PROJECT INFORMATION

Project Title	**Privacy-preserving Decentralized Identities through Blockchain & OAuth2**		
Technology Area	**Identity Management/Decentralized Networks**		
Project Team	**Name of Team Members**		
	1	**Eric Sedore**	
	2		
	3		
	4		
Keywords (max. 4)	**1. OAuth2**		**2. Identity management systems**
	3. Decentralized ledger technology		**4. Zero-knowledge proofs**

Abstract

Centralized control over private user data is a growing concern for users of the modern internet. This is an issue increasing in scale as users are necessitated to identify themselves on a persistent basis while accessing internet services. The companies that maintain these persistent identification processes have monopolized control on end user data through these services. Nonetheless, identification for secure access of data is a necessity for the online world. This report considers the problems inherent to centralized identity management in favor of a decentralized identity management system. Single points of failure, increasing complexity & friction to the end user, and monopolized control over end user data are growing concerns adjacent to centralized systems. To decouple the identification process from traditional centralized identity management solutions, a privacy-preserving decentralized digital identity solution developed through blockchain technology is proposed. Use of blockchain technologies allows for the decentralized validation of transactions. Zero-knowledge cryptographic proofs are utilized to ensure that private user data is not leaked in transit when authenticating to a resource or identifying a user. To ensure ease of integration and adoption, the solution is designed to be compatible with OAuth2 authentication portals for easier standardization. OAuth2 compliance would also allow for the use of token-based authentication in comparison to password authentication which carries a series of specific attack vectors. The proposed application of the research will develop a web application to initiate and validate authentication requests for end users in a manner compliant to the OAuth2 access token request flow. Authorization records will be stored and validated on a decentralized ledger anonymized through privacy preserving zero-knowledge proofs. This provides a decentralized ledger in place of an OAuth2 authorization server for credential authentication when requesting access to protected resources.

1. INTRODUCTION

Extensive research has been conducted in recent years into using blockchain systems to provide a form of decentralized identity (DID), also sometimes known as self-sovereign identity (SSI). These are decentralized identity management systems which record and store user credentials on a decentralized and publicly or privately accessible ledger. The necessity for these systems as proposed in the literature often comes down to two central factors. Firstly, modern identity management systems are centralized under a 'siloed' approach. This instigates the risk of a single point of failure bringing down a whole network and its identification/authentication process. A centralized system if breached or impacted by a cyber-attack would bring down the dependent systems upon which it relies. A breach of these systems also incurs the loss of all data stored on these central solutions. The secondary concern refers to user control over identifying data. If data used to authenticate a user is stored on a centralized service owned by the authenticating service, the end user loses control over the private identifying information stored there. Using a study performed in January 2021, adults surveyed globally reported that 66% of participants felt that tech companies held too much control over personal data (Johnson, 2021). This lends public support towards the development of a privacy-conscious identity management system. The approach we employ leverages the decentralized nature of blockchain networks to place secure storage and access of user data into the hands of the consumer. Centralized identity management systems are also growing in complexity and end user friction as alternative factors of authentication become necessary to protect against increased skill of attackers (Gisolfi, 2020). This added complexity increases the attack surface of these identity management systems.

The field of digital identities, also known as identity and access management (IAM) dictates the level of access and authorization end users have over protected resources. The main goal of these systems is to deter unauthenticated or otherwise unauthorized access to private content or resources. Examples of modern IAM systems include the commonly utilized Active Directory on Windows environments, or LDAP on Linux environments. These are usually centralized systems which are used to manage all aspects of user and identity account management. Common use cases of IAM technology are in user registration, access control, privileges, and password management functionality. These systems will typically be hosted on premises or on a cloud environment. Regardless, these IAM systems still form a centralized store of user account information. Centralized IAM systems also provide the security features of logging connection attempts

and events. Single sign on (SSO) and federated identity management are two popular forms of deploying IAM processes. These two examples of IAM allow for end users to utilize and access services of partnered services using the same shared credentials across all services. This allows partnered services and systems to verify the credentials supplied to them through a centralized and trusted third party. The annual identity management market revenue is also forecasted to grow from a market value of 23 billion in 2020 to a projected 49.5 billion annually by 2026 (Alexandra Sava, 2021). This growth is largely expected due to increased volume of identity fraud and identity-based attacks as well as governmental regulation put in place to deter these attacks.

Identity based attacks account for much of the initial compromise phase of an attack. Microsoft reports in their 'Digital Defense Report' that 70% of attackers are choosing to use phishing as the means to harvest user credentials (Burt, 2020). Phishing and the more targeted spear-phishing are examples of targeted identity attacks which aim to steal user credentials. Attackers can forge SSO or federated identity management log in pages to appear 'credible'. This is where attackers will forge an Office365 or Amazon log in page to appear legitimate to the end user. By supplying credentials through these forged websites, attackers can gain credentials to a federated or centralized system. These links are often supplied through phishing emails. Attackers use both compromised and breached credentials to access user accounts as well. In password spraying and credential stuffing attacks, an adversary will make use of lists and databases of leaked or breached user credentials. Due to the large number of connected online accounts that an end user will have, password reuse will occur for multiple accounts. In 2020, password management company LastPass produced a report on the "The Psychology of Passwords". This report details how 66% of global respondents to their survey reused password credentials across online services (Fremery, 2021). This increases the attack surface for users who reuse identifying credentials across multiple online accounts. When considering systems which use a federated identity management approach, password reuse provides significant risk.

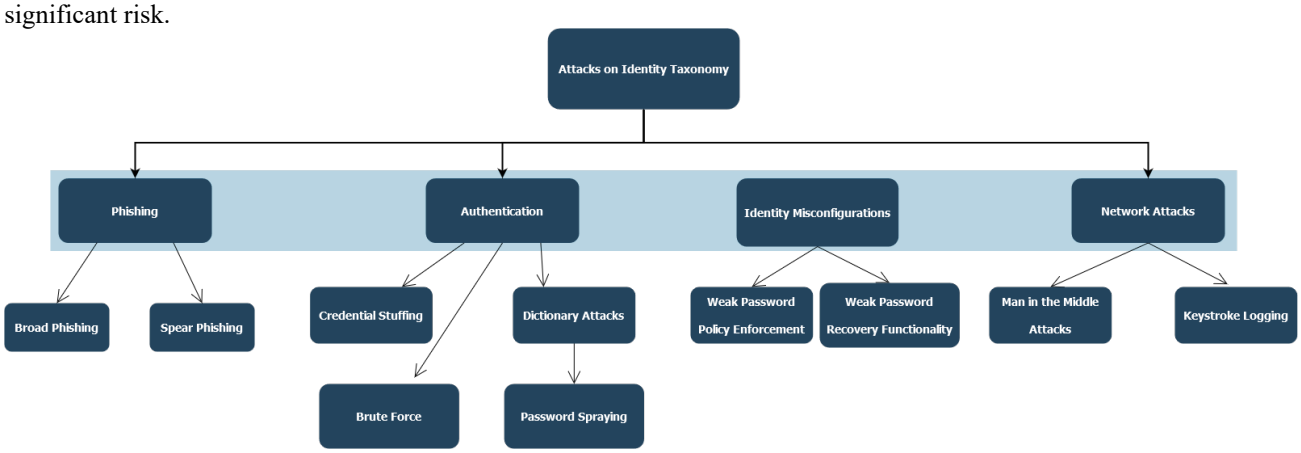


Fig 1. Taxonomy of attacks on identity & authentication

Blockchain development itself carries on the tradition of decentralized computing in a modern application. Blockchain has a long history prior to the release of the Bitcoin white paper in 2008. Dating back to 1982, blockchain-adjacent protocols have been proposed to create cryptographically secure protocols which are tamper proof and decentralized in nature. However, these decentralized applications of an immutable record were not popularly accepted and researched until the Bitcoin white paper was released in 2009 by the anonymous researcher Satoshi. Similar decentralized protocols and networks were proposed in the early to mid-2000s during the heyday of peer-to-peer torrenting and file sharing systems such as napster. Nowadays, spending on blockchain solutions and platforms is expected to increase from 4.5 billion USD in 2020, to 19 billion USD by 2024 (Statista, 2022). This shows that the field of blockchain technology is expecting large growth in the next couple of years. In a 2021 study conducted by Statista which discusses use cases of companies developing blockchain solutions, secure information exchange was the use case of 45% of companies adopting the technology. This was the largest use case listed. However, relevant to this paper, the fourth most adopted use case was digital identification which 40% of respondents listed as a current use case in development (Statista, 2022).

Blockchain technology forms a decentralized and shared ledger of transactions or assets. A distributed ledger is comprised of blocks which are validated in order of occurrence. A block can be any data that is supplied in a transaction. When a new transaction occurs, it's data is added to a block on the blockchain. Each block is linked and connected to the transactions which occur before it. This block is only added to the ledger once it has been validated by other nodes on the network. Validation of individual transactions through external peers and nodes is what makes these systems decentralized. No central authority is present in the validation of a transaction which makes these systems peer-to-peer. This would remove the need for centralized systems to be consistently online to validate transactions or assets. The system used to validate transactions on a decentralized ledger is known as the 'consensus' mechanism. Consensus mechanisms come in many forms and carry different pros and cons for differing use cases. The Bitcoin system uses a 'proof of work' consensus mechanism which uses computation of cryptographic hashes to add new transactions to a block. This adds an energy and processing hardware requirement to systems which utilize proof of work. The environmental considerations of this at scale is an essential consideration. Once a consensus mechanism's requirements are met, then the node can submit transactions to the decentralized ledger for validation as a new block.

