User-owned Data in Educational Technology: EduVault White Paper

Learners Can Gain Control Of Their Data And Increase Privacy, Interoperability, And Utility.

Jacob Cohen-Rosenthal

Abstract:

The public discussion about control and ownership of data has centered on social media and search. This article makes the case that the use of data in education warrants special attention. Giving learners control over their data unlocks new possibilities for interoperability, innovation, and the democratization of development. User-owned data increases privacy and protects the education sector from falling into a 'surveillance capitalism' profit model that undermines democracy and may not align with learners' needs. Effective AI-guided education will need to know the learner's entire learning history. Whoever owns that data will gain even deeper insights into the learner's mind than social media and search. Besides being used for advertising, those insights could be used to influence or manipulate the most impressionable members of society - youth. User-owned educational data lays the groundwork for powerful AI-guided education while maintaining transparency and learner autonomy.

User-owned data requires both technological and regulatory solutions. This paper examines the current technical options. This paper proposes a definition and description of user-owned data in the context of education and a practical implementation, 'EduVault,' a user-owned data wallet for educational apps.

This paper shows that the technology to give users private, personal cloud databases already exists but might not be mature enough for large-scale, mission-critical operations. The technologies that give users the most control are still developing but can already be used for smaller, less critical use cases.

1. Introduction

Increasing interoperability among Educational Technology (EdTech) Applications (apps) can allow deeper analytical insights from learners' data. These advancements may have the potential to increase the efficiency and efficacy of education [1]. The current convention in apps seeking to provide cross-device synching is to use a custodial approach, where user data is 'siloed' in a cloud database controlled by the app. The data in the cloud is the 'source of truth,' and data stored on any one device might only be a part of the whole. Therefore, each time the

user opens the app, they must request permission from the app provider to access the data. The app provider has complete control over the data. They could deny access or use the data in any way they choose (in accordance with applicable legal constraints) and without the user's awareness. 'User-owned data' turns this model on its head. Instead, apps request access to data from the user's device or a personal private cloud. Data can sync across apps, even from different app providers, which increases interoperability and user control.

EdTech data can include the content data (course content, quiz content, flashcards, notes, discussions), progress and performance data (test results, grades, course history), and metadata (interaction behavior, completion times of various tasks). Combining these can create a complete picture of a learner's progress and learning style.

In this article, I survey the literature and currently existing technologies around how to incorporate User-owned data into the EdTech sector. Finally, after analyzing and comparing the available technologies, I use those insights to suggest a potential structure for a user-owned data wallet for educational apps.

1.1 Defining 'User-owned Data'

Users of internet-connected applications are demanding more control and ownership of their data, and this is a topic that is gaining attention from the public sector and the media. From the recent U.S. senate hearings on big tech monopoly to the passing of data protection bills like the EU's GDPR or California's CCPA, the problems have been identified by many. Solutions or improvements to the current situation, both technical and regulatory, have been proposed by various academics, companies, organizations. Some of these parties call for 'digital agency,' 'data democracy'[13], 'data fiduciaries'[21], 'data stewardship,' 'open data'[15], 'local-first data'[18], and 'data unions' [20]. Many of these groups/people also use the term 'user-owned data,' but with different contexts and definitions. This paper will propose a definition useful for discussing user-owned data in education, presenting two main criteria; control and hosting.

User-Owned Data: No third party can access, read, write, sell, analyze, or perform any actions on a user's data without express permission. This includes the right to data deletion, and the "right to be forgotten" (as exists in the EU [33])

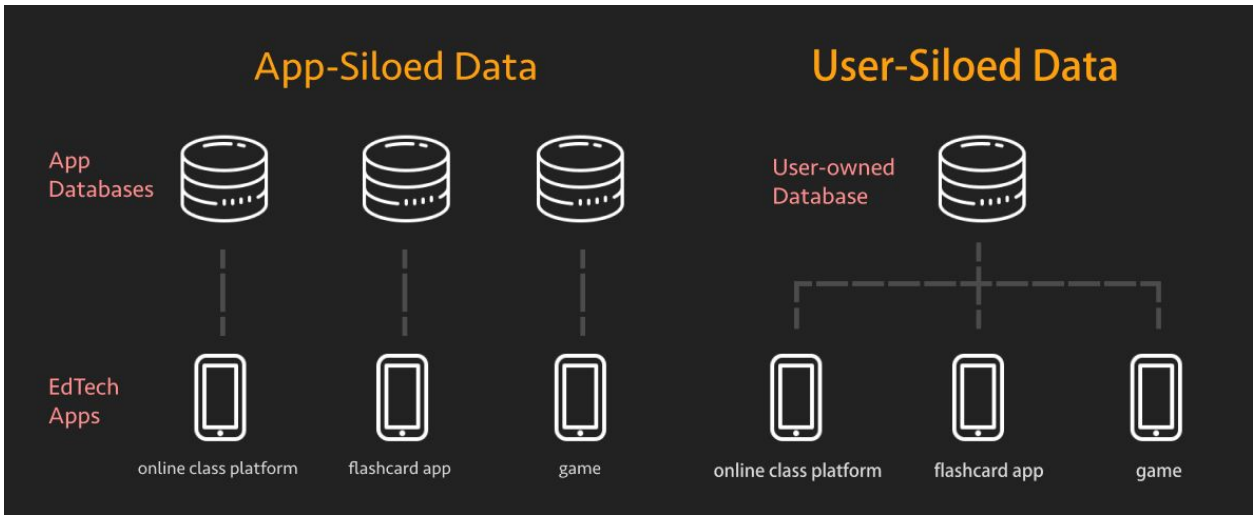
1.1.1 Defining User Data (Access) Control

There are some subtextual definitions that can expand on our understanding of how user-owned data exists in practice. The first is the concept of *Data Control. Data Control* includes the right to proceeds from the use of personal data.

Once an internet-connected app can read a piece of user data, it can copy it, upload it to a cloud server, and sell, analyze or edit it at will. Control of user-data, once it is out of the user's hands, is not possible, so 'control', or 'access control', here mostly relates to who can access

the user-siloed database. Therefore, it roughly corresponds to the idea of ‘access control’ in the study of databases. Only regulatory solutions like the GDPR can enforce the rights of users once the data leaves their device. New technology discussed further on in this paper offers more control in this area to users.

