EASEID- A SESSION-BASED SINGLE SIGN-ON SELF-SOVEREIGN IDENTITY AND ACCESS MANAGEMENT SYSTEM USING BLOCKCHAIN

Goluguri Satyanarayana Reddy

Research Scholar, Department of Information Technology, GITAM University,
Visakhapatnam. AP, India - 530045
Assistant Professor, Department of Information Technology,MVGR College of engineering
Chintalavalasa, Vizianagaram, AP- 535005
gsnreddy125@gmail.com,
https://www.mvgrce.com/faculty-of-mvgr/?dept=IT&fid=4

Dr.Thammi Reddy Konala

Professor, Computer Science and Engineering, GITAM Institute of Technology, GITAM University,
Rushikonda, Visakhapatnam, 530045, AP, India.
tkonala@gitam.edu

https://vspgitcse.gitam.edu/faculty/profile/1205

Abstract

To gain easy access to internet-based services, digital identity and access management systems may be employed. An identity management system based on blockchain can offer access to all of the services required by the user without jeopardizing their privacy or security. Users may have complete control over who the individual has access to their credentials. Service providers' validation will become simpler and quicker. The article gives an overview of blockchain, self-sovereignty, digital identity, and access management, as well as a recommended approach EASEID for a Session-based Single sign-on self-sovereign Identity and access management model built on the blockchain platform overcoming the vulnerabilities in the existing systems like Google, Facebook, etc. for gaining access to a variety of online services.

***Keywords*: identity and access management, blockchain, single sign-on, self-sovereign.**

1. Introduction

Recognizing a particular entity like a person, or an object can be done by using the attributes of that entity. These collective attributes are called an identity which can be used to determine who or what. In the digital world, to access any services based on the internet, identity plays a vital role. Managing, Authorizing, and validating identity to access any services offered by the service providers can be termed as identity and access management(IDAM) [1]. It is a common use case to allow authorized users to access digital resources. The evaluation of this identity and access management can be marked into 3 phases [2][3]. 1) Isolated centralization 2) monopolistic centralization 3) decentralization.

1) Isolated centralization of implementing IDAM is fully centralized as the user's identity and access information are stored, maintained, and managed by centralized servers controlled by centralized authorities which are usually the service providers shown in Figure 1. As with the increasing online services, for a user to access multiple services need to have multiple unique identities, one identity for accessing a service which is not user-friendly. This implementation even burdens the service providers as they need to build separate registration and authentication systems to allow valid users to access their service. Most importantly the user's private data is under the control of service providers which might lead to security issues and it's based on the trust, users need to provide their data to service providers and they don't have any control over their subsequent usage of the information[2].

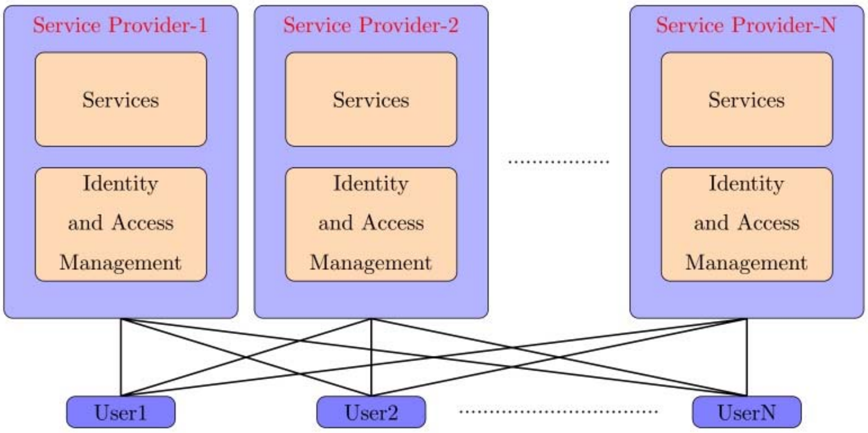

Figure 1 isolated centralization of implementing IDAM.

2) Monopolistic centralization of implementing IDAM is to some extent improve the concerns raised in isolated centralization w.r.t ease of accessing services. This method allows single login to access multiple services. Ex: - Google, Facebook, etc.[4]. These service providers, with massive users, can provide identity authorization to access any other services on the internet, based on the acceptance of the specific services provider which is shown in Figure 2. This leads to raising concerns like a single-point failure and security vulnerability. Misuse of private data is a huge concern that is purely dependent on the trust of these identity authorization providers. Facebook data leakage sets an alarming example of this concern[2].

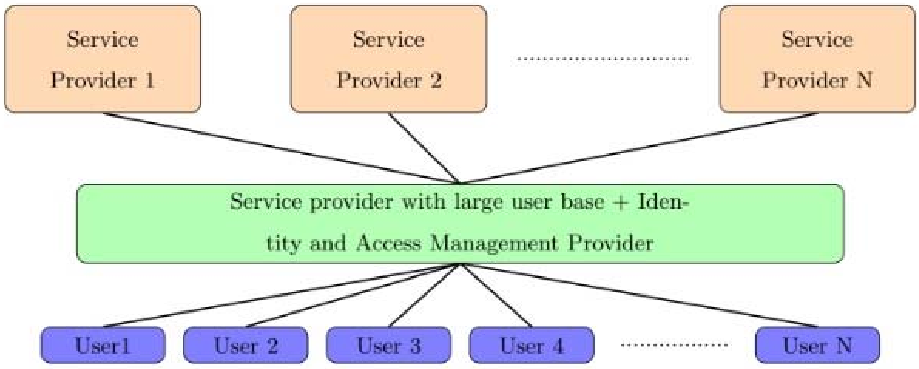

Figure 2 monopolistic centralization of implementing IDAM.