It is also worth noting that blockchain solutions come in a variety of network configurations. The first type of blockchain is a public ledger. This is one in which any user can participate, which comes with the tradeoff of public visibility of transactions. This may be desirable still if full transparency of the system is a goal. Private blockchain networks are

decentralized and record transactions in a peer-to-peer manner, however the governance of the network is maintained through a singular, centralized organization. This allows the organization to dictate the consensus mechanism used or allows them to alter other key components of the network. Private networks are scalable and are good applications for internal corporate uses of blockchain technology. Consortium blockchains are the last configuration of note for blockchain networks. Consortium networks are like the federated identity management systems mentioned above. This allows for multiple organizations to confer between each other the manner of which the decentralized ledger is maintained. This forms a hybrid model somewhere between a public and private ledger. Pre-determined authorities in a consortium ledger network are the central figures who determine what nodes can validate or submit transactions on a network. This means that all users of consortium or private blockchain need to be permissioned users who are authenticated for transactions on the ledger. This contrasts with public ledgers in which any user can submit transactions to the ledger. The tradeoff here is in public visibility for more permissioned privacy. Therefore, a consortium ledger approach is likely the most viable candidate for development of a decentralized identity system due to its similarity with well adopted federated identity management systems.

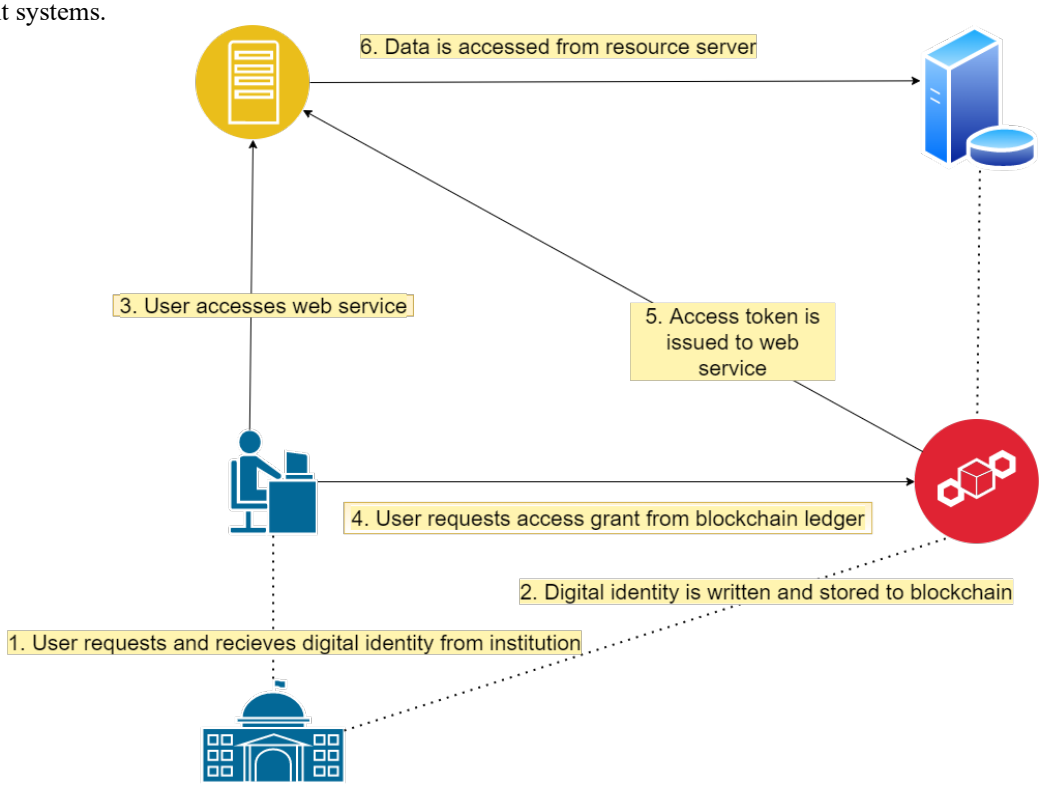


Fig 2. A generalized example of a decentralized identity process utilizing a decentralized ledger to store credentials

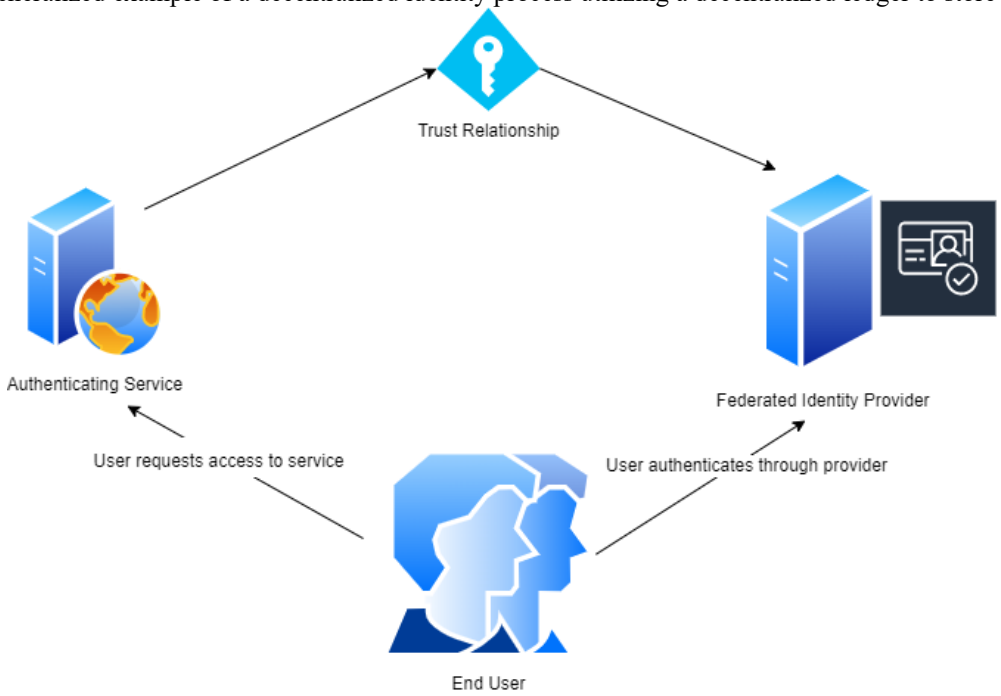


Fig 3. Federated identity management example

Blockchain technologies have been searching for practical applications since their inception. Currency uses, DNS solutions, and file management systems have all been proposed as blockchain applications. However, this proposal uses a decentralized ledger to store private information and user credentials on a decentralized ledger. As blockchain ledgers are publicly trackable and transactions can be correlated to a unique address on the blockchain, privacy controls should be implemented to ensure both confidentiality of data, but also anonymized availability of private data stored on-chain. Multiple approaches in this space implement 'Zero-knowledge proofs'. This is a method of ensuring that an exchange of information can occur without leaking user-private data in the process. Publicly accessible ledgers need a form of anonymization to occur in this process to secure user data stored in this manner. The existence of publicly traceable blockchain addresses which can be correlated back to an end user makes public blockchain addresses 'pseudonymous' rather than truly anonymous. This is where the implementation of zero knowledge proofs anonymizes this access control securely.

Limitations arise with blockchain solutions due to lack of governmental oversight and regulation. Lack of standardized protocols also lends to the decision not to adopt these technologies. Modern corporations are hesitant to adopt decentralized solutions in comparison to the traditional centralized 'silo' approach of data storage. There are numerous factors for these decisions, but mostly these corporations do not wish to lose control over valuable private user data. This factor led to the design decision to make a decentralized authentication system compliant with modernly used and standardized web authentication protocol OAuth2.

In total, the proposed approach would be to create a decentralized authentication mechanism in which the end user does not leak private information that is unnecessary for authentication to a service. This approach utilizes cryptographically verifiable credentials rather than a password-based approach for authentication. This is to mitigate against password-based attacks as have been outlined in Figure 1. Privacy defines the degree of control that end users have over the personal or business information which they share. As centralization and consolidation of tech monopolies occurs, users have increasingly less access over where and why their data is shared with both first - and third-party providers. As data is stored in centralized solutions, user visibility into how their data is used becomes restricted. A novel solution applying the aforementioned technologies allows for users to visualize where their data is being shared via a verifiable ledger. This is a value intrinsic to the public nature of decentralized ledgers. This would be delivered to the end user through a web-based authentication portal visually similar to other OAuth2 authentication portals.