Questions remain about whether or not it would be beneficial to users or the economy at large to implement some kind of data marketplace. Further research is needed to identify a solution for who controls user data and who profits. One solution is ‘Data dividends’ like those proposed by tech leader and presidential candidate Andrew Yang [22]. These dividends would give users a portion of the profit generated from the analysis or sale of their data.



Note: Although this paper will continue to use the term ‘user-owned data,’ the technical solutions explored below could more accurately be described as ‘user-siloed’ data, where each user has their own personal private cloud database. ‘User-siloed data’ as opposed to ‘app-siloed data’ is already a significant step towards full ownership.

1.1.2 Defining Data Hosting Control

Data hosting here refers to how the data is hosted or stored. To be truly user-owned, users must have control over the storage of their data. An encrypted database, where only the user has read/write access, could fulfill the first criteria of ‘data access control’. However, if that database is on a server/datacenter where a third-party could restrict access or intercept messages, the data is not truly user-owned. Platforms like Digi.me and the Hub of All Things (HAT) [3, 4] give users private personal cloud databases and have made advances in the ‘control’ criteria, but fail on the data hosting criteria because they can still cut off users’ access at any time. ‘Self-hosted cloud’ services like nextcloud.com or owncloud.com are a slight improvement over HAT of digi.mi because they cut out the middlemen who can restrict service. Most self-hosting services still rely on centralized cloud hosting services like Amazon AWS, but some can use multiple cloud providers to gain redundancy and lower the risk of service cuts.

Unfortunately, self-hosting is too technically challenging for most casual internet users and does not fully solve the user-owned hosting criteria.

The following section will describe several new technologies that leverage blockchain or related distributed networks that host data without over-reliance on third parties, giving users much greater assurance of continued availability.

1.2. EdTech Data Sharing Challenges and Opportunities

Social media, entertainment, office work, and many other parts of online life might also benefit from user-owned data. The technology that gives users control of their data can be used in any domain, but EdTech warrants special attention and a specific approach. It is a promising industry to bootstrap adoption of user-owned data because, unlike social media, EdTech users are looking for increased productivity and performance, which can be improved through interoperability. EdTech apps often lack interoperability when compared to office apps, like cloud text editors, note-taking apps, project planning apps, and calendars.

Platforms like Digi.me and the Hub of All Things (HAT) Community Foundation [3, 4] have attempted to create an all-encompassing user-owned data model for life across the web, focusing mainly on creating options for user-owned data for use on social media networks. Digi.me and HAT's primary mission is to improve privacy, but their app store reviews show that some users are skeptical that their privacy is really being protected. Some users report feeling nervous that a new platform can now read all of their data. Beyond that, users report not finding any functional benefit to using the service [5].

For user-owned data to be adopted, it will likely need to attract users by offering greater utility, not by using privacy as the main selling point. Unlike social networks, which have coalesced around several large players, EdTech is more atomized. It doesn't currently have the same network effects keeping users tied to a particular platform. User-owned data can also improve EdTech apps because students and independent developers can more easily access the dataand create new features and uses of that data.

EdTech is in a unique position to overcome the difficulty of bootstrapping the adoption of user-owned data. The new features and increased interoperability of connecting even just two EdTech apps already provide value to learners, whereas social networks need to reach a critical mass of adoption before they can attract adoption.

Several technologies are emerging from the field of distributed computing that can give users more control and ownership over their data [Textile, etc.]. The risk-averse public education sector might want to give these new technologies more time to mature and prove their reliability, performance, and security. Nevertheless, some of them are reasonably ready for adoption by

app creators today. Therefore, it is important that more researchers and app developers are aware and informed of these technologies.

1.2 Attempts at EdTech Interoperability

From 2011 to 2014, the EdTech initiative inBloom worked to achieve interoperability across EdTech apps by being a one-stop-shop for EdTech apps in public schools. But the company rapidly dissolved in 2014 due to a data security scandal, amid concerns from parents fearing misuse of student data in inBloom. The company's demise coincided with Edward Snowden's revelations of widespread domestic surveillance. But, despite this bad timing for the EdTech innovators, the public implosion of inBloom revealed growing concerns in the U.S. over student data privacy. New laws, like California's SOPIPA, were enacted to protect student data [6] with advocacy and support from non-profit organizations like Common Sense, an organization focused entirely on student privacy in EdTech [34].

For a time, inBloom's failure dashed the dream of achieving total interoperability or a unified single sign-on across EdTech apps. Since then, Clever [2] has revitalized that dream and has made improvements over inBloom. Clever has improved security practices over inBloom, and perhaps more importantly, improved PR; it has put privacy and security as a prominent feature. However, Clever is primarily aimed at U.S. public schools and institutions. It is only used as a login for school district student records and for EdTech apps used by the school districts. It is not a broad enough universal solution for EdTech apps in general. Furthermore, there are several fundamental flaws to their custodial approach to user data, including being a 'honeypot,' which is a slang term for a data-rich host that is a prime target for hacking. Also, there are concerns associated with any one company becoming gatekeepers for applications that wish to interface with user data. Although this gatekeeping can help keep users safe, there are other methods, some examined in this paper, that can democratize development and protect users in a more empowering and less patronizing way.

1.3 Benefits of User-owned Data in EdTech

New features and capabilities are enabled when users can take their data with them between apps. For example, an online course's subject material might overlap with content reviewable in a game. Learners who play the game could skip redundant parts of the course. Concepts in the course that have been shown to be more difficult could be slowed down or repeated more often in the game. Crucially, the course and game could be created by different companies that each specialize in one area. This promotes the kind of healthy competition that does not exist in 'walled gardens' or big platforms that lock-in users and try to meet their every need. As Carson Farmer from Textile [11] describes, with user-owned data, apps become 'wrappers for data' instead of warehouses [12]. Users can switch effortlessly between apps and be able to enjoy

the next app's full suite of features with little to no onboarding hassle. Instead of relying on user lock-in to retain users, apps must provide the best experience possible to remain competitive.