3) Decentralization of implementing IDAM would be the preferred way as it could address the concerns related to misuse of private data by identity authorization service providers which is shown in Figure 3. Individual users have the ability to self-sovereign their data, which allows them to select when and how they reveal their identity. [2].

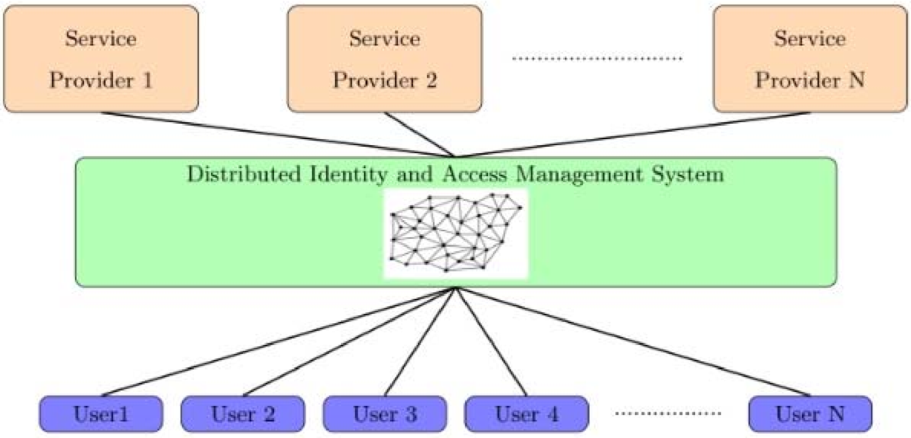

Figure 3 Decentralization of implementing IDAM.

The decentralized way of implementing IDAM can be the best possibility by using blockchain technology.

The proposed model makes the following contributions: 1) EASEID - A Session-based Single Sign-on Self-Sovereign Distributed Identity and Access Management System which is represented in Figure 4 a system that may be used for authentication to access any online services that the user requires during a session. The EASEID's nodes were supposed to be users and service providers. 2) It assesses and summarize the design considerations in the existing popular IDAM's (Sovrin, uPort, ShoCard) which are built using blockchain and their adaptability to be used per session authentication by being self-sovereign.

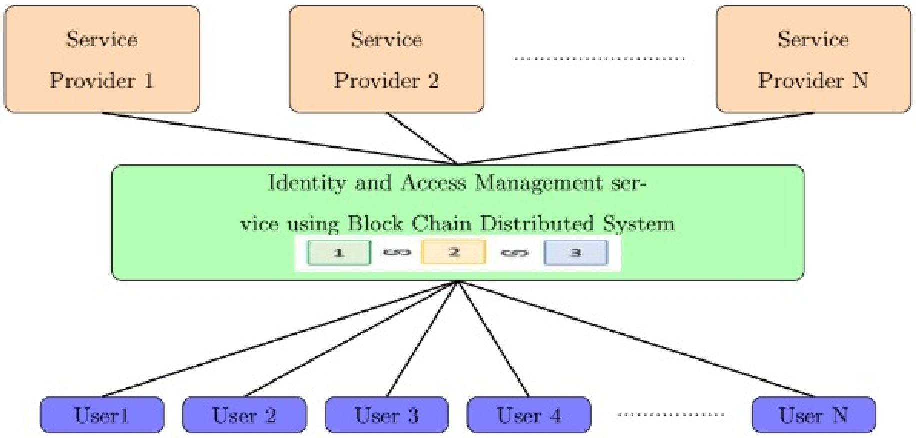

Figure 4 Blockchain implementing IDAM.

2. Preliminaries

2.1. Identity and identity Management System

2.1.1. Identity

Identity is defined as "the collection of characteristics by which a person or item is readily identifiable or known," as well as "the fact or state of being the same as something else". Subjects are entities that have the ability to interact with objects. "An object is a passive entity that a person may interact with. Subjects are assigned labels in order to keep track of the information that is available about them" [1]. The characteristics of an entity are described via attributes. A discrete and quantifiable name-value characteristic of an entity is called an attribute[5][3].

2.1.2. Identity management (IDAM)

Identity management refers to the administrative process of creating and managing user accounts for usage in online services such as authentication and identification [6][3]. It is necessary to streamline the user provisioning process and guarantee that only authorized users have access to services[7].

2.2. Access and access management

2.2.1. Access

Access refers to the capacity, right, or permission to approach, as well as the state or attribute of being approachable[3].

2.2.2. Access management

An access management system is a set of tools, methods, and rules for managing individual identities, such as authentication, authorization, privileges and roles, both within and outside of organisations [5][3].

2.3. Self-sovereignty

Without the intervention of administrative authorities, an entity should own and govern its identity.

According to Allen[8], The following 10 principles define an SSI system:

i) Access-"Users must have access to their own data."

ii) Consent - "Users must consent to their identities being used."

iii) Control - "Users must be able to manage their own identities."

iv) Existence - "Users must be able to exist independently."

v) Interoperability - "Identities should be usable by as many people as feasible."

vi) Minimization - "Claim disclosure must be kept to a minimum."

vii) Persistence - "Identities must endure for a long time."

viii) Portability - "Information and services regarding identification must be transportable,"

ix) Protection - "User rights must be safeguarded."

x) Transparency-"Systems and algorithms must be transparent."

2.4. Blockchain

The blockchain, which was first introduced as Bitcoin, is a peer-to-peer network that ensures transaction transparency by consensus. Because of its immutability and consensus function, blockchain eliminates the need for a central authority, making it an excellent option for a dispersed setting[6]. Because data being the most important asset, blockchain's application in data-driven architecture may provide decentralization, anonymity, audibility, and persistence.

2.4.1. Node and Block

A peer-to-peer network node is a computer that represents the owner of transactions carried out by a certain user. [6].

In the blockchain, a block is an immutable entry of a distributed ledger. To the blockchain, a block is added When a transaction is approved[6].