2. PROJECT OBJECTIVES

List of Proposed Objectives

- **Objective 1:** Research various ledger platforms to determine suitable platform to build an authentication process on top of.
- **Objective 2:** Investigate the development of a decentralized ledger to authenticate users through the application of OAuth2 access tokens and credentials.
- **Objective 3:** Ensure that the authentication system is privacy-conscious regarding end user data through development & deployment of zero-knowledge proofs to verify transactions.
- **Objective 4:** Develop smart contracts to handle transactions regarding the following use cases: user registration, authentication, OAuth integration, access token generation.
- **Objective 5:** Develop web authentication portal application to handle requests and authentication flow from the end user to the authenticating service.
- **Objective 6:** Evaluate network metrics and performance indicators to determine scalable indicators.
- **Objective 7:** Perform user use case testing of software to generate end user feedback on ease-of-use, and other user experience factors of platform.

3. LITERATURE REVIEW

Boo, Kim, and Ko provide a lightweight implementation of Zero-knowledge (ZK) proofs in their research (Boo et al. 2020). Boo et al. determine a need for a transaction system which can automatically process multiple payments without the involvement of a central party (Boo et al. 1). A lead concern of their research is to ensure user privacy through encryption and anonymization. However, blockchains are transparent public ledgers and this does not ensure user privacy which is a concern shared by multiple researchers across this space. A single Ethereum (ETH) block can process 476 standard transactions (Boo et al. 2). Computing a ZK proof (ZKP) limits this to six ZKP calculations able to be computed in a single ETH block due to hardware constraints in IoT devices (Boo et al. 2). This dramatically lowers the throughput of the network. The main goal of Boo et al. and their proposed solution is to reduce the burden of ZKP operations on IoT devices and the larger decentralized ledger network as this is a main bottleneck to efficiency and throughput. The implementation chosen by Boo et al. is built on the ZoKrates framework like the approach of other authors in this literature review. LiteZKP was found to reduce latency and energy consumption for a ZK proof operation by 55% on IoT/MEC devices (Boo et al. 2). The verification scheme requires 8% of transaction costs as a fee per fifty anonymized payments, this fee is used to cover gas fees incurred over the Ethereum network (Boo et al. 2). The first challenge listed by Boo et al. is scalability (Boo et al. 4). Nodes do not process transactions in parallel and individual nodes must submit transactions in a block that is written to the chain once consensus is achieved. This means that when the number of nodes increases, the throughput does not as consensus must be received by all nodes. This means that latency may occur while mining happens across all nodes to achieve consensus.

Decentralized identity and authentication systems can also be achieved through a password-based approach as determined by Pawel Szalachowski in their article (Szalachowski. 2021). Single sign-on (SSO) systems ensure that

username/password credentials can be extended to be reused without the creation of a new identity across multiple services (Szalachowski. 1). These systems however place all the data and trust in the hands of the federated systems which centralize and funnel data into their silos. Szalachowski refers to the creation of distributed public key infrastructure (PKI) systems as a peer-to-peer network which distributes trust and lessens the necessity of globally trusted authorities. The proposal of Szalachowski is a password-authenticated decentralized identity (PDID). Szalachowski details the issue of a transparent public ledger regarding user privacy becoming visible. ZK Proofs, cryptographic commitment, and multi-party computation are raised as possible solutions to this public ledger issue (Szalachowski. 2). However, these systems are currently inefficient and introduce throughput bottlenecks due to the resource intensive operations they require. The goals of the PDID system are to provide human readable usernames that are accessible globally and are collision secure and cannot be hijacked. These are all combined to ensure a password based secure authentication system (Szalachowski. 3). The "Global Password Manager" (GPM) is deployed as an entity to standardize user credentials and authentication requests as a confidential smart contract delivered over the blockchain (Szalachowski. 4). This provides the smart contract basis for ensuring users can create self-sovereign identities (SSI) which are associated to a X.509 PKI public key all chained together to be deployed over a decentralized ledger (Szalachowski. 9).

Mulaji and Roodt summarize sixty-nine selected academic sources to develop a media synthesis on modern enterprise applications of blockchain identity management (Mulaji and Roodt. 1). The paper synthesizes modern papers and research to determine the practicality of blockchain as a distributed identity management (IDM) system. The current desire to move away from centralized identity management is in the 'Single Point of Failure' of these systems (Mulaji and Roodt. 5) as similarly determined by other authors in this review. Mulaji and Roodt determine that IoT distribution has made IDM systems more complex due to interconnected devices and accounts (Mulaji and Roodt. 6). Centralized IDM, IDM-as-a-service, Federated-IDM, and User-Centric IDM based approaches are compared against a blockchain IDM solution. In creating a blockchain based solution, Mulaji & Roodt determine three key needs. As stated, they are: What consensus protocol to use, who can join a network, and whether a validator is needed (Mulaji and Roodt 8). Mulaji and Roodt come back around to zero-knowledge cryptographic proofs as a possibility to implement a 'Self-Sovereign Identity" (SSI) over blockchain (Mulaji and Roodt 10). This allows users to create immutable identity records on chain which are unique to the end user. Critics to this approach as outlined by Mulaji & Roodt determine that corporations view the cost of a data breach and related credential loss as preferable to losing control over their users (Mulaji and Roodt. 12). After comparing pros and cons, the authors determine that when considering the benefits of a blockchain based IDM multiple questions must be addressed. As stated by the authors the system must "add value, eliminate a weakness, provide an advantage, or preclude a threat from competitors" (Mulaji and Roodt. 12). For a blockchain based IDM solution to be effective, it is stated that it must address the following: vulnerabilities in authentication methods, vulnerabilities in IDM architecture, the balance between security and privacy, credential reuse and weak credentials, and secure cloud and IoT authentication (Mulaji and Roodt. 15). The paper also addresses that blockchain currently does not prevent credential misuse or authentication failures, it is merely a system to mitigate these risks through distributed cryptography which lacks a single point of failure. Mulaji and Roodt conclude that this technology does not apply at an organizational level just yet but does address IDM challenges around data breaches and cost of IDM stored organizationally. Further research and study is directed towards blockchain standards and regulations. The authors summarize that lack of current guidelines, regulation, and policy is leading to a gap in organizational adoption of this technology (Mulaji and Roodt. 16).

Self-sovereign identity (SSI) solutions are compared in an industrial IoT (IIoT) environment by Figueroa-Lorenzo, Benito, and Arrizabalaga in their paper (Figueroa-Lorenzo, 2021). Their stated approach is to enhance the security of the Modbus protocol which is relevant to data sharing between IIoT devices. The solution provides an on-chain authentication/authorization process based on SSIs delivered over the Hyperledger Fabric Blockchain; this proposed protocol is named "mbapSSI" (Figueroa-Lorenzo et al. 2). Modbus is used to transmit data between programmable logic controllers (PLCs), sensors, and actuators. Figueroa-Lorenzo et al. reference extensive literature that the modbus protocol by itself is insecure which provides the need for a secure authentication system to access the underlying hardware (Figueroa-Lorenzo. 3). Hyperledger Fabric was chosen as the blockchain framework to implement mbapSSI over. This is an open-source decentralized ledger solution maintained by the Linux Foundation. X509 certificates are chosen to implement the decentralized identity for end users and IIoT devices, this is done to replace the need for a Certification Authority associated with common PKI infrastructure (Figueroa-Lorenzo. 5). Encrypted JSON Web Tokens (JWTs) are used to verify credentials sent across this network from the end user to the authentication system. The model proposed by Figueroa-Lorenzo et al. is evaluated on three metrics: latency, processing time, and throughput (Figueroa-Lorenzo et al. 11). Latency was referred to as a critical metric for success in IIoT networks. This is due to the frequent volume of data exchanged over sensor networks and ensures stable performance over the modbus protocol. Network throughput declines over time with increasing number of clients using the modbus protocol over this SSI system. This also goes in hand as latency increases with more clients using the mbapSSI protocol concurrently (Figueroa-Lorenzo. 15). An average latency of 100ms was determined for normal modbus transactions. Through the evaluation process an average latency of 63ms was determined for thirty-two concurrent mbapSSI clients (Figueroa-Lornezo. 15). Figueroa-Lorenzo et al. provide their proof of concept for a fully automated machine-to-machine network which verifies transactions through an OAuth scheme using decentralized identities (Figueroa Lorenzo. 16).