User-owned data can democratize development. Users can try out new apps more easily, which benefits smaller firms and independent developers. Small 3rd party apps or even students themselves could make apps that interact with learner's existing data in new ways. EdTech that is created by bureaucratic organizations can be distant from the daily experience and needs of teachers and students. Often the best improvements come from the grassroots. For example, quizlet.com and gimkit.com were both created by highschool students who felt that the EdTech apps they were using lacked certain features.

Compliance with new data laws could potentially become easier with user-owned data. If app creators do not own any of the user's data, they might not have to worry about the complexities of the EU's GDPR or California's CCPA.

User-owned data, if widely implemented, could help to prove skills or completion of studies and to grant credentials and certificates. Learners could decide which parts of their educational history to share with schools and employers. Decentralized Identities (DIDs) [38] enable this kind of granular permissions to personal information. The difficulty in requesting and sharing college transcripts and other official documents is often brought up as an example of the potential benefits of DIDs. DIDs, or other blockchain technologies, which can provide immutable proof of identity of the learner and authenticity of their study record [19].

Finally, User-owned data could be used to build powerful AI-guided education. Personalized instruction is an officially recognized potential use of AI in education [6], and further exalted in media as a holy grail of education [7]. Stories like Neal Stephenson's *The Diamond Age* [8] portray a future where AI-guided education can teach anyone anything in the fastest way possible and in the best and most engaging format tailored perfectly to that individual. It is yet to be determined whether or not such capabilities are within reach of modern technology. Regardless, the training of any AI requires a large amount of data to be effective. Whether for more narrow, already available uses of AI in education or more ambitious future possibilities, the AI systems will require a history of the learner's performance, behavior, and learning progress. Currently, this data is spread across many app providers and siloed in separate databases. User-owned data could bring all of that data into one location, providing a more complete picture of the learner. Learners could choose to send their data to AI programs for various personalization benefits. Whether to share or not would be entirely under the learner's control, which could avoid some of the risks outlined below.

1.4 Privacy and Autonomy

Some applications, like Netflix and Spotify, rely on a subscription profit model. However, the most successful (largest) technology companies today, like Google and Facebook, rely on an advertising-driven profit model that Shoshana Zuboff has famously coined ‘surveillance capitalism’ [9]. Edtech apps that are not created by these companies generally rely on subscription, pay upfront, business or institutional corporations, or a more simple form of advertising as revenue models. Google does not have to rely on these more traditional revenue models and can afford to take little or no direct immediate profit from its EdTech ventures. Its ‘free’ services like Google Classroom and G Suite for Education or its Chrome Book laptops for public schools are examples of Google’s encroachments into the classroom.

The surveillance capital profit model is not directly incentivized to improve learning outcomes or the mental well being of the user. It is only incentivized to encourage the user to produce more and more behavioral data useful for advertising. User-owned data could help stave off this profit model that is potentially misaligned with learners’ needs. Even if surveillance capitalist EdTech apps do provide quality educational experiences, Zuboff warns that putting more data in their hands could have disastrous results for the future of democracy and even human autonomy. User behavioral data in education should be used strictly to improve outcomes or experience in the vein of Netflix or Spotify, not Google or Facebook

Education is the next huge market for tech companies. They’ve already set their sights on it and are closing in [10]. It’s a large part of many people’s day that is not currently surveilled, so it is one of the last untapped reservoirs of human behavior to harvest. It’s where the most impressionable members of society spend most of their day. It offers powerful insights into people’s minds. Besides used for advertising, those insights could be used to control or manipulate learners.

AI-guided education has huge potential for improving performance. Training the AI will require massive amounts of data. Either one company creates a monopoly, or that data is collected and traded in opaque processes. User owned data could let learners opt-in to AI training programs and sell or anonymize their data.

Educational performance could be used to prove competency and let those with less impressive formal credentials get ahead on merit. It might also be used punitively. Whether to share this information should be in the hands of the individual.

2. Considerations for the EduVault Prototype: Identifying Best Practices

As mentioned above, there is currently no perfect technical solution to user-owned data available because once data is out of users’ hands, it is out of their control. The (introductory) prototype for user-owned data explored below aims to create a new data storage architecture that changes the initial power dynamics between apps and users.

Instead of having a separate silo of data for each app, each user will have their own personal private cloud database. Instead of the user logging in to the app's system and requesting access to (what is ostensibly their own) data, the app asks for access to a certain section of the user's database. The user can take their data with them between apps and set granular read/write permissions to different parts of their database for different apps.

2.1 System features

Such a system requires several components; the database(DB) itself, an authentication system, a user interface for data management, and the ability to interface/integrate into apps. It would also need to conform to the two user-ownership criteria mentioned above; control and hosting.

Some bonus features would include using 'local-first'[18] design principles and the ability to be used on local P2P connections without connection to the wider internet. That would help in places with poor connections and bandwidth restrictions, which is common in classroom settings, especially in developing countries.

Each core feature could warrant a long discussion. This paper will just give a brief summary of each feature's importance in user-owned data systems, introduce some of the current technologies or strategies available, and list some considerations that are useful when picking one.

2.1.1 The Database

Traditional databases(DBs) have some kind of access control mechanism, whereby only users with certain privileges can see certain data. Usually, users can't access other users' data but the administrator has access to all users' data. Theoretically, traditional databases could achieve the 'control' criteria for user-owned data by simply not granting administrators that overarching access. Most apps, however, do not directly query a database, they go through a server first. The app/server administrator would still be able to intercept the user credentials.

Blockchain and related technologies enable data storage that reduces reliance on third parties, which can help meet the 'hosting' criteria of user-owned data. The simplest method is to append data to transactions for example using Bitcoin's 'OP_RETURN' feature, although Bitcoin Core does not recommend doing this, because it's far too slow and expensive [24]. This data will be stored indefinitely as part of the blockchain. The data would be completely public but could be encrypted. This seemingly achieves both the 'control' and 'hosting' criteria. Ethereum has block(transaction) times of 15 seconds, and fees ranging from several cents to several dollars, which is still not performant enough for most apps' needs. The next few major updates to Ethereum will improve performance and cost, but the Ethereum Foundation still suggests not to use the Ethereum blockchain directly for data storage [25].