2.4.2. Consensus

Through the acceptance of node decisions, the consensus method is employed for processing and validating the transaction. “Proof-of-work” and “proof-of-stake” are two widely used consensus algorithms[6].

2.4.3. Smart Contract

Smart contracts are computer programmes that run when specific conditions are met and are stored on a blockchain. They're frequently used to automate the execution of a contract so that both parties may be certain of the outcome at the outset, without the need for a middleman.

3. IDAM Frameworks using blockchain

We'll focus on three IDAM schemes based on DLT: Sovrin, uPort, and ShoCard. These schemes were chosen because, they provide as important examples of common design issues and decisions in their respective genres, as well as serving a comparable purpose for the greater DLT-based IDAM environment. They've also released the most technical details about their scheme concepts, and they're either backed by large online communities or have a significant amount of venture capital backing[9][1].

3.1. Sovrin

Sovrin's major purpose is inclined to provide people with complete jurisdiction over all elements marked in their digital identity. Each user may choose which attribute credentials are published and who has access to them[9][10].

Sovrin is a public service focused to blockchain identity management that is open-source. It's designed on a permissioned blockchain, which means that only authorized nodes may participate in the consensus process, but anybody will be able to use it to perform transactions. By employing a permissioned blockchain, The Sovrin can gain increased efficiency in reaching consensus, a higher transaction rate, and a lower vulnerability to 51 percent attacks. The Sovrin is intimately linked to the Hyperledger-Indy blockchain, which is now maintained by the Linux Foundation, and Evernym, a platform resolute to products and services based on Sovrin identification. Sovrin uses ZKP[7] for all confirmed identities, they claim to minimise data exposure to a bare minimum, and that it is possible to convey selected attributes linked with identity without exposing the credentials. Sovrin also tries to avoid correlation by separating data from identifiers, making it hard to associate an identity without additional data kept separately. [11]. The Sovrin Network employs two node rings to achieve great scalability: validator nodes ring that accept write transactions, as well as a much bigger ring of observer nodes that can process requests for read using read-only copies of the blockchain[9][10].

3.2. Uport

With Uport, an open identity system, users may register using their own identity on Ethereum blockchain, they can request credentials and transmit, securely manage keys and data, and sign transactions. For each identity, the uPort mobile app generates three smart contracts and a key pair. A Proxy Contract acts as the user's unique identification, while a Controller Contract allows identity access and a Recovery Quorum Contract assists in the restoration of a user's identity if they lose it. Identity owners must appoint trustees for key restoration, who will use the Controller Contract to call a vote to generate a new public key; if a quorum is reached, the controller replaces the misplaced public key with a newly nominated key via a customised proxy function.[9][12].

3.3. Shocard

ShoCard is a public Bitcoin blockchain-based digital identification and authentication platform. The blockchain stores user identifying information in the form of signed cryptographic hashes. The blockchain is used to verify such information and certify the identification of third parties that have validated the user's identity. There is no central location or storage for users' private information, and elements of a user's identify do not need to be shared across multiple services to authenticate or confirm account ownership. The user performs an identity verification transaction with a third party on the blockchain. The data is encrypted and kept in a secure data envelop that only the receiver can decipher. When both IDs have been validated, the transaction may commence. The system can create a publicly verifiable blockchain with five million user entries in 30 minutes [9].

4. Proposed model- EASEID

With the user's approval, digital identity systems must only release information that identifies a user. For public and private entities to utilize, a global identification system must offer both "Omni-directional" and "unidirectional" IDs. The long-term approach that discloses the lowest amount of identifying details and limits its use is the most stable. A global identity system must channel and permit multiple identifying technologies managed by numerous identity suppliers. With explicit human-machine communication channels, the human user must be an integrated component of the distributed system. The unified identity metasystem must give users with a straightforward, consistent experience while permitting context separation via various operators and technologies.[13].

The EASEID system describes session sign-on authentication IDAM framework adopting self-sovereignty principles.

Identity and Authentication Providers, Service Providers, and Users are the players engaged in the life cycle. Enrolment, authentication, issuance, and verification are the four stages of the identity management system life cycle.

4.1. Players

4.1.1. Role of Identity and Authentication Provider

By validating both sides' credentials, identity and authentication providers must make it simple for users and service providers to establish identities. It is also in charge of establishing a session for each user when their credentials have been confirmed. It should be easy to remove an entity, a user, or a service provider but only with their concern. Valid data in an existing entry should be updatable. Tokens for use with the services of service *providers* should be able to be created. The identities of users and service providers should be checked, and omnidirectional access to required data should be allowed only with the agreement of the associated identity owner.

4.1.2. Role of Service Provider

The paradigm ensures that all genuine IDAM users are treated as clients of legitimate service providers. Service providers must register with the IDAM and display the requisite criteria in order to become a registered service provider entity with the IDAM. All users with valid tickets should be able to use the registered services of service providers. This saves users time and money by eliminating the need to re-register with different service providers. Service providers are not required to keep a database of users and authenticate them every time they choose to access it. The service providers will be providing access to all valid service providers.

4.1.3. Role of User

Users must register with the service provider by providing proper information. After successful entry, they will become authorized users. Before being permitted to utilize the services of any registered service provider, they must first be authenticated by the IDAM.

By signing in once per session, users may access all services from numerous service providers who are genuine IDAM companies.

This looks to be a centralized approach for implementing IDAM, However, the underlying system is a blockchain, which is a distributed system that provides tremendous security and distributes data across several nodes, preventing a single point of failure.

4.2. Stages of the EASEID system life cycle.

4.2.1. Enrolment

Users and service providers should submit their characteristic attributes to the IDAM in order to produce a unique identity that will be recognized as enrolling.