Mir, Roland, and Mayrhofer author an article which proposes a new novel authentication scheme named Decentralized Anonymous Multi-Factor Authentication (DAMFA) in their paper (Mir et al. 2022). A distributed ledger technology

(DLT) is used to employ this scheme and the TOPRF protocol is used to ensure cryptographic randomness. Mir et al. propose an anonymous MFA scheme which focuses on preserving user data privacy. Single point of failure is again addressed by Mir et al. as a problem of centralized SSO solutions. Secondary concerns arise from tracking of user data from the centralized identity provider (Mir et al. 2). Passive verification of users is achieved through SSO data stored on the publicly accessible 'append-only' DLT. This removes the cost of an always-online SSO provider (Mir et al. 2). Non-interactive zero-knowledge proofs are deployed again to prove authentication to a randomized 'pseudonym' stored on-chain which is linked to the SSO data (Mir et al. 7). Mir et al. implement their proposed solution over the cryptocurrency 'Namecoin' which was the first hard fork from the Bitcoin network. Namecoin is used to store, query, and share 'names and related values' on a public ledger. Registration using this technology takes on average 2 hours for the first registration blocks to be confirmed on ledger. Authentication uses a different process which is much faster (Mir et al. 13). This network is compared to a deployment on the Ethereum network as well. Confirmation times for authentication takes anywhere from 10 mins to 2 hours on Namecoin. Authentication confirmation on Ethereum takes a few seconds to 3 minutes making Ethereum the faster system to deploy on (Mir et al. 13). Transaction cost in "Gas fees" is also analysed where Ethereum was found to cost 0.0465 USD less per transaction than Namecoin. However, deployment over Ethereum leads to a larger initial ledger size (5.08GB compared to 5.3GB) (Mir et al. 13). Smart contracts for this Ethereum network were coded using the 'Solidity' programming language for the Ethereum Virtual Machine. Following evaluation by Mir et al. DAMFA was found to fulfil several properties (Mir et al. 15). Firstly, the system ensures decentralization through a DLT which reduces vulnerabilities associated with a trusted central party. Secondly the system provides passive verification which enables verification without interaction with a central identity provider. SSO is built-in to this system and ensures that parties who trust DAMFA would allow for verification across multiple parties through this shared SSO system. Finally, as all values stored on the DLT are anonymized and ZK proofs are enabled for transactions, user anonymity can be confirmed. Mir et al. surmise that due to these properties their system is efficient and practical for authentication (Mir et al. 15).

Argento, Buccafurri, Furfano, Graziano, Guzzo, Lax, Pasqua, and Sacca author their paper (Argento et al. 2020) which is the result of a three-year research project which aims to implement a Blockchain-based solution for an eIDAS-compliant public digital identity system. Non-repudiation and accountability are central themes to the goals of this research project. This infers the need for a secure and reliable digital identity scheme that can be used across multiple organizational workflows (Argento et al. 1). Accountability is defined by Argento et al. as the concern of certifying and verifying a digital identity in the use of a service. The authors determine accountability to be important under the regulatory policies of the EU's eIDAS and GDPR. The proposed solution is named 'ID-Service.' Argento et al. implement their solution on an Ethereum blockchain using the solidity programming language like (Mir et al. 2022). The authors Argento et al. use smart contracts to implement 'Accountability Nodes' (AN) (Argento et al. 4). ANs are trustworthy 'root' nodes which provide services to users on the 'ID-Service' network. These nodes delegate for the end user and track and certify interactions to its services. Users can register themselves to the ID-Service network through public/private key pairs which are written to the blockchain through each AN the user registers on. Argento et al. deployed their solution through a test case with a Biomass production chain. At least five ANs are used to provide services to the organizations along this supply chain at each step of the transaction process (Argento et al. 8). As seen in other studies, once the number of actors/users on the ID-Service system increases, the number of blockchain participants required to compute the transaction on-time rises as well or else performance degrades (Argento et al. 11). Average response time of individual ANs as well as network throughput are selected as variables to evaluate this solution (Argento et al. 11). Analysis of the data after a year of trials led Argento et al. to choose a ledger configuration with three AN nodes as it managed the throughput better than 1 or 2 AN nodes (Argento et al. 13). This motivation is furthered by the fact that less AN nodes on the network makes the system more susceptible to a single point of failure (Argento et al. 13). The conclusions of Argento et al. determine that the ID-Service network provide the principle of security by design, as well as non-repudiation of transactions (Argento et al. 16). Limitations addressed by the authors determine that the platform does not implement a self-sovereign identity model which could lessens user control over personal data (Argento et al. 18).

The researchers Hong and Kim implement a self-sovereign identity (SSI) framework over blockchain which is compatible with the commonly used SSO protocol OAuth 2.0 in their paper (Hong and Kim. 2020). Hong and Kim first discuss the 'Silo' model of authentication as a model which leads to user password fatigue. Password fatigue and reuse was a key concern in the development of the OAuth protocol which allows third party services to delegate authentication through a well-recognized service such as Google or Facebook (Hong & Kim. 1). The authors determine that a lack of standardization for SSI identity frameworks is a fundamental issue barring adoption. This is due to SSI solutions having differing authentication models and processes. This is where Hong and Kim propose a blockchain-based SSI model to comply with the widely adopted OAuth2 protocol (Hong & Kim. 2). The novelty of their approach is in the user-centric authentication process using credentials stored on ledger rather than a siloed or centralized provider. SSI models ensure individual user ownership of their digital identity. Combining this with a decentralized ledger solution ensures integrity and non-repudiation of data stored on ledger (Hong & Kim. 5). Hong and Kim propose their solution named VaultPoint as a proposed OAuth compliant solution. VaultPoint utilizes three smart contracts written in Solidity and deployed on the Ethereum network like Mir et al. and Argento et al. The 'notification' contract and 'client management' contracts are used to register a client to authenticated services on chain. The 'identification' smart contract is the largest and stores the personal information of the registered client, as well as any claims to services users can authenticate to (Hong & Kim. 7). The security of the proposed VaultPoint solution is based in the assumption that OAuth2 is a standardized and trusted

protocol (Hong & Kim. 17). Security analysis of the VaultPoint solution focuses on two major issues: identity theft, and leakage of personal information (Hong & Kim. 17). Push notification tokens are used to supply access tokens to the end users. These access tokens were unable to be forged due to the integrity provided by a ledger which stores these access tokens. VaultPoint also generates a random secret code for each authentication request that users can verify through their browser to ensure authentication requests are legitimate (Hong & Kim. 17). To prevent data leakage, user blockchain addresses are stored on a private ledger. Further, all access requests made are encrypted before being verified on the public chain (Hong & Kim. 18). Conclusions derived by Hong & Kim determine that they were able to implement a blockchain-based SSI framework which is compliant to OAuth 2.0. Hong & Kim determine that this solution is best provided to the issue of user information stored monopolistically and hoarded by a centralizing group of major IT companies (Hong & Kim 19). This ensures the individual right to control their own digital identity through a decentralized identity process.

Cha, Chang, Xiang, Huang, and Yeh further explore OAuth implementations with blockchain technologies in their paper (Cha et al. 2021). Cha et al. propose Non-Interactive Zero-Knowledge proofs to ensure anonymity in their solution. This is an approach shared across multiple authors in this review. Their proposed solution is called "MyDataChain" and is a portable authentication scheme to support authentication requests, grants, and revocations. This solution provides a user-centric, privacy-preserving approach to authentication over a system with portable OAuth credentials. As addressed by Cha et al. the EU's GDPR requires identity providers to let the end users reuse credentials and collected personal data across services. This is where the portability of user data overlaps with OAuth technology and the concern of preserving private user data. As standardization has been raised as an issue by other authors in this review, a solution compliant to both OAuth2 and the GDPR would provide a reliable mode of authentication which may be more ready for adoption. Further GDPR considerations include the "right to be forgotten." As OAuth does not currently include revocation processes, this is an area of concern for the research. This is where Cha et al. propose MyDataChain to store authentication/authorization grants locally for each user where they can be deleted and revoked upon user preference (Cha et al. 2). The proposed solution created by Cha et al. ensures the following functions for the end user (Cha et al. 7). Firstly, the resource requesting party can obtain the users consent to access the end user data stored on a resource server through OAuth. Secondly, the framework provides non-repudiation of transactions to leave evidence of authentications for the end user. Thirdly, the approach allows the end users to know what resource providers have access over their data, revocation of data access rights can also be achieved through this process. Finally, authorization servers have the power to withdraw malicious requests supplied to the authentication system.

Andre Boysen provides a relevant local analysis of digital identity systems as they pertain to Canada in his article (Boysen. 2021). It analyses a local case study of a product named 'Verified.me' which demonstrates how industry can collaborate across a variety of use cases to ensure a digital identity network secured through SSI principles. Verified.me launched in 2019 and was able to secure partners across seven of Canada's banks (Boysen. 3). Boysen states that security and useability are two key principles to bolster trust and increase adoption of these technologies (Boysen. 2). To confirm security, Boysen overviews the principles established by the Canadian Commission on Enhancing National Cybersecurity from 2016. These principles state that consumers must thrive in a digital environment, and companies need to be fair and competitive to secure a new digital economy. A digital economy needs to be based on a secure underpinned digital identity system or else the end user does not have fair and transparent participation in this economy. Boysen believes that communication must occur between private, public, and blockchain based organizations together to ensure a secure digital identity (Boysen. 2). Fragmented, separate online accounts have led to credential overuse, misuse, and overall complexity for the end user. An impact of this Boysen concludes is that centralized identity forces consumers to participate in systems that fragment their data across multiple centralized sources. This leads to users having to supply more data than necessary (Boysen. 3). With the Verified.me project, partner organizations were able to transition some identity services away from the siloed approach to the SSI framework Verified.me provides. The solution was registered under the 'Decentralized Identifiers specification' which is a policy framework set up by the 'Decentralized Identity Foundation' (Boysen. 6). Interoperability between this specification as well as the EU's eIDAS was deemed a necessity when considering how the identity system would be regulated. A well-regulated system with good governance policies is key in adoption of these technologies. Boysen notes that this is a necessary concern for corporations hesitant to adopt these technologies (Boysen. 6). Boysen concludes that while adoption is slow for SSI systems due to lack of regulatory frameworks, decentralized identity systems continue to increase in usage in Canada due to their intrinsic user privacy benefits (Boysen. 8).