A more viable alternative is Bitcoin SV, which is much more amenable to storing data on the blockchain. Transaction speed(0-conf is near-instant) is fast enough, and cost (fractions of a cent) is low enough to be feasible for apps without a large volume of data writes [26]. However, there are limitations to Bitcoin SV as well. Using Bitcoin SV requires either users or app providers to pay these costs even from the initial onboarding. The cost of blockchain storage is 'front-heavy,' meaning the entire expected cost to maintain the data indefinitely is paid upfront. Data stored on the blockchain cannot be edited or deleted. Every change of the data must be saved again in a new transaction. Despite these difficulties, apps like Twetch [27] and others [28] have used Bitcoin SV to store data.

Another alternative to traditional 'app-siloed' databases is Blockstack's 'Gaia' storage system that purportedly "enables user-controlled private data lockers" [29]. Gaia is agnostic about where the data is stored. Data can be stored encrypted on any cloud storage provider. Multiple cloud service providers could be used for redundancy and to avoid denial of access. Digi.me and HAT are less flexible and have set providers they use. All three of these platforms have created their own rather closed ecosystem, creating a certain amount of vendor lock-in for apps looking to build on top of them.

There are also database solutions that do not employ blockchain software. The Inter Planetary File System (IPFS) is not a blockchain but uses related concepts [30]. It can be used directly to store data, but similar to the issues with blockchains listed above with blockchains, it will not operate like a traditional database. Data on the IPFS is addressed by (a hash of) the content, such that when the content changes, the address changes. Textile [11] and 3Box [31] build on top of the IPFS and enable developers to use the IPFS much in the same way they would a normal database.

Solid is a promising data privacy solution directed by Sir Tim Berners-Lee, the creator of the World Wide Web. In Solid, each user is provided with a 'pod' which is a private personal data store. Solid easily satisfies the 'control' criteria but as of yet only works with centralized hosting options like Amazon AWS. Solid is unique in that it uses 'linked data'. Linked data is organized in 'triples', instead of the 'key':'value' pairs that most databases use. The drawback of this database is that it combines two unfamiliar concepts, user-owned data and linked data, in one project, creating a steep learning curve for many developers. [32]

Some of the many aspects to consider when choosing a database:

- Access Control. Is that granted by the database operator or is it user-controlled and built into the DB?
- Hosting. Does it rely on a single database/datacenter provider who could restrict access? How easy is it to migrate (how much vendor lock-in is there)?
- Developer experience and ease of implementation. Is it similar to familiar DBs? Does
- Cost. Who pays for it and how and how much?
- Database reliability and performance

.

2.1.2 Authentication and Identity

Identity management is currently a thorny barrier to success for user-owned data apps. There is often a tradeoff between convenient user experience(UX) and trusting a third party. For example, cryptocurrency wallets or similar systems that use public/private key cryptography. They offer the highest amount of user ownership and the least amount of trust and reliance on third-parties, but they have a steep learning curve. They often require the user to store long back-up passwords(seed phrases) and don't offer any way to recover access otherwise [23]. Using third-party providers like Google and Facebook (oAuth2.0) is not an ideal login and authentication method for user-owned data apps. If this method is used to control access to a database, then these third-party custodial services can potentially access the user's data without explicit permission.

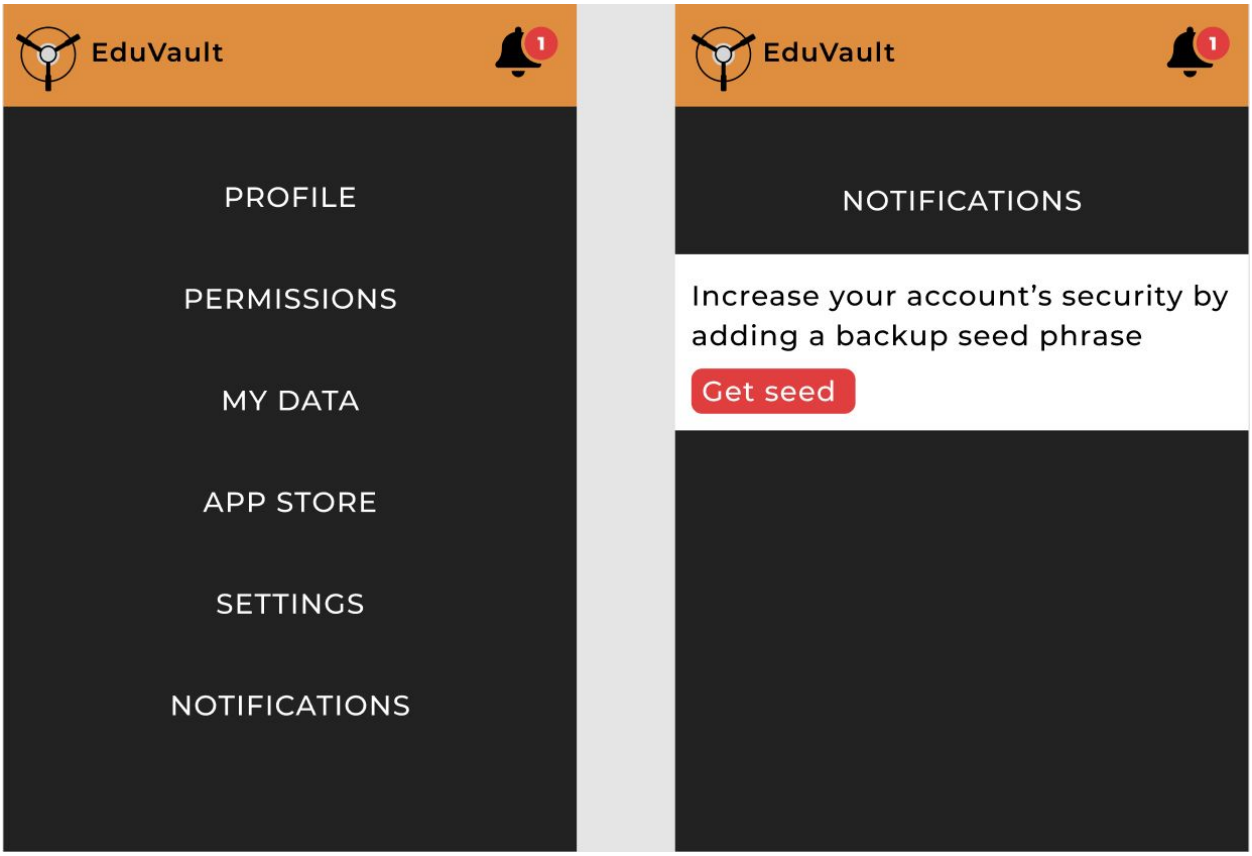
Some user-owned database platforms like digi.me, HAT, and blockstack come with an in-house authentication method. Others like Textile, accept private key challenge authentication, allowing developers to connect cryptocurrency wallets. Some notable projects that provide a good balance between user ownership and UX are Metamask, 3Box. Generally, custodial crypto wallets are easy to use and non-custodial ones are less user friendly.