4.2.1.1. Users Enrolment

The following are some basic approaches for determining a user's identification that may be used independently or in combination:

- A password, a personal identification number (PIN), or the answers to a series of pre-determined questions. Cryptographic keys, smart cards, electronic key cards, and physical keys are examples of privately held items. A token is the name for this sort of authenticator.
- Static biometrics: fingerprint, retina, and face recognition are examples of static biometrics.
- Something that a person does (dynamic biometrics): speech pattern recognition, handwriting characteristics, and typing rhythm are all instances of dynamic biometrics.

User enrolment is done by providing user encrypted attributes which are used to generate unique identity.

4.2.1.2. Service Providers Enrolment

Service providers provide the attributes like service details, company details whichever can be specified as their characteristics. These are used to generate a unique identity

4.2.2. Authentication

It is the process of confirming a system entities or its own stated identification. Users and service providers are to be authenticated for accessing a service or for providing a service.

4.2.2.1. User Authentication

Each user is authenticated once per session. User authentication is invoked by the user whenever he wants to access services. By submitting his credentials, the user must seek access to services. After authentication, the user should be a legal entity to access any registered services.

4.2.2.2. Service Provider Authentication

By revealing his name, the service provider should be verified. Service provider authentication is invoked by the service provider whenever the service provider is ready to provide services. The session per service provider is generally long, and when it ends, it indicates that the service is down or that the service provider is temporarily or permanently suspending his services.

4.2.3. Issuance

The identity is issued to users and service providers in two phases:

- After enrolment, the identity is issued to users and service providers.
- When a user logs in, they are issued with a session token.

- After service provider authentication, Access keys are issued to service providers.
- Users are issued with tokens to access all services.

4.2.4. Verification

The process of verification is to validate the data that's received. User verification, service provider verification, Session token verification, and ticket verification are done at different stages throughout the IDAM transactions.

4.3. Smart Contracts

4.3.1. Identity Contract

The identity contract accepts all of the entities' encrypted information, develops a unique identity, and returns it to the entity, which may subsequently be used for authentication to access any services.

4.3.2. Authentication Contract

The authentication contract accepts identity from the entities and on successful verification, Sends the attributes to the session contract.

4.3.3. Session Contract

The session contract creates the access key pair for service providers or session key for users by using all the credentials provided by the authentication contract. The session key generated is valid till the entity logout.

4.3.4. Ticket Contract

After receiving a request from a user to produce a token, the Ticket Contract generates a ticket that is used as a gate pass to access all of the services.

4.4. Notations

U-User.
SP-Service Provider.
BC-Blockchain.
IC-Identity Contract.
AC-Authentication Contract.
SC-Session Contract.
TC-Ticket Contract.
IC-Identity Contract.
AC-Authentication Contract.
SC-Session Contract.
TC-Ticket Contract.
PU_{SP}*-Public Key of "SP".*
PR_{SP}*-Private Key of "SP".*
$Attributes_{SP}$*-Attributes of "SP".*
PU_{IC}*-Public Key of EASEID's "IC".*
ID_{SP}*-Unique identity of "SP".*
PU_{AC}*-Public Key of "AC".*
PU_{TC1}*- First public key of "TC" known to "U" and "SP".*
PU_{TC2}*- Second public key of "TC" known only to "SP".*
$Access_{PU_{SP}}$*-Access public key of "SP".*
$Access_{PR_{SP}}$*-Access private key of "SP".*
PU_{User}*-Public Key of "U".*
PR_{User}*-Private Key of "U".*
$Attributes_{User}$*-Attributes of "U".*
ID_{User}*-Unique identity of "U".*
$K_{User,sc}$*-Session key between "U" and "SC".*
Ticket-Used for accessing any services required by the "U".
PR_{TC2}*-Second private key of "TC".*
$\cup_{i=1}^{n} Access_{PU_{sp_i}}$*-Set of "SP's" access public keys.*