Georgy Ishmaev addresses SSI solutions and blockchain in a larger context as an identity management system and the ethical considerations that may apply in his paper (Ishmaev. 2020). Ishmaev raises concerns that both 'experiments on identity can have undesirable consequences' and that 'noble aspirations of its creators will (not) translate into desirable social outcomes' (Ishmaev, 1). Ishmaev believes that the nominalisation of identity wherein personal characteristics is reduced to forensic descriptions carries with it moral concerns and considerations that cannot be removed from the context of development into these tools (Ishmaev. 2). Blockchain from its inception has been related to causes of sovereignty and individual freedom. Ishmaev uses a paper from David Chaum (1985) to argue that a society dependent on computer systems will extend the logic of computer systems outwardly to other realms of social relations to unknown effect (Ishmaev. 2). Nation-state projects, centralized tech giants, are all in a rush to develop digital identity management systems (Ishmaev. 4). Expansion into these spaces adds a technological surveillance aspect to be considered as these entities claim the right to legitimize their expansion into these technological domains. When this occurs, the moral considerations, privacy of data, and cause for existence must be examined when developing these systems. Data breaches, data abuses, and functionality creep are all side effects that will occur through the development of these systems (Ishmaev.

4). Ishmaev proposes decentralized public key infrastructure (DPKI) as a building block to provide an SSI solution on top of. This is due to the novelty of blockchain reducing reliance on centralized identification and authentication authorities (Ishmaev. 6). This is where decentralized identifiers (DID) are proposed again similarly to Hong & Kim (2020). DIDs provide a resource like a URI for referencing a personally identifying endpoint rather than a webpage. However, Ishmaev reviews several papers that conclude that standardization and regulation of digital identity formats is essential in the true adoption of these technologies (Ishmaev. 9). Ishmaev himself concludes that 'persistent identification creep' is a growing concern for the technological space where users are mandated to identify themselves online for participation in a growing digital economy (Ishmaev. 11). SSI systems represent a shift away from centralized data silos but come with their own set of unknown ethical and moral concerns. Ishmaev's main concern is that proposed identity management solutions should evaluate themselves against the possibility of a solution which does not require 'persistent identity' (Ishmaev, 12). He concludes on the need that development into a moral-theoretical framework should be proposed to scrutinize the desirability of identification solutions.

4. DESCRIPTION OF THE PROPOSED WORK

The goal of the proposed research is to investigate the development of a decentralized ledger to store user credentials in a manner compliant to OAuth2. As OAuth2 is an industry standard growing in adoption for authorization, it is necessary for compliance to be achieved for adoption of this technology. Building on top of established protocols lends assurance to the practicality and standardization of the proposed research. The novelty of the research would be to ensure the system would be one backed in Zero Knowledge Proofs which do not leak private information during any transaction of private information. This would create a user-centric model of authentication which lessens power and access centralized solutions have over end user data. The goal of this research is to develop a solution backed in these principles to ensure user access to services compliant with this OAuth2 authorization/authentication portals. The below diagram details how OAuth2 is utilized to issue access tokens which enable authenticated access to protected resources. In this implementation of the technology, the authorization server would be replaced with authorization grants and end user credential data stored on a decentralized ledger. The ledger then would be queried by a resource server or resource owner to determine access and authorization permissions of the requesting client. Authentication to protected resources would be provided over a normal OAuth2 authentication portal.

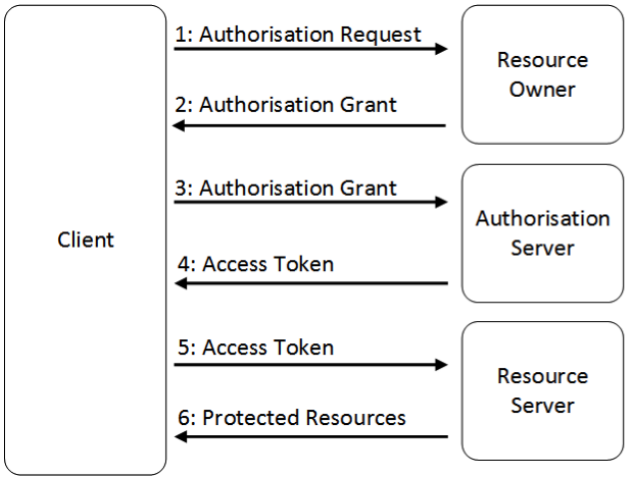


Fig 4. OAuth2 flow diagram of an authorization request which utilizes an access token to provide authentication to protected resources. (Li, 2015)

Compliance with the OAuth2 protocol is essential for standardization of this technology. Articles and media synthesis pieces analyzed for this proposal indicate that lack of regulation and standardization is the largest factor opposing corporate adoption of blockchain technologies. As OAuth2 is used commonly for modern day authentication using a singular digital identity, it would be of great interest to build an identity management system compliant to the OAuth2 protocol. On a note of standardization, we have been able to identify policy on this topic written by the World Wide Web Consortium (W3C). This is referred to as the DID Core v1.0 and manages recommendations on the architecture, data models, and design goals of Decentralized Identifiers (DID) as they are used online. This document will be essential in ensuring compliance of our solution to modern regulatory and standardized frameworks.

In the creation of the self-sovereign identity network Sovrin – which was one of the first consumer grade SSI solutions -, the authors of the network state three principles to ensure a self-sovereign identity process (Tobin, and Reed. 10). These three goals are key design considerations to be kept in mind during the research of our proposed project. Firstly, the identity platform must be secure. This ensures that private identity information is protected, stored persistently, and only contains the minimal amount of identifying information necessary for authentication. Secondly, the user must have controllability over their data. This details that the end user must have visibility into who can access and view their data. This ensures the informed consent from the end user over actions taken with their private data. This must allow the end user to have control over the actions taken using their data. Thirdly, the stated outcome of self-sovereign identity systems is to ensure portability of user credentials and identifying data. This allows the user to have interoperability between services which will request their data or credentials for access. Having portability over where the user can supply credentials gives the user a higher degree of access and transparency into how identifying data is used across services.

As there are several blockchain networks compatible with application development, the one crucial objective would be to choose the ledger solution to build the authentication mechanism on top of. Ethereum and Hyperledger Fabric are two of the most popular blockchain solutions for developers. Both share a wider range of institutional support and have been in existence longer than other new and less-tested platforms. Smart contracts are used to execute applications on top of the ledger network. A smart contract on Ethereum is written in the proprietary language of Solidity whereas Hyperledger Fabric allows for development in Solidity, Node.js, Java, GO, or Python with a wider range of SDKs accessible as well. Hyperledger provides a degree of flexibility in this regard that Ethereum does not. Hyperledger Fabric is also maintained by the Linux Foundation and sees a decent bit of institutional support through IBM. Research should be directed towards gathering data which compares the network throughput and other performance indicators between the two platforms.

The next main objective of this research is to ensure that the end system is privacy-conscious regarding end user data. This is where the analysis of different Zero Knowledge Proof libraries and applications will be researched for implementation. Zero knowledge proofs are implemented to ensure an environment where the end user does not need to supply private personal data to verify authentication. Bulletproofs and Zokrates have been identified as two possible open-source libraries to achieve this if implemented over Ethereum. On the Hyperledger Fabric network the library libsnark is used to implement Zero Knowledge proofs. Hyperledger also contains a library of cryptographic implementations named Hyperledger Ursa which could be used to compute privacy-preserving transactions on the blockchain. It has been identified that solutions and technologies exist to implement a decentralized identity system using the aforementioned technologies. An objective of the research would be to determine the most effective library, platform, and approach to use to.

In determining which platform and libraries work best for the approach, several performance metrics can be used to assess practicality and scalability. Through evaluation of the network is necessary for ensuring that the solution will be effective. Blockchain networks are typically measured in transactions per second (TPS). Further network metrics to consider include the latency of requests and total throughput of concurrent requests.

The figure below details a workflow process in which an end user (credential holder) is able to be issued identifying credentials through an institution (credential issuer). Once issued to an end user the credential record is written and recorded to a decentralized ledger. If multiple institutions are issuing credentials to their end users, it is likely a consortium blockchain technology would be used for interoperability of credentials across services. The credential holder then issues a request to access data from a service. In this process the request is received by the endpoint accessed by the end user, and is then requested for verification on the decentralized ledger.