A successful implementation of user-owned data will need to consider the following about the authentication scheme:

- Is it custodial/non-custodial? Can the provider access or recreate the credentials at any point in the process?
- Does it conform with DB standards? Is it the right format to unlock access a DB?
- User Experience (UX) - How familiar is it? Can users onboard quickly? How is password or account recovery handled? Does the user need to deal with unfamiliar procedures like seed phrases?

2.1.3 Data Explorer

Traditional apps usually only let users explore a subset of data collected by the app. For example, a flashcard app might let the user see their collections of flashcards, but not their performance history metadata. Apps that have integrated the user-owned database might also want to collect certain data but not display it anywhere in the app. The 'data explorer' would be a separate app that lets users explore *all* of the data in their personal database.



The explorer could also act as a learner's EdTech homepage or hub, with an app store that links to all of the compatible interoperable apps in the ecosystem. There does not need to be one 'official' data explorer app. Because apps in a user-owned database are just wrappers for data, various explorers could be built with different graphic interfaces or features. As suggested in the EduVault specifications below, the app explorer could also be used to curate and vet featured apps for learner privacy and safety.

The data explorer should be able to:

- Visualize, organize, edit, share data
- Manage access and permissions to data

And could include:

- An app store to discover apps that have seamlessly integrated the user-owned database.
- Payment integration to manage the buying and selling of EdTech apps, content, and related services.

2.1.4 App integration and language support

For user-owned databases to be adopted widely, they need to be easy to integrate into an app. Because the database will be in a cloud, basic reading and writing to the cloud database will be

done through network protocols like HTTP, which are language agnostic. Therefore, any operational user-owned database should run on any internet-connected device in apps written in any language without much difficulty for the developer. However, to have responsive app performance, and because the network is not always available, apps need to first read and write to local device storage, then sync changes to the cloud when the network is available. If the same data was edited on different devices offline, when they come back online, conflicts could occur, and it is hard to determine which changes to write to the cloud.

'Offline-sync' and 'conflict-resolution' are challenging engineering problems for developers to tackle from scratch and must be rewritten for each front-end language. This problem takes up a large part of the discussion of Martin Kleppmann et al.'s "Local-first software: you own your data, in spite of the cloud"[18]. They compare the existing methods for offline-sync. They give comparably favorable marks to CouchDB for its offline sync abilities but note that it does not offer robust user privacy and user-control. Much of Kleppman's research centers around Conflict-Free Replicated Data Types (CRDTs), a promising technology to give apps better collaborative, offline, and user-owned capabilities. However, he concedes that CRDTs are not yet a complete solution to the offline sync problem as they still have some performance issues.

Most of the databases discussed above offer a standard library or Standard Development Kit (SDK), which handles the syncing between the app on the device and the cloud. Because JavaScript can be used to write desktop, iOS, Android, and web apps, most database projects start by providing JavaScript SDKs. HAT and Textile, for example, only offer SDKs in JavaScript.

To help developers integrate the database into apps, a user-owned data system should:

- Have clear documentation.
- Provide libraries or SDKs for the most common client-side languages, starting with JavaScript.
- Handle syncing issues to provide offline support and conflict resolution.

3. EduVault

EduVault is a proposal, currently in development, to bring user-owned data to EdTech [35].

The new kinds of databases compared in this paper provide solutions to the foundational technology of user-owned data. EduVault expands on those foundations and offers solutions to several of the difficulties in the adoption and implementation of user-owned data systems in practice. EduVault deals with the lack of an ideal authentication option by offering users progressive tiers of authentication. EduVault deals with the problem of users losing control of their data once it leaves an app by auditing approved apps on the EduVault app store. EduVault promotes user adoption by increasing app discoverability. EduVault promotes developer adoption by making it easier to integrate user-owned data into apps. It is an open source project [36].

3.1 Database Choice: Textile

Based on the considerations discussed above, I compared several options for databases and authentication. 3Box and Textile's ThreadDB emerged as the strongest candidates. I chose to work with Textile due to a better developer experience. ThreadDB has comprehensive documentation, and its Application Specific Interface (API) is very similar to MongoDB, a commonly used database.