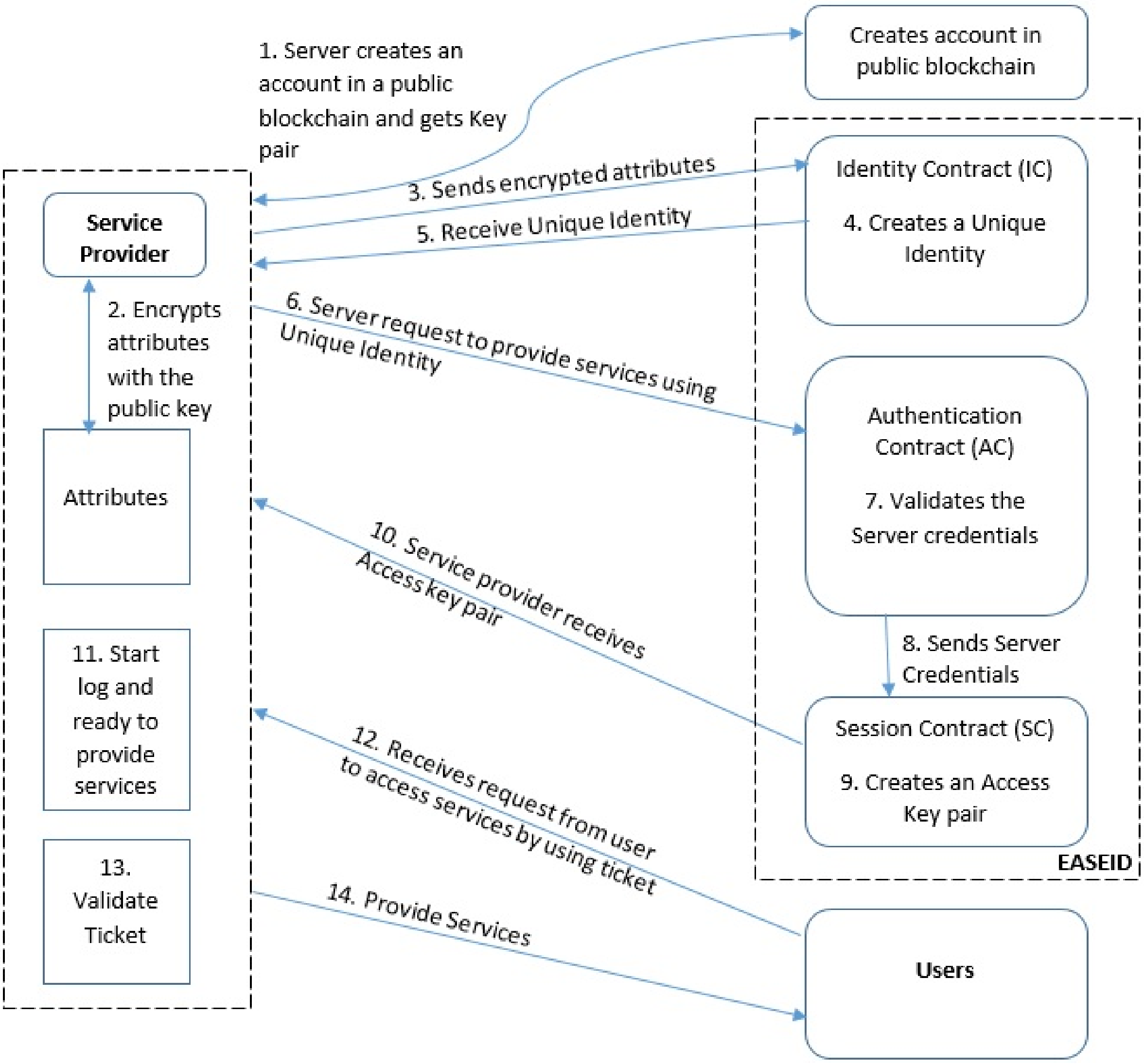
1. Server creates an account in a public blockchain and gets Key pair
Creates account in public blockchain
Service Provider
2. Encrypts attributes with the public key
Attributes
3. Sends encrypted attributes
5. Receive Unique Identity
Identity Contract (IC)
4. Creates a Unique Identity
6. Server request to provide services using Unique Identity
Authentication Contract (AC)
7. Validates the Server credentials
8. Sends Server Credentials
10. Service provider receives Access key pair
Session Contract (SC)
9. Creates an Access Key pair
EASEID
11. Start log and ready to provide services
12. Receives request from user to access services by using ticket
13. Validate Ticket
14. Provide Services
Users

Figure 5 EASEID Model – Service Provider Interaction

4.5. EASEID-Service Provider interaction specification in Figure 5:

1. As a result of the creation of an account in the existing public blockchain, the service provider receives a key pair.

$$\text{BC} \rightarrow \text{SP: } (PU_{sp}, PR_{sp})$$

2. The service provider chooses all of his attributes, which may be used to define all of the service and service provider characteristics, then encrypts them using his public key.

$$E\big(PU_{sp}, [Attributes_{sp}]\big)$$

3. The service provider requests a unique identity by submitting his encrypted characteristics to EASEID's identity contract.

$$\text{SP} \rightarrow \text{IC: } E(PU_{IC}, [E(PU_{sp}, Attributes_{sp})||PU_{sp}])$$

4. The identity contract uses the service provider's encrypted information and the current state of the system to generate a unique identity.
5. By encrypting the unique identity with the service provider's public key, the identity contract transfers it to the service provider.

$$\text{IC} \rightarrow \text{SP: } E(PU_{sp}, [ID_{sp}])$$

6. When the service provider is ready to deliver the service to users, it uses its ID to request a session key from the EASID's authentication contract.

$$\text{SP} \rightarrow \text{AC: } E(PU_{AC}, [ID_{sp}||PU_{sp}])$$

7. Authentication contract validates the identity of the service provider.
8. Authentication contract sends the credentials of the service provider to the session contract.

9. Session contract creates an Access key pair.
10. Session contract sends this Access key pair to the service provider.

$$\text{SC}\rightarrow\text{SP: } E(PU_{sp},[(Access_{PU_{sp}},Access_{PR_{sp}})||PU_{TC1}||PU_{TC2}])$$

11. Service provider decrypts the Access key pair, starts log.
12. Service provider is now waiting for user requests to provide services.
13. On receiving the request from the user the service provider validates the ticket.
14. Starts providing services to the user.