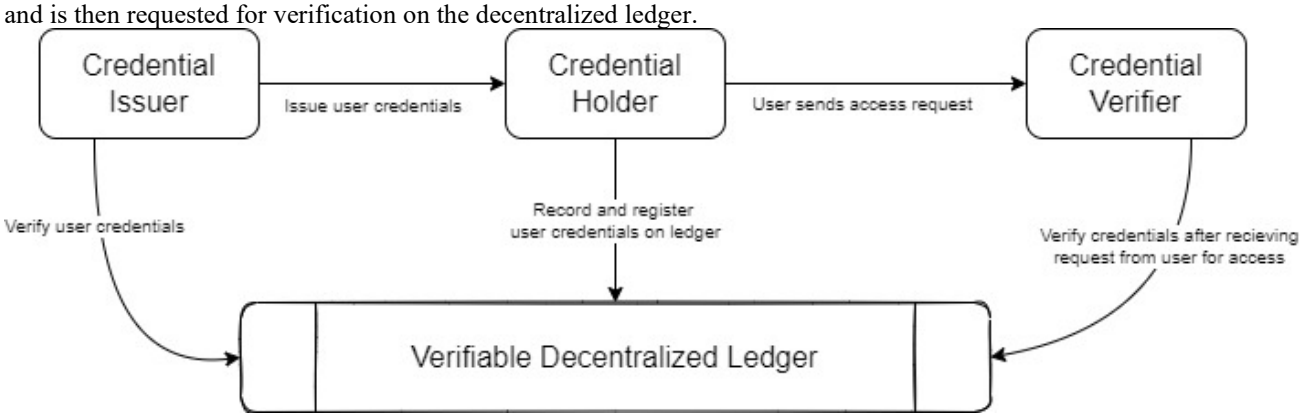


Fig 4. A decentralized identity authentication process

4.1 Approach, tasks, and phases: Details of the approach utilized to achieve each objective of the project are clarified and illustrated **(Table 1)**. Research tasks and activities are divided into groups of assignments, listed in logical sequence, and linked with the project objectives to be achieved (**Table2**).

TABLE 1: APPROACH UTILIZED FOR ACHIEVING OBJECTIVES

Objective	Approach of achieving the objective
1. Research ledger platforms to build the authentication mechanism on top of.	**-Research open-source, free blockchain platforms to develop on -Down to Ethereum, Hyperledger, Corda, and Azure as candidates -Further research into performance and scalability of respective platforms is needed - The result of this research should end in the creation of through design documentation for the platform.**
2. Investigate the development of a decentralized ledger to authenticate users compliant with OAuth2 tokens and credentials.	**-JSON Web Tokens (JWT) development for decentralized identity authentication -Python libraries exist for ensuring compatibility with OAuth2 tokens -Decentralized ledger in place of authorization server to validate credential requests**

3. Ensure that the authentication system is privacy-conscious regarding end user data through development & deployment of zero-knowledge proofs.	**-Determine secure cryptographic architecture for transactions on network. Private/public keys can be used to securely encrypt transaction requests and responses. ECC DSA is a commonly used private key cryptography in blockchain development** **-Registration and authentication of identities need to be secured** **-Libsnark, Bulletproofs, Zokrates, etc. Are identified as possible libraries.** **-Research benchmarks of each library to determine impacts on network throughput** **-Implement privacy-preserving zero knowledge proofs.** **-Test to ensure that de-anonymization of transactions to user address cannot occur. If it can, the system is pseudonymous rather than anonymous.**
4. Develop smart contracts to handle transactions regarding the following use cases: user registration, authentication, OAuth integration, access token generation.	**-Development on either Ethereum through Solidity, or Hyperledger through Python SDK.** **-Smart contracts are to be written for user registration, identity verification, and client/server management.** **-Secure JWT access tokens compliant to OAuth2 and encrypted with ECC-DSA are used to encrypt and secure the transfer of authentication tokens. This will be tied to smart contract development**
5. Develop web authentication portal application to handle requests and authentication flow from the end user to the authenticating service.	**-Research development of OAuth authentication portals** **-NodeJS/JavaScript/HTML is known to project members for web application development** **-If security and configuration of the application is of concern, security testing through Burp Suite could be employed.**
6. Evaluate network metrics, performance indicators to determine scalable indicators.	**-Evaluate performance metrics such as throughput, latency, and transactions per second. Compare metrics of developed platform against transaction metrics of established platforms.** **-Transactions must be scalable and not degrade dramatically with concurrent requests.**
7. Perform user use case testing of software	**-Have multiple users external to the project members use the platform to test useability.** **-Generate user feedback following testing which can be used to indicate changes which should be made to the process** **-User feedback assists us in knowing if solution is easy to use and generally feels more or less of a hassle to authenticate to a service**

TABLE 2: MAPPING OF PHASES AND TASKS TO ACHIEVE OBJECTIVES

Objectives	Phases	Tasks
1. Research ledger platforms to build the authentication mechanism on top of.	**Phase I – Analysis/ Software Requirements**	**Task 1.1:** Review candidate ledger platforms for development **Task 1.2:** Decide on ledger platform for development
2. Investigate the development of a decentralized ledger to authenticate users compliant with OAuth2 tokens and credentials.	**Phase I – Analysis/ Software Requirements**	**Task 1.3:** Determine libraries to provide cryptographic proof **Task 1.4:** Determine libraries for OAuth integration **Task 1.5:** Create document outlining platforms, libraries, software, technologies planned for use in implementation
3. Ensure that the authentication system is privacy-conscious regarding end user data through development & deployment of	**Phase II - Design**	**Task 2.2:** Create design of JSON Web Token Credential Schema **Task 2.3:** Design detailed ledger architecture diagram

zero-knowledge proofs.		
4. Develop smart contracts to handle transactions regarding the following use cases: user registration, authentication, OAuth integration, access token generation.	**Phase II – Design** **Phase III - Development**	**Task 2.1:** Create diagram/pseudocode of proposed smart contracts (Registration, Authentication, OAuth2 request/verification flow). **Task 2.3:** Design ledger architecture diagram **Task 3.1:** Review design goals & specifications **Task 3.2:** Assign project members to development roles **Task 3.3:** Develop code for ledger network **Task 3.4:** Generate ledger wallet addresses for development/testing **Task 3.5:** Develop code for smart contract (registration) **Task 3.6:** Develop code for smart contract (authentication) **Task 3.7:** Develop code for smart contract (Token requests/OAuth integration)
5. Develop web authentication portal application to handle requests and authentication flow from the end user to the authenticating service.	**Phase II – Design** **Phase III - Development**	**Task 2.3:** Design ledger architecture diagram **Task 3.8:** Create web application to initiate access token requests for end user in manner visually alike OAuth portals **Task 5.2:** Host web application on cloud-based infrastructure
6. Evaluate network metrics and performance indicators to determine scalable indicators.	**Phase IV - Evaluation**	**Task 4.3:** Evaluate authentication transaction times (latency, bandwidth, RTT) **Task 4.4**: Security testing and analysis of technical elements of designed platform **Task 5.4:** Test external access to authentication network
Deployment of smart contracts and ledger network	**Phase IV - Evaluation** **Phase V - Deployment**	**Task 4.1:** Deploy ledger network on a private/hybrid blockchain **Task 4.2:** Perform testing of user registration times and functionality **Task 5.1:** Deploy multiple validation nodes for transaction validation to reach consensus on network & increase performance **Task 5.3:** Publish ledger to public/private hybrid blockchain
7. Perform user use case testing of software to generate end user feedback on ease-of-use, and other user experience factors of platform.	**Phase VI – Post Deployment Review**	**Task 6.1:** Allow time for technical issues during deployment or other issues which may arise **Task 6.2:** Identify willing users for possible testing of platform **Task 6.3:** Have users test platform to generate feedback **Task 6.4:** Allow for time to incorporate user feedback into platform **Task 6.5:** Write and issue some documentation on the platform and authentication system

4.2 Research Methodology

To determine the most efficient technologies that can be used in accordance with out group's capabilities, research into individual platforms, technologies, and libraries is essential. A near-term goal of the research would be to concretely determine the most efficient platform to use. Scalability of the solution is a key goal. To do this, a blockchain platform which can handle a high volume of transactions per second should be selected. One that also enables private ledger transactions is also essential as some privacy preserving authentication transactions should be handled off the public ledger to ensure total user privacy. Research into different zero knowledge proof algorithms should also be implemented to determine which libraries ensure the highest degree of performance latency during the process of authenticating an end user. This information is of use in the creation of a detailed system architecture document once the strengths and weaknesses of various platforms and libraries are compared.

Data should also be collected from other similar identity management systems to determine a baseline of transactions per second that the system should be able to process without error. Comparisons should be made to determine what a 'necessary' minimum transactions per second would be for modern identity management systems. Our proposed platform should be able to ensure a similar level of throughput that is necessitated by modern identity management infrastructure. This implements a quantitative approach which measures network data. Throughput will be reduced due to the implementation of zero knowledge proofs which are resource intensive, therefore considerations must be made to determine how a lightweight implementation of these proofs can be used to improve performance. Once a platform is developed and deployed to a private testing network environment, network artefacts and traffic should be collected to analyse data on the viability of the developed platform. This is done to ensure that the platform is scalable to sufficient rates of transactions the system can validate.

Once a platform is developed, it is of great interest to have an end-user testing phase of development. This would provide valuable personal feedback which could be implemented into later versions of the project. Data could be collected through a survey of users following testing with the authentication platform. This data would greatly assist with learning if the end user experience to authenticate has been improved or reduced in friction. This allows for external validation of if we have achieved project goals and where room for improvement can be introduced.