Comparison of database choices for user-owned data

Red = meets requirements.
Yellow = partially meets requirements.
Green = fully meets requirements

	Access Control	Acces controlled by database operator?	Flexible access levels (read only, etc.)?	Fine grained access control?	Hosting	Centralized database/datacenter?	Vendor lock-in?	Developer Experience	API familiarity	Complexity	Documentation quality	Cost/funding model
Traditional DB		🔴	🟢	🟡		🔴	🔴		🟢	🟢	🟢	🟢	
HAT		🟡	🟢	🔴		🔴	🔴		🟢	🟢	🟢	🟢	
Digi.me		🟡	🟢	🔴		🔴	🔴		🟢	🟡	🟢	🟢	
Blockstack		🟢	🟢	🔴		🟢	🟡		🟡	🔴	🟡	🟢	
Solid		🟢	🟡	🔴		🟢	🟢		🔴	🔴	🔴	🟢	
Bitcoin SV		🟢	🟡	🔴		🟢	🟢		🔴	🔴	🔴	🟡	
3Box/orbitDB (IPFS)		🟢	🟡	🔴		🟢	🟡		🟡	🟡	🟡	🟢	
Textile ThreadDB (IPFS)		🟢	🟢	🟡		🟢	🟢		🟢	🟡	🟢	🟢	

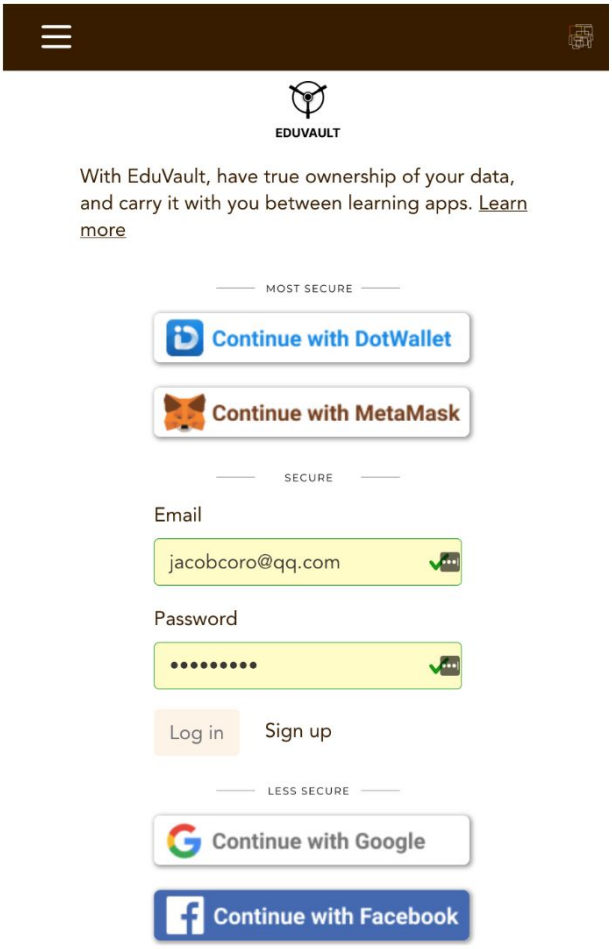
3.2 Authentication Choice: Multitiered

One of the problems listed above about authentication is the trade-off between user control and UX. After comparison, no single authentication method was clearly superior.

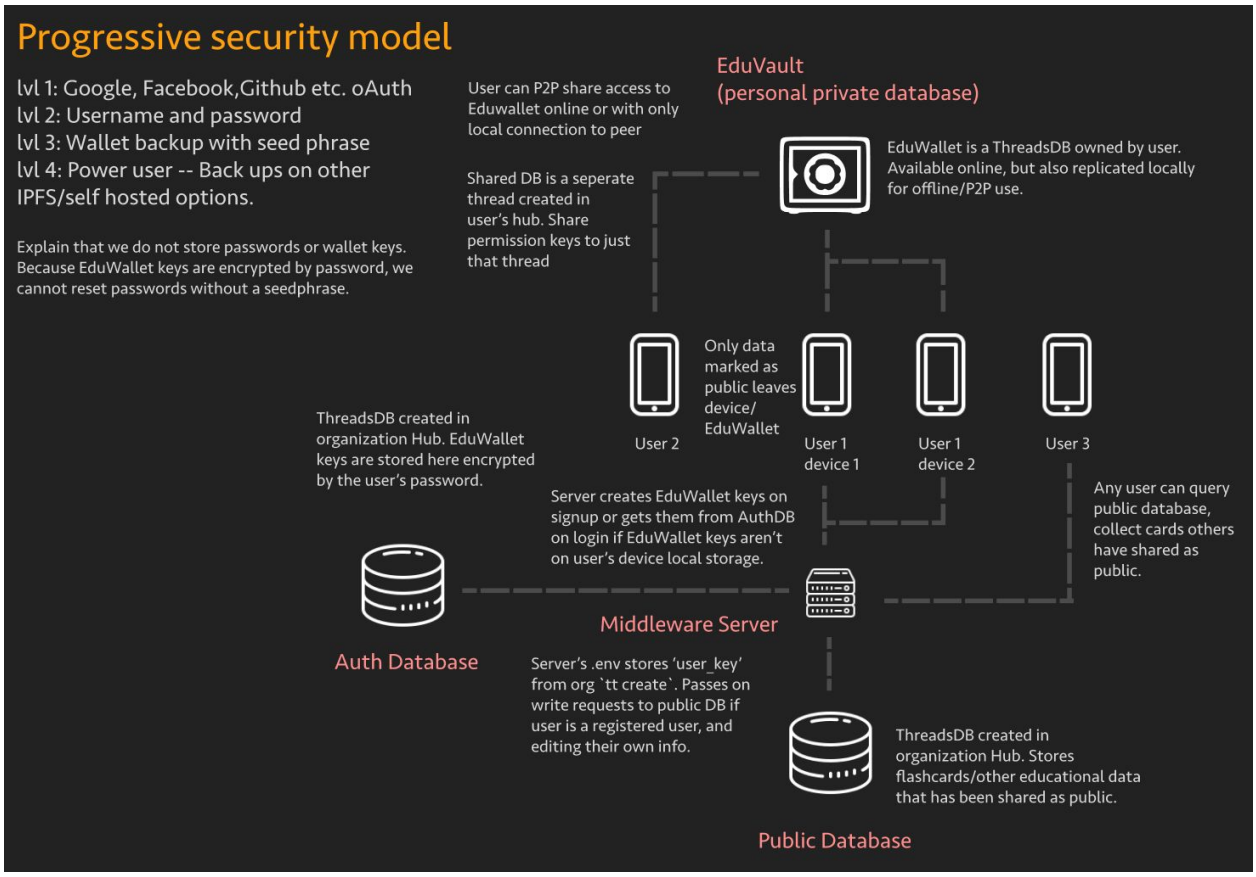
Comparison of authentication services

Red = meets requirements. Yellow = partially meets requirements. Green = fully meets requirements	Custodial/Non-custodial?	Auth method	oAuth	Key signature	User Experience	Familiar? Quick onboarding?	Password recovery?	Confusing 2FA or seed phrases?			
Google, Facebook oAuth	🔴		x			🟢	🟢	🟢			
Clever	🔴		x			🟢	🟢	🟢			
Digi.me, HAT	🔴		x			🟡	🟡	🟡			
Blockstack	🟡			x		🟡	🟡	🟡			
3Box	🟡			x		🟡	🟡	🟡			
Metamask	🟢			x		🟡	🔴	🔴			
Non-custodial crypto wallets	🟢			x		🔴	🔴	🔴			
Custodial crypto wallets	🔴		x	x		🟡	🟡	🟡			