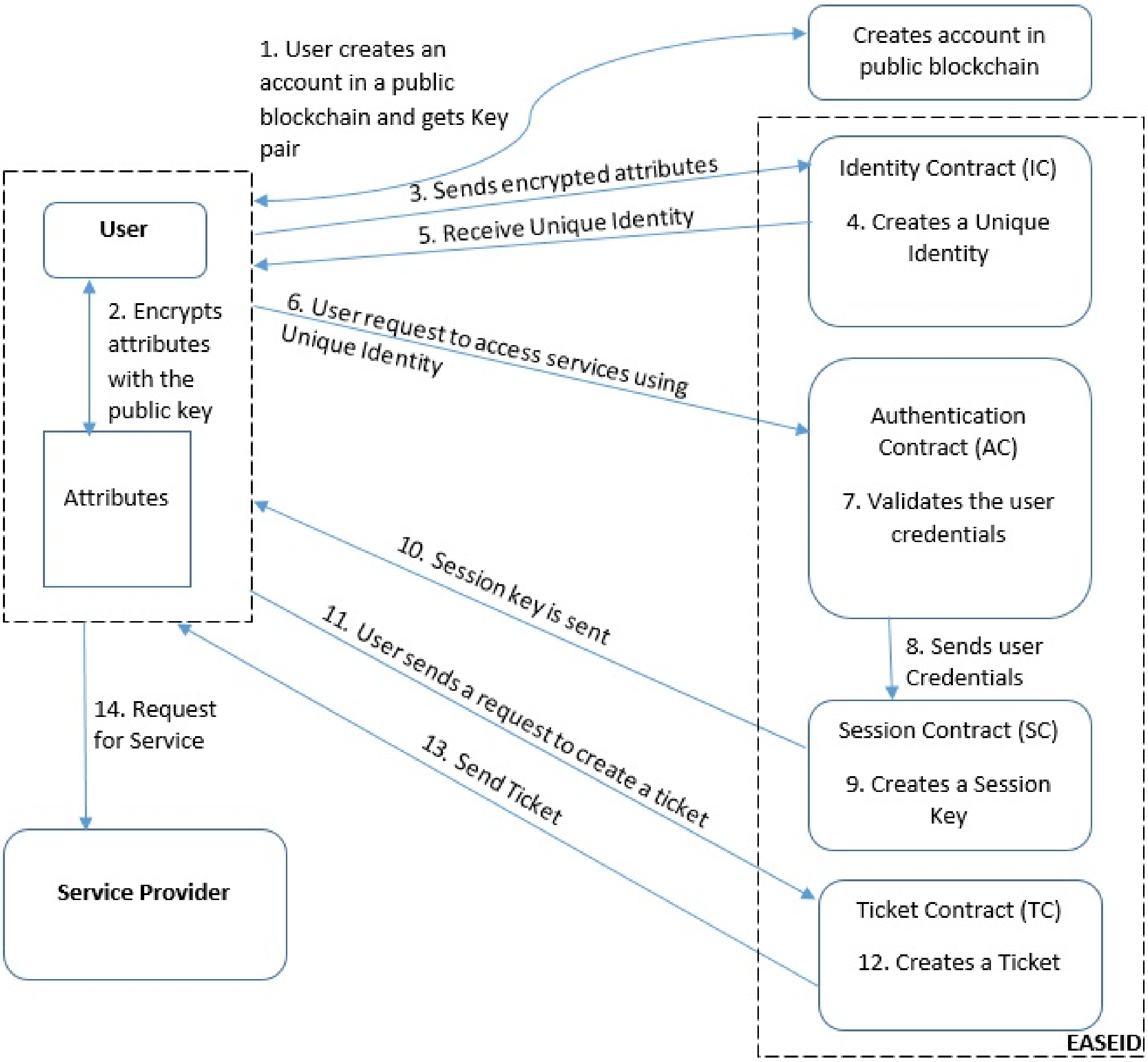
1. User creates an account in a public blockchain and gets Key pair
Creates account in public blockchain
User
2. Encrypts attributes with the public key
Attributes
3. Sends encrypted attributes
5. Receive Unique Identity
Identity Contract (IC)
4. Creates a Unique Identity
6. User request to access services using Unique Identity
Authentication Contract (AC)
7. Validates the user credentials
8. Sends user Credentials
10. Session key is sent
11. User sends a request to create a ticket
13. Send Ticket
Session Contract (SC)
9. Creates a Session Key
14. Request for Service
Service Provider
Ticket Contract (TC)
12. Creates a Ticket
EASEID

Figure 6 EASEID Model – User Interaction

4.6. EASEID-User interaction specification in Figure 6:

1. The user creates an account in the blockchain and gets a key pair as a result of account creation.

$$\text{BC}\rightarrow\text{U: } (PU_{user},PR_{user})$$

2. User decides all his attributes which can describe all his characteristics and encrypt them by using his public key.

$$E(PU_{user},[Attributes_{user}])$$

3. User sends his encrypted attributes and his public key to the EASEID Identity contract. This message is transmitted after encryption, using the public key of identity contract.

$$\text{U}\rightarrow\text{IC: } E(PU_{IC},[E(PU_{user},[Attributes_{user}])||PU_{user}])$$

4. Identity contract creates a unique Identity for the user using the data received.
5. The unique user identity is transmitted to the user

$$\text{IC}\rightarrow\text{U: } E(PU_{user},[ID_{user}])$$

6. User sends a request to the authentication contract to generate a session key.

$$E(PU_{AC}, [ID_{user} || PU_{user}])$$

7. Authentication contract validates the received credentials.
8. Authentication contract sends the user credentials to session contract.
9. Session contract on receiving the credentials generates a session key for the user.
10. Session contract sends the session key to the user.

$$\text{SC} \rightarrow \text{U: } E(PU_{user}, [K_{user,sc} || PU_{TC1}])$$

11. User when ready to access services, sends a request to generate a token to ticket contract.

$$\text{U} \rightarrow \text{TC: } E(PU_{TC1}, [K_{user,sc} || ID_{user} || PU_{user}])$$

12. Ticket contract creates a universal ticket for the user to access services from service providers.
13. Ticket contract sends all public access keys of service providers, ticket, and userID is encrypted using 1st public key of ticket contract along with user ticket.

$$\text{TC} \rightarrow \text{U: } E(PU_{user}, [Ticket || \cup_{i=1}^{n} Access_{PU_{sp_i}} || E(PR_{TC2}, [Ticket, ID_{user}])])$$

14. User on receiving, uses the ticket for accessing services from a specific service provider by carrying his ticket and the encrypted data received from the ticket contract.

$$\text{U} \rightarrow \text{SP: } E(Access_{PU_{sp_i} where\ i \in n}, [Ticket || ID_{user} || E(PR_{TC2}, [Ticket, ID_{user}])])$$

5. Results

The EASEID model is intended to ensure that users may access all services in a session by signing once. Cameron's identity rules[13] are alluded with regard to IDAMs[14][3], as well as the adoption of these IDAMs for single session sign-in Table 1.