4.3 Management Plan

TABLE 3: ROLE AND INVOLVEMENT DURATION OF RESEARCH TEAM

Team Members	Role
Eric Sedore	Blockchain smart contract development, research into applicable performance metrics, research into zero-knowledge proofs for preserving privacy
Sayyaf Manesiya	Management plan development for next phase proposal, research into technologies outlined in this report

4.4 Project Deliverables:

- A concrete analysis of the scalability and cost of development across several candidate blockchain platforms. These should be compared based on performance metrics, development costs, scalability of solution, and accessibility of documentation. Platforms analysed could include:
  - Ethereum (ETH)
  - Hyperledger Fabric
  - Microsoft Azure Blockchain
  - Corda
- Once a platform is selected, then relevant dependencies and libraries need to be outlined. The ideal platform allows for SDK development in programming languages known to the project members such as Python, Node.js, or Java. Outlined libraries should be chosen for public/private key generation, encryption, and zero-knowledge proof development. These libraries are chosen for implementation of encryption capabilities to ensure secure verification.
- Design of a secure platform architecture detailing how smart contracts can be implemented for all phases of the user identification and authentication process. This also would detail how the architecture of nodes on the testing network reach consensus and validate transactions as they occur. This architecture document would outline the workflow and transactional process of authentication requests across the network as they are initiated by a client, through smart contracts, validated on network nodes, and authenticated by the end service.
- The final deliverable would be the development of the public/private hybrid ledger network to test decentralized identity authentication on. The novelty of the proposed research is in the privacy-preserving nature of implementing zero-knowledge proofs in a manner compliant to OAuth2 authentication. Using known protocols such as OAuth2 will hopefully help in implementing this goal as libraries exist for development with this protocol. This authentication process is to be delivered to the end user through an OAuth2 authentication portal web application.

TABLE 4: PROJECT WORK PLAN

PHASES & TASKS	INVOLVEMENT DURATION	1 M	2 J	3 J	4 A	5 S	6 O	7 N	8 D	9 J	10 F	11 M	12 A
PHASE I: Analysis/Software Requirements													
Task 1.1: Review ledger platforms	**14 days**												
Task 1.2: Decide on ledger platform for development	**1 day**												

Task 1.3: Determine libraries to provide cryptographic proof (Zero-knowledge proofs)	**14 days**												
Task 1.4: Determine libraries for OAuth integration	**7 days**												
Task 1.5: Create document outlining platforms, libraries, software & technologies to use for implementation	**4 days**												
PHASE 2: Design													
Task 2.1: Create diagram/pseudocode of proposed smart contracts. (Registration, Authentication, Requests)	**14 days**												
Task 2.2: Create design of JSON Web Token Credential Schema	**7 days**												
Task 2.3: Design ledger architecture diagram	**14 days**												
PHASE 3: Development													
Task 3.1: Review design goals & specifications	**1 day**												
Task 3.2: Assign project members to development roles	**1 day**												
Task 3.3: Develop code for ledger network	**14 days**												
Task 3.4: Generate ledger wallet addresses for development/testing	**3 days**												
Task 3.5: Develop code for smart contract (registration)	**14 days**												
Task 3.6: Develop code for smart contract (authentication)	**14 days**												
Task 3.7: Develop code for smart contract (Token requests/OAuth integration)	**14 days**												
Task 3.8: Create web application to initiate access token requests for end user in manner visually similar to OAuth2 portal	**14 days**												
PHASE 4: Evaluation													
Task 4.1: Deploy ledger network on a private/hybrid blockchain	**3 days**												
Task 4.2: Perform testing of user registration times & functionality	**5 days**												
Task 4.3: Evaluate authentication transaction	**10 days**												

times (latency, bandwidth, RTT)													
Task 4.4: Security testing and analysis of technical elements of designed platform	**14 days**												
PHASE 5: Deployment													
Task 5.1: Deploy multiple validation nodes for transaction validation to reach consensus on network & increase performance	**5 days**												
Task 5.2: Host web application on cloud-based infrastructure	**10 days**												
Task 5.3: Publish ledger to public/private hybrid blockchain	**10 days**												
Task 5.4: Test external access to authentication network	**3 days**												
PHASE 6: Post-Deployment Review													
Task 6.1: Allow time for technical issues during deployment or other issues which may arise	**14 days**												
Task 6.2: Identify willing users for possible testing of platform	**5 days**												
Task 6.3: Have users test platform to generate feedback	**14 days**												
Task 6.4: Allow for time to incorporate user feedback into platform	**5 days**												
Task 6.5: Write and issue some documentation on the platform and authentication system	**14 days**												

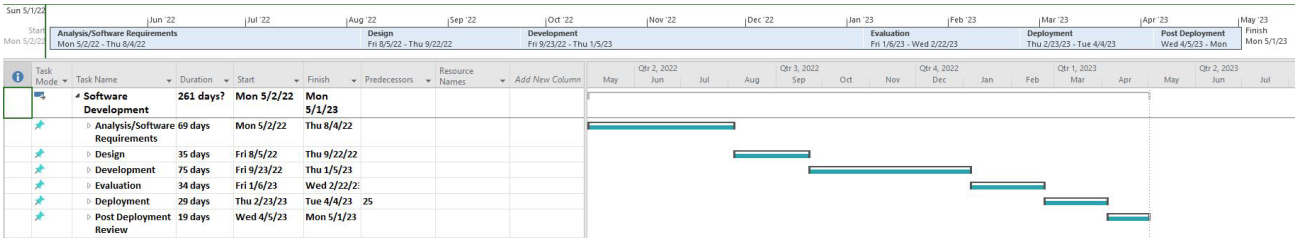


Fig 5. Timeline of the GANTT chart designed in relative accordance with an SDLC (Software Development Lifecycle) lifecycle. All tasks included for each phase are outlined above in the work plan. The fully expanded GANTT chart detailing tasks and phases is accessible in this report following the below bar chart.

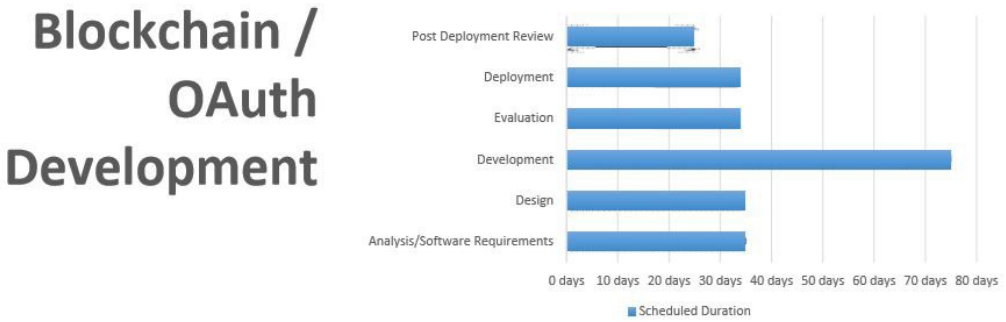


Fig 6. Bar chart outlining time allotted to individual phases of the development lifecycle for this project. Development is the longest phase allotted for this project.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
0		**Software Development**	**261 days?**	**Mon 5/2/22**	**Mon 5/1/23**		
1		**Analysis/Software Requirements**	**69 days**	**Mon 5/2/22**	**Thu 8/4/22**		
2		Review ledger platforms	14 days	Mon 5/2/22	Thu 5/19/22		Analyst
3		Decide on ledger platform for development	1 day	Fri 5/20/22	Fri 5/20/22	2	Analyst
4		Determine libraries to provide cryptographic proof	14 days	Mon 5/23/22	Thu 6/9/22	3	Analyst
5		Determine libraries for Oauth integration	4 days	Fri 6/10/22	Fri 7/29/22	4	Analyst
6		Create document outlining platforms, libraries, sotfware, & technologies used in implementatio	4 days	Mon 8/1/22	Thu 8/4/22	5	Analyst

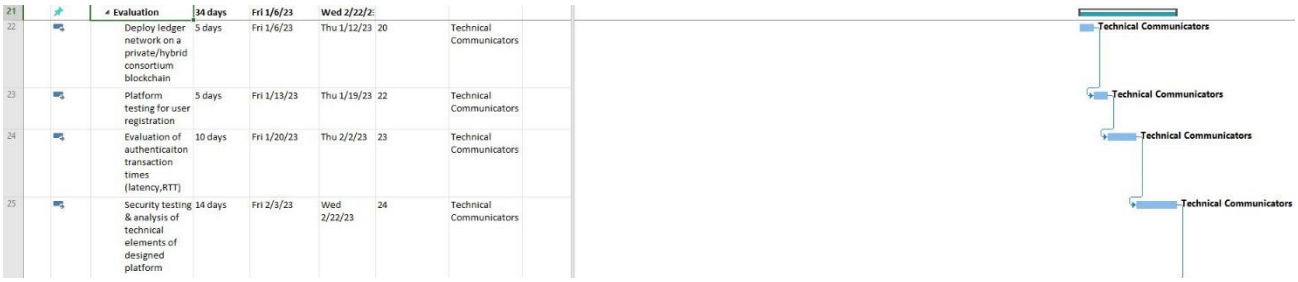
Fig 7. Phase I of the outlined development lifecycle on the GANTT chart. This covers the tasks that comprise the start of the ‘Analysis/Software Requirements’ phase.