EduVault deals with this problem by offering a progressive, multi-tiered approach to user sign-up and login. Users can choose more custodial login options like Google or Facebook, or more difficult to manage options like cryptocurrency wallets, which give more control. Users can upgrade their level of self-ownership at any time.



If users choose the crypto wallet MetaMask as a login method, their authentication credentials never leave their device. In this situation, there is no need for an EduVault server. The EduVault server saves the credentials for the ‘less-secure’ options. In EduVault’s design, I took care to have the least amount of contact with the user’s credentials. For the password option, the password is hashed on the client-side, then hashed again on the server-side. Only the double hashed password is saved on the server. The EduVault server can never recreate the password.



3.3 Minimization of Required Trust in Third-Parties

The EduVault login page is designed to minimize the amount of trust the user needs to place in EduVault. In the crypto wallet and username/password tiers, the credentials never leave the device. What is unavoidable, however, is that the client-side EduVault login app can see the credentials. EduVault is an open source project and can be audited to show that the EduVault login page code never lets user credentials leave the device. Unfortunately, it is difficult to prove that a site is running the same code from its GitHub repository. Banking sites often encourage the user to check if the URL is correct. This can help ensure that the site is not an imposter site, but it still does not solve the fundamental problem that users need to trust the site to manage their credentials properly. One possible solution would be a browser extension that compares the code sent to the browser with a verified codebase that the page has registered with the browser extension. This is only a partial solution because it would only work for web apps. This is still an open area of research. For now, users still need to place a certain amount of trust in the EduVault login page.

EduVault, or other data explorers for user-owned databases can solve the problem raised in section 1.1.1 whereby users cannot control how their data is used once it leaves their devices. Although not a purely technical solution, EduVault can solve this by thoroughly vetting the apps presented in the app store. EduVault could perform code reviews to verify data isn't leaving

apps and being sent anywhere other than the user-owned database. Apps in the app store could be marked verified or not. Again, this does require that users place trust in EduVault and the EduVault reviewers. However, because users can switch to another data explorer at any time, there could be competition for the most reliable data explorers/app stores.

3.4 Minimizing Vendor Lock-in

The EduVault data explorer will give users the ability to export their login credentials easily. Users can leave any time they want and take their data with them. Using their credentials, they can directly connect to their Textile ThreadDB without going through the EduVault login page, albeit with a diminished and less convenient UX. Textile ThreadDB requires a server to interface with the DB. They provide a service called 'The Hub' that is such a server, run by the Textile company. The code for The Hub is open source, and anyone (who is technically capable enough) could run their own server, diminishing reliance on the Textile company. Textile also plans to support more options for duplicating and backing-up the ThreadDB through third-party 'pinning' services like Pinata [37]. Pinning services ensure that data on the IPFS will be continually available.

3.5 Costs and Incentives

For users to own their own data, they must be responsible for its upkeep. This cost could present a barrier to initial adoption by users. Fortunately, Textile and Pinata offer generous free tiers. Textile has pledged to keep its sever code open source, enabling competitors, and ensuring continued low costs. Pinata currently charges 0.15$/GB per month. If users are not storing media files in their database, the upkeep costs will be minimal. Furthermore, without the need for an app-siloed database, the cost of developing and maintaining an EdTech app will decrease. Apps could pass these savings on to users.

Data explorers, acting as a user's 'home' in the user-owned data ecosystem, are a natural place to handle the upkeep payments. Data explorers should already include a payments system and could handle payments to apps on behalf of the user, much like the Apple Appstore and Google Play store do today. They could then also conveniently handle the data upkeep costs for the user. Users could, of course, pay Textile or Pinata directly, but the data explorer could help users keep track of their storage and how much space different apps are taking up. Data explorers could handle back-up and duplication services for the users, reducing hassle. For these services, data explorers could tack on a small fee on top of the data upkeep costs.

Besides this revenue stream, EduVault and other data explorers could charge apps for the safety auditing mentioned above and for favorable display in the app store. They could also add a mark-up to other payments made with the explorer.

3.6 Challenges and Future Development Goals

Data is not truly user-owned unless no third-parties have any ability to access that data without express permission. There is still a small amount of trust in EduVault required by the user. More work is required to get this down to zero. Although EduVault is a considerable improvement over current systems, it needs to be improved until the front-end side of EduVault never has access to user credentials or can prove that those credentials remain private.

Textile ThreadDB is also in continual development. Fine-grained access controls are coming soon, which would allow the user to permission only parts of the database to different apps. EduVault must continue to adopt to and adapt textiles new features.

At the time of writing this paper, EduVault has a working proof of concept, but more work needs to be done to realize the full proposal. That includes a more complete home page, a data browser, an improved login page, fine-grained access controls, support for more languages, and development of SDKs.

4. Conclusions

User owned-data is especially important and promising in EdTech. It can increase EdTech app interoperability, and help learners build a more complete learning record. It can promote democratization and innovation in app development. It can reduce onboarding friction for trying out new apps and could eventually be used to enable AI-guided education.

It can help protect learner's privacy and autonomy, give learners control over their learning progress and academic records, and help those with less impressive formal credentials prove their competency. EdTech deserrves special attention because it is a promising sector to bootstrap the wider adoption of user-owned data.