Identity Principles and session-sign-in	IDAM			
	Sovrin	uPort	Shocard	EASEID
User Control and Consent.	Users may select which ID to use and which characteristics to reveal. Possibility of using a trusted web to protect users against deceit.	Users have complete control over the creation and publication of uPortIDs, and they may establish ownership. There is a risk of attribute leaking in the register.	Users have authority over the creation and release of ShoCardIDs. Only parties invited by the owner of the ShoCardIDs will be able to browse the attributes of the user, and the same will be confirmed by ShoCard servers.	The user will have total control over the attributes that will be used to create the ID, and only the user's consent will be required to divulge the attributes.
Minimal Disclosure for a Constrained Use.	Anonymized credentials built on zero-knowledge proofs ensure that the "least amount of identifying information" is disclosed.	When obtaining an uPort identification, there is no need to reveal any personal information.	ShoCardIDs are bootstrapped using the trusted identity document.	To create an ID, encrypted attributes are used, ensuring that the least amount of information is revealed.
Justifiable Parties.	The attributes are only accessible to authorized parties and agencies.	Everyone has access to the registry's characteristics. There is a chance that encrypted data will be leaked.	Only parties invited by the owner of the ShoCardIDs may access the characteristics, and ShoCard servers can obtain the attributes without being invited.	The attributes are accessible only to the authorized parties.
Directed Identity.	Allows for omnidirectional identifiers.	Allows for the unidirectional exchange of IDs between parties.	Supports the unidirectional exchange of IDs between parties.	Supports the unidirectional exchange of IDs between parties.

Pluralism of Operators and Technologies.	Creates a platform for intermediates between users and its entities in-network, as well as an interface for other identification systems.	Allows for type customization, however utilizing a specified data format is preferable.	Following interfaces with ShoCard centralized servers, parties can parse available trustworthy credentials.	Allows data to be as defined by the entity. The same ID can be used for accessing multiple services.
Human Integration.	Sovrin's usability and user comprehension of privacy are unclear.	Although a mobile application is available, its usability and user comprehension of privacy is unclear.	Although a mobile application is available, its usability and user comprehension of privacy is unclear.	Users are a part of the system since the properties must be given by the entity.
Consistent Experience Across Contexts.	It's difficult to say because it depends on whether Sovrin chooses various platforms or not.	Users interact with the mobile app, and a QR code scan is available.	Users interact with the mobile app, and a QR code scan is available.	Users can access any services on successful authentication
Single-session sign-in	Not defined	Not defined	Not defined	EASEID enables users to utilize a single sign-in to access services from service providers during a session.

Table 1 Identity principles and session sign-in relation to the IDAMs

IDAM's initial step is to generate Identity. Depending on the type of IDAM and its implementation, An identity must be formed for each individual service provider or just once, and when a user wants to access services from several service providers, they must submit credentials once or multiple times and get authenticated. In this contest, data is populated by taking into account a user accessing ten services. *Figure 7* depicts the time it takes to generate an identity in several IDAMs, whereas *Figure 8* depicts the time it takes to get authenticated when a user wishes to access services from 10 different service providers.

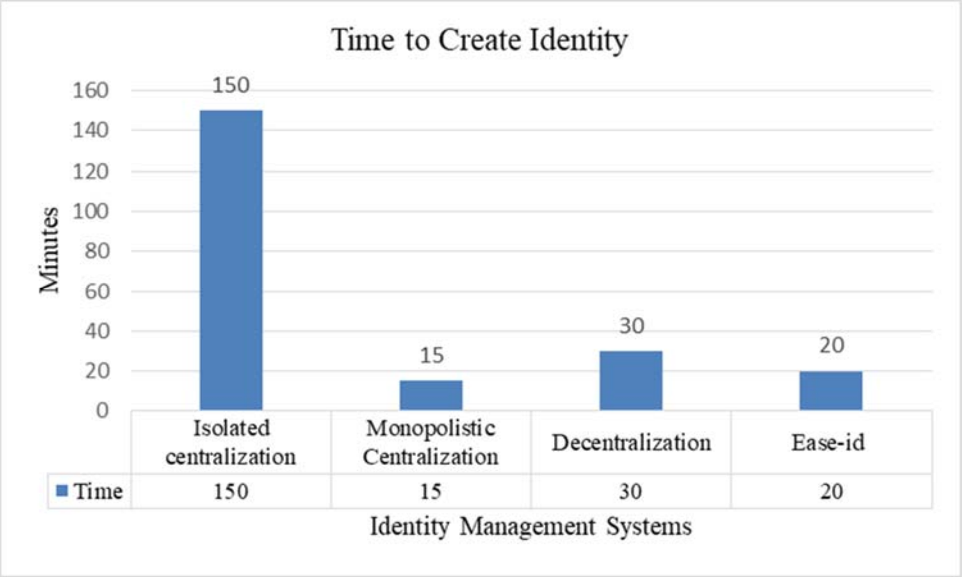

Figure 7 Time to create an identity (10 service providers)