7		**Design**	**35 days**	**Fri 8/5/22**	**Thu 9/22/22**		
8		Create diagram/pesud of proposed smart contracts	14 days	Fri 8/5/22	Wed 8/24/22	6	Analyst
9		Create design of JSON Web Token Schema	7 days	Thu 8/25/22	Fri 9/2/22	8	Analyst
10		Design ledger architecture diagram	14 days	Mon 9/5/22	Thu 9/22/22	9	Analyst
11		Design complet	0 days	Thu 9/22/22	Thu 9/22/22	10	

Fig 8. Then follows Phase II of the development lifecycle which covers the ‘Design’ phase and it’s outlined tasks.

12	Development	75 days	Fri 9/23/22	Thu 1/5/23		
13	Review design goals & specifications	1 day	Fri 9/23/22	Fri 9/23/22	11	Developer
14	Assign project members to development roles	1 day	Mon 9/26/22	Mon 9/26/22	13	Developer
15	Develop code for ledger network	14 days	Tue 9/27/22	Fri 10/14/22	14	Developer
16	Generate ledger wallet addresses for	3 days	Mon 10/17/22	Wed 10/19/22	15	Developer
17	Develop code for smart contract (user registration)	14 days	Thu 10/20/22	Tue 11/8/22	16	Developer
18	Develop code for smart contract (Authentication	14 days	Wed 11/9/22	Mon 11/28/22	17	Developer
19	Develop code for smart contract (Token requests/Oauth Integration)	14 days	Tue 11/29/22	Fri 12/16/22	18	Developer
20	Web application development for Oauth portal	14 days	Mon 12/19/22	Thu 1/5/23	19	Developer

Fig 9. The above screenshot details the tasks which comprise Phase III of the development lifecycle. This covers the tasks under the Development phase. This is where a bulk of the time allotted to this project is expected to be spent.



21	Evaluation	34 days	Fri 1/6/23	Wed 2/22/2		
22	Deploy ledger network on a private/hybrid consortium blockchain	5 days	Fri 1/6/23	Thu 1/12/23	20	Technical Communicators
23	Platform testing for user registration	5 days	Fri 1/13/23	Thu 1/19/23	22	Technical Communicators
24	Evaluation of authenticaiton transaction times (latency,RTT)	10 days	Fri 1/20/23	Thu 2/2/23	23	Technical Communicators
25	Security testing & analysis of technical elements of designed platform	14 days	Fri 2/3/23	Wed 2/22/23	24	Technical Communicators

Fig 10. Following this is Phase IV which covers 'Evaluation' of the developed project.

26	Deployment	29 days	Thu 2/23/23	Tue 4/4/23	25	
27	Deploy multiple validation nodes to increase consensus time	5 days	Thu 2/23/23	Wed 3/1/23	25	Deployment Team
28	Host web application on cloud based infrastructure (Azure)	14 days	Thu 3/2/23	Tue 3/21/23	27	Deployment Team
29	Publish ledger to public/private consortium blockchain	5 days	Wed 3/22/23	Tue 3/28/23	28	Deployment Team
30	Test external access to authentication network	5 days	Wed 3/29/23	Tue 4/4/23	29	Deployment Team

Fig 11. The above screenshot details the tasks which comprise the Deployment phase of the development lifecycle.

31	Post Deployment Review	19 days	Wed 4/5/23	Mon 5/1/23		
32	Identify willing users for testing of platform	5 days	Wed 4/5/23	Tue 4/11/23	30	Project Manager
33	Have users test platform to generate feedback	5 days	Wed 4/12/23	Tue 4/18/23	32	Testers
34	Allow for time to incorporate user feedback into platform	4 days	Wed 4/19/23	Mon 4/24/23	33	Developer
35	Write & Issue Documentation on the platform & authentication system	5 days	Tue 4/25/23	Mon 5/1/23	34	Technical Communicators

Fig 12. The final tasks of the Post-Deployment Review phase are outlined above. This concludes all tasks outlined for the GANTT chart outlining the phases and tasks for the development of the proposed project.

5. REFERENCES

Alexandra Sava, Justina. “Global Digital Identity Solution Market Value 2020 and 2026.” *Statista*, July 2021, www.statista.com/statistics/1263580/worldwide-digital-identity-solution-market-revenue.

Argento, Luciano, et al. “ID-Service: A Blockchain-Based Platform to Support Digital-Identity-Aware Service Accountability.” Applied Sciences, vol. 11, no. 1, MDPI AG, 26 Dec. 2020, p. 165. Crossref, doi:10.3390/app11010165.

Boo, EunSeong, et al. “LiteZKP: Lightening Zero-Knowledge Proof-Based Blockchains for IoT and Edge Platforms.” IEEE Systems Journal, Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 1–12. Crossref, doi:10.1109/jsyst.2020.3048363.

Boysen, Andre. “Decentralized, Self-Sovereign, Consortium: The Future of Digital Identity in Canada.” Frontiers in Blockchain, vol. 4, Frontiers Media SA, 29 Apr. 2021. Crossref, doi:10.3389/fbloc.2021.624258.

Burt, Tom. “Microsoft Report Shows Increasing Sophistication of Cyber Threats.” *Microsoft On the Issues*, 5 May 2021, blogs.microsoft.com/on-the-issues/2020/09/29/microsoft-digital-defense-report-cyber-threats.

Cha, Shi-Cho, et al. “Enhancing OAuth with Blockchain Technologies for Data Portability.” IEEE Transactions on Cloud Computing, Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 1–1. Crossref, doi:10.1109/tcc.2021.3094846.

Fremery, Rose de. “Breaking the Cycle of Password Reuse.” *The LastPass Blog*, 26 Aug. 2021, blog.lastpass.com/2021/09/breaking-the-cycle-of-password-reuse.

Figueroa-Lorenzo, Santiago, et al. “Modbus Access Control System Based on SSI over Hyperledger Fabric Blockchain.” Sensors, vol. 21, no. 16, MDPI AG, 12 Aug. 2021, p. 5438. Crossref, doi:10.3390/s21165438.

Gisolfi, Dan. “Decentralized Identity: An Alternative to Password-Based Authentication.” *IBM Supply Chain and Blockchain Blog*, 27 Feb. 2020, www.ibm.com/blogs/blockchain/2018/10/decentralized-identity-an-alternative-to-password-based-authentication.

Goodell, Geoff, and Tomaso Aste. “A Decentralized Digital Identity Architecture.” *Frontiers in Blockchain*, vol. 2, 5 Nov. 2019, 10.3389/fbloc.2019.00017. Accessed 6 Dec. 2019.

Hong, Seongho, and Heeyoul Kim. “VaultPoint: A Blockchain-Based SSI Model That Complies with OAuth 2.0.” Electronics, vol. 9, no. 8, MDPI AG, 31 July 2020, p. 1231. Crossref, doi:10.3390/electronics9081231.

Ishmaev, Georgy. “Sovereignty, Privacy, and Ethics in Blockchain-Based Identity Management Systems.” Ethics and Information Technology, vol. 23, no. 3, Springer Science and Business Media LLC, 30 Nov. 2020, pp. 239–252. Crossref, doi:10.1007/s10676-020-09563-x.

Johnson, Jospeh. “Global Consumers Opinion on Personal Data Control by Tech Companies 2021.” *Statista*, Statista, 2 Nov. 2021, www.statista.com/statistics/1233743/global-consumers-opinion-tech-personal-data.

Li, Wanpeng, and Chris J. Mitchell. “Addressing Threats to Real-World Identity Management Systems.” *ISSE 2015*, 2015, pp. 251–59. *Crossref*, https://doi.org/10.1007/978-3-658-10934-9_21.

Mir, Omid, et al. “Decentralized, Privacy-Preserving, Single Sign-On.” Security and Communication Networks, edited by David Meghias , vol. 2022, Hindawi Limited, 22 Jan. 2022, pp. 1–18. Crossref, doi:10.1155/2022/9983995.

Mulaji, Sarah S. M., and Sumarie S. Roodt. “The Practicality of Adopting Blockchain-Based Distributed Identity Management in Organisations: A Meta-Synthesis.” Security and Communication Networks, edited by Yinghui Zhang , vol. 2021, Hindawi Limited, 28 Nov. 2021, pp. 1–19. Crossref, doi:10.1155/2021/9910078.

Statista. “Global Blockchain Solutions Spending 2017–2024.” *Statista*, 18 Mar. 2022, www.statista.com/statistics/800426/worldwide-blockchain-solutions-spending.

Statista. “Use Cases for Blockchain Technology in Organizations Worldwide 2021.” *Statista*, 18 Mar. 2022, www.statista.com/statistics/878732/worldwide-use-cases-blockchain-technology.

Szalachowski, Pawel. “Password-Authenticated Decentralized Identities.” IEEE Transactions on Information Forensics and Security, vol. 16, Institute of Electrical and Electronics Engineers (IEEE), 2021, pp. 4801–4810. Crossref, doi:10.1109/tifs.2021.3116429.

Tobin, Andrew, and Drummond Reed. "The Inevitable Rise Of Self-Sovereign Identity". *Sovrin*, 28 Mar. 2017, https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf.