 133:104001.

Dounas, T., Jabi, W., and Lombardi, D. (2021). TOPOLOGY GENERATED NON-FUNGIBLE TOKENS - Blockchain as infrastructure for a circular economy in architectural design. In 26th CAADRIA Conference, volume 2, pages 151–160, Hong Kong.

Ethereum (2022). Introduction to Web3. https://ethereum.org/en/Web3/.

Gajria, H. (2021). Web 3.0. https://medium.com/variablelabs/web-3-0-e0d817ec05c6.

Hamledari, H. and Fischer, M. (2021). Construction payment automation using blockchain-enabled smart contracts and robotic reality capture technologies. Automation in Construction, 132:103926.

Hiro (2023a). Hiro Wallet. https://wallet.hiro.so/.

Hiro (2023b). Stacks Explorer. https://explorer.stacks.co/?chain=testnet.

Honic, M., Kovacic, I., and Rechberger, H. (2019). Improving the recycling potential of buildings through Material Passports (MP): An Austrian case study. Journal of Cleaner Production, 217:787–797.

Hunhevicz, J., Dounas, T., and Hall, D. M. (2022). The Promise of Blockchain for the Construction Industry: A Governance Lens. In Blockchain in Construction. Springer.

Hunhevicz, J. J. and Hall, D. M. (2020). Do you need a blockchain in construction? Use case categories and decision framework for DLT design options. Advanced Engineering Informatics, 45(February):101094.

Hunhevicz, J. J., Schraner, T., and Hall, D. M. (2020). Incentivizing High-Quality Data Sets in Construction Using Blockchain: A Feasibility Study in the Swiss Industry. In Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC), pages 1291–1298, Japan (Online).

Luscuere, L. and Mulhall, D. (2018). Circularity information management for buildings: The example of materials passports. In Designing for the Circular Economy. Routledge.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. www.bitcoin.org.

React (2023). A JavaScript library for building user interfaces. https://reactjs.org/.

Sheffer, D. (2011). Innovation in Modular Industries: Implementing Energy-Efficient Innovations in US Buildings. Stanford University.

Stacks-Network (2023a). Gaia: A decentralized high-performance storage system. https://github.com/stacks-network/gaia.

Stacks-Network (2023b). Stacks - A Bitcoin Layer for Smart Contracts, DeFi, NFTs, and Apps. https://www.stacks.co/.

Swarm (2021). SWARM - Storage and Communication Infrastructure for a Self-Sovereign Digital Society. https://www.ethswarm.org/swarm-whitepaper.pdf.

Swisslife (2017). What is the lifespan of a house? https://www.swisslife.com/en/home/hub/what-is-the-lifespan-of-a-house.html.

Tao, X., Das, M., Liu, Y., and Cheng, J. C. P. (2021). Distributed common data environment using blockchain and Interplanetary File System for secure BIM-based collaborative design. Automation in Construction, 130:103851.

Tezel, A., Febrero, P., Papadonikolaki, E., and Yitmen, I. (2021). Insights into Blockchain Implementation in Construction: Models for Supply Chain Management. Journal of Management in Engineering, 37(4):(ASCE)ME.1943–5479.0000939.

Tian, Y., Lu, Z., Adriaens, P., Minchin, R. E., Caithness, A., and Woo, J. (2020). Finance infrastructure through blockchain-based tokenization. Frontiers of Engineering Management, 7(4):485–499.

Web3-Foundation (2022). Web 3.0 Technology Stack. https://web3.foundation/about/.