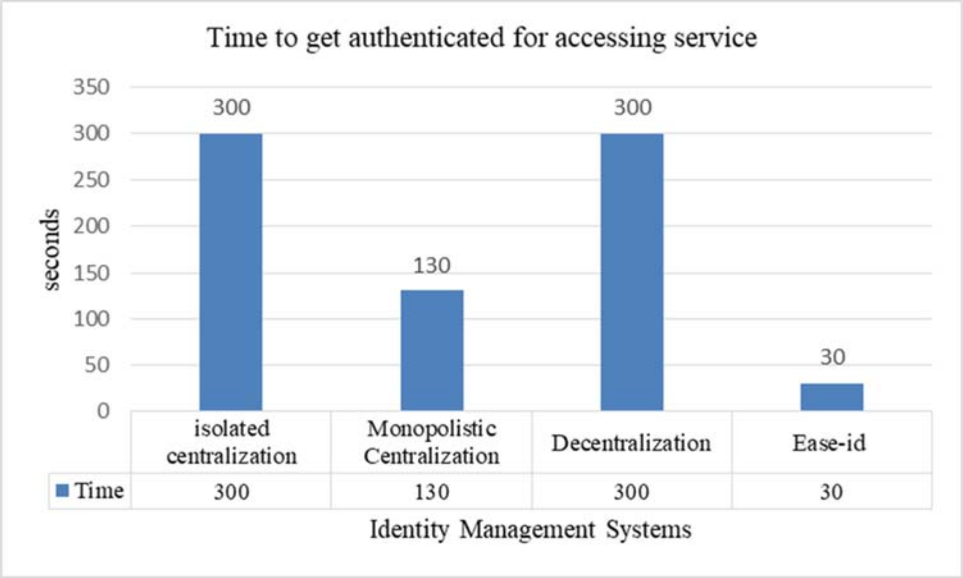

Figure 8 Time to get authenticated for accessing service

6. Conclusion

The paper introduces a model for session based self-sovereign identity and access management systems that can be built on the block chain. The EASEID model could overcome the problem of single point of failure while taking the leverage to use the advantages of centralized systems. It reduces the burden of repeated sign-in for the entire session and gives the ease of access to all the services provided by different service providers who are registered entities in the system. This could save users' time in two folds. The user need not register with every service provider and he need not get authenticated for accessing every single service accessed throughout the session. The burden of maintaining user's data can be relieved from the service providers which can help them to provide better services.

7. Conflicts of Interest

The authors declare no conflict of interest.

8. References

[1] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K. K. Raymond Choo, "Blockchain-based identity management systems: A review," *J. Netw. Comput. Appl.*, vol. 166, no. May, p. 102731, 2020, doi: 10.1016/j.jnca.2020.102731.

[2] J. Liu, A. Hodges, L. Clay, and J. Monarch, "An analysis of digital identity management systems - A two-mapping view," *2020 2nd Conf. Blockchain Res. Appl. Innov. Networks Serv. BRAINS 2020*, pp. 92–96, 2020, doi: 10.1109/BRAINS49436.2020.9223281.

[3] T. Rathee and P. Singh, "A systematic literature mapping on secure identity management using blockchain technology," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.03.005.

[4] S. Lohar, S. Babar, and P. Mahalle, "A proposed approach for Digital Identity management using Self Sovereign Identity," *Int. J. Next-Generation Comput.*, vol. 12, no. 2, pp. 158–168, 2021.

[5] R. Soltani, U. T. Nguyen, and A. An, "A Survey of Self-Sovereign Identity Ecosystem," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/8873429.

[6] K. Gilani, E. Bertin, J. Hatin, and N. Crespi, "A Survey on Blockchain-based Identity Management and Decentralized Privacy for Personal Data," *2020 2nd Conf. Blockchain Res. Appl. Innov. Networks Serv. BRAINS 2020*, pp. 97–101, 2020, doi: 10.1109/BRAINS49436.2020.9223312.

[7] X. Yang and W. Li, "A zero-knowledge-proof-based digital identity management scheme in blockchain," *Comput. Secur.*, vol. 99, 2020, doi: 10.1016/j.cose.2020.102050.

[8] Christopher Allen, "The Path to Self-Sovereign Identity," 2016. http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (accessed Feb. 07, 2022).

[9] A. Satybaldy, M. Nowostawski, and J. Ellingsen, "Self-sovereign identity systems: Evaluation framework," *IFIP Adv. Inf. Commun. Technol.*, vol. 576 LNCS, pp. 447–461, 2020, doi: 10.1007/978-3-030-42504-3_28.

[10] A. Tobin and D. Reed, "The Inevitable Rise of Self-Sovereign Identity," *White Pap.*, vol. 29, no. September 2016, p. 10, 2017, [Online]. Available: https://sovrin.org/library/.

[11] A. E. Panait, R. F. Olimid, and A. Stefanescu, "Identity management on blockchain – Privacy and security aspects," *Proc. Rom. Acad. Ser. A - Math. Phys. Tech. Sci. Inf. Sci.*, vol. 21, no. 1, pp. 45–52, 2020.

[12] R. Heck, J. Torstensson, Z. Mitton, and M. Sena, "UPORT : A PLATFORM FOR SELF-SOVEREIGN IDENTITY," 2017.

[13] K. Cameron, "The Laws of Identity," *Microsoft Corp*, 2005. http://www.ict-21.ch/ICT.SATW.CH/IMG/Kim_Cameron_Law_of_Identity.pdf (accessed Feb. 07, 2022).

[14] P. Dunphy and F. A. P. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE Secur. Priv.*, vol. 16, no. 4, pp. 20–29, 2018, doi: 10.1109/MSP.2018.3111247.

Authors Profile

Goluguri Satyanarayana Reddy
Working as Asssistant Professor in Department of Information Technology,MVGR College of enginnering, Vizianagaram, Andhra Pradesh,INDIA from 2007.

Educational Qualifications

B,Tech in Information Technology from MVGR College of engineering,
M.Tech in Information Technology from Andhra University.
Pursing PhD in Department of Information Technology from GITAM University.

Interseted Areas-Block chain Technologies,Distributed Systems,System Software, Operating Systems- UNIX.

Dr.Thammi Reddy Konala
Working as Professor in Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM University, Rushikonda, Visakhapatnam, 530045, AP, India.

Educational Qualifications

B.E from RV College of engineering,
M.Tech in Computer Science and Engineering from Andhra University.
PhD in Computer Science from JNTUH College of engineering, Hyderabad.

Interested Areas-Data Engineering, Quality Enhancement in Education, Data Mining, Text Mining, Cloud Computing, Distributed Systems