This paper has shown that user-owned data, or at least user-siloed data, is feasible with today's technology. Eduvault, or a system like it, is a viable alternative to the app-siloed approach which is hampering interoperability in EdTech. Specifically, Textile ThreadDB and 3Box are good choices for user-owned database. However, they cannot be directly used on their own. Additional features, proposed by EduVault, are required. These include; support for flexible authentication options, a data explorer and user hub, and the manual curation and auditing of apps. Developers interested in user-owned data for any domain should consider a user-owned data system similar to the EduVault proposal.

REFERENCES

[1] Congressional Research Service. 2018. AI and education.
[2] Single Sign-on for Education | Clever. https://clever.com/. Accessed 1 Dec. 2020.
[3] Digi.Me - Your Life, Your Terms. https://digi.me. Accessed 1 Dec. 2020.
[4] “Hub-of-All-Things.” Hub-of-All-Things, https://www.hubofallthings.com. Accessed 1 Dec. 2020.
[5] Digi.Me - Apps on Google Play. https://play.google.com/store/apps/details?id=me.digi.app3&hl=en_US&gl=US. Accessed 1 Dec. 2020.
https://fas.org/sgp/crs/misc/IF10937.pdf
[6] Bulger, M., McCormick, P. and Pitcan, M. 2017. The Legacy of inBloom. Data & Society.
[7] AI and the Holy Grail of Education |. 14 May 2018, https://universitybusiness.com/ai-and-the-holy-grail-of-education/.
[8] Stephenson, Neal. The Diamond Age. Bantam trade pbk. reissue, Bantam Books, 2008.
[9] Zuboff, Shoshana (2019). The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power. New York: PublicAffairs.
[10] Rudgard, Olivia, and James Titcomb. “Big Tech Companies Have Got Universities in Their Sights.” The Telegraph, 5 July 2020. www.telegraph.co.uk, https://www.telegraph.co.uk/technology/2020/07/05/tech-giants-go-big-game-hunting-universities-sights/.
[11] “Textile - A Network of Applications, Connected through Interoperable Data, Where Data Is Owned by the Users.” Textile.Io, http://www.textile.io. Accessed 1 Dec. 2020.
[12] Bring Your Own Data: A New Model for an Interoperable DWeb – Carson Farmer - YouTube. https://www.youtube.com/watch?v=DpRaxjIiYew. Accessed 1 Dec. 2020.
[13] “Why It’s So Hard for Users to Control Their Data.” Harvard Business Review, Jan. 2020. hbr.org, https://hbr.org/2020/01/why-companies-make-it-so-hard-for-users-to-control-their-data.
[14] “MyData Global - an Award-Winning International Nonprofit.” MyData.Org, https://mydata.org/. Accessed 1 Dec. 2020.
[15] The ODI – Open Data Institute. https://theodi.org/. Accessed 1 Dec. 2020.
[16] CitizenMe, https://www.citizenme.com/. Accessed 1 Dec. 2020.
[17] Chakravorti, Bhaskar. “Why It’s So Hard for Users to Control Their Data.” Harvard Business Review, Jan. 2020. hbr.org, https://hbr.org/2020/01/why-companies-make-it-so-hard-for-users-to-control-their-data.
[18] Kleppmann, Martin, et al. “Local-First Software: You Own Your Data, in Spite of the Cloud.” Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software - Onward! 2019, ACM Press, 2019, pp. 154–78. DOI.org (Crossref), doi:10.1145/3359591.3359737.
[19] “Blockchain in Education.” Office of Educational Technology, https://tech.ed.gov/blockchain/. Accessed 1 Dec. 2020.
[20] TheDataUnion. https://thedataunion.eu/. Accessed 1 Dec. 2020.
[21] Heller, Nathan. “We May Own Our Data, but Facebook Has a Duty to Protect It.” The New Yorker,

https://www.newyorker.com/tech/annals-of-technology/we-may-own-our-data-but-facebook-has-a-duty-to-protect-it. Accessed 1 Dec. 2020.
[22] Data Dividend Project. https://datadividendproject.com. Accessed 1 Dec. 2020.
[23] “Ethereum Wallets.” Ethereum.Org, https://ethereum.org. Accessed 1 Dec. 2020.
[24] Bitcoin Core Version 0.9.0 Released.
https://bitcoin.org/en/release/v0.9.0#rebranding-to-bitcoin-core. Accessed 1 Dec. 2020.
[25] “Decentralized Storage.” Ethereum.Org, https://ethereum.org. Accessed 1 Dec. 2020.
[26] Bitcoin Wiki. https://wiki.bitcoinsv.io/index.php/Main_Page#Applications. Accessed 1 Dec. 2020.
[27] Home - Twetch. https://twetch.app. Accessed 1 Dec. 2020.
[28] AGORA - Your Homepage on the Metanet. https://www.agora.icu/. Accessed 1 Dec. 2020.
[29] Blockstack White Paper.
https://uploads-ssl.webflow.com/5e7b1a27d160ce49af1c24e1/5f1596b27c92eb866da76462_whitepaper.pdf. Accessed 1 Dec. 2020 from https://www.blockstack.org/.
[30] Protocol Labs. “IPFS Powers the Distributed Web.” IPFS, https://ipfs.io/. Accessed 1 Dec. 2020.
[31] Shared Backend for Web3 | 3Box. https://3box.io/. Accessed 1 Dec. 2020.
[32] Home · Solid. https://solidproject.org/. Accessed 1 Dec. 2020.
[33] Huergo Lora, Alejandro and Agencia de Protección de Datos (Espanya). La Potestad sancionadora de la Agencia Española de Protección de Datos. Aranzadi : Agencia Española de Protección de Datos, 2008.
[34] How Tech Is Changing Childhood | Common Sense.
https://www.commonsense.org/our-impact/. Accessed 1 Dec. 2020.
[35] Eduvault. https://www.eduvault.org/. Accessed 1 Dec. 2020.
[36]“EduVault.” GitHub, https://github.com/EduVault. Accessed 1 Dec. 2020.
[37] Pinata - Add Files To IPFS Effortlessly. https://pinata.cloud/. Accessed 1 Dec. 2020.
[38] Decentralized Identifiers (DIDs) v1.0. https://www.w3.org/TR/did-core/. Accessed 1 Dec. 2020.