zkMe: The Web3 zk-Credential Network

zkMe Team

15 March 2023

v1.0

Abstract

As the Web3 ecosystem evolves and matures, the demand for secure, decentralized, and privacy-preserving identity solutions becomes increasingly critical. Conventional identity authentication methods, dependent on centralized authorities and intermediaries, are inefficient, vulnerable to various attack vectors, and frequently infringe upon users' privacy. In response to this challenge, we introduce zkMe, a privacy-centric credential network harnessing the capabilities of zero-knowledge proofs to facilitate secure and confidential credential issuance and verification. zkMe empowers users to selectively disclose their credentials to authorized entities without jeopardizing their privacy, providing enhanced control over their digital identities.

The protocol adheres to a privacy-by-design paradigm in which personal data is utilized solely as input for credential proof verifications and remains undisclosed to verifiers. Trust-minimization is ensured by delegating computations to a decentralized network of node operators. The protocol is transparent, open-source, and fully compliant with FATF Know-Your-Customer (KYC) and travel rule regulations. Moreover, zkMe exhibits high composability in accordance with W3C standards, enabling seamless integration with other identity silos and ensuring interoperability across both on- and off-chain ecosystems.

By delving into zkMe's most compelling use case—the application of zero-knowledge in KYC, or "zkKYC"—we offer a comprehensive architectural overview of zkMe, alongside an in-depth implementation process. KYC constitutes a vital procedure for financial institutions and various businesses to validate customer identities and adhere to regulatory mandates. Traditional KYC processes, however, tend to be laborious, time-consuming, and necessitate the sharing of sensitive personal data. Our decentralized, privacy-centric credential network employs zero-knowledge proofs to enable secure and confidential credential issuance and verification, granting users the ability to manage their credentials and selectively disclose them to authorized parties without exposing sensitive information. zkMe presents a more secure, efficient method for users to manage their credentials, diminishing their dependence on centralized authorities. The range of zkMe use cases in the Web3 ecosystem is extensive, encompassing applications for permissioned DeFi, credit loans, and DAO management, among others.

Our vision for zkMe is to establish it as the preeminent standard for decentralized, trustless, and privacy-preserving identity authentication infrastructure within the Web3 domain. By

returning control of personal data to users, zkMe seeks to facilitate a more secure, efficient digital landscape characterized by enhanced privacy and reduced reliance on centralized authorities. Our mission at zkMe is to deliver a cutting-edge infrastructure for presenting verified credentials in a trustless and private manner, ultimately empowering users to govern their own digital identities.

Table of Contents

1. Introduction

The advent of blockchain technology has brought about a paradigm shift in the way we think about digital identities and credentials. In traditional systems, personal information is stored on centralized databases and controlled by trusted third parties, which often leads to issues of privacy, security, and identity theft. The web3 ecosystem aims to address these concerns by enabling users to take control of their digital identities through decentralized systems.

To address these challenges, we present zkMe, a decentralized, privacy-focused credential network that utilizes zero-knowledge proofs (ZKPs) to enable secure and private credential issuance and verification. zkMe offers a flexible and extensible framework for managing a wide range of digital credentials, including identity, academic, professional, and financial credentials.

The protocol is built on privacy-by-design principles, with end-to-end zero-knowledge processing of personal data that is selectively disclosed as needed for verification. Users maintain full control over their data sharing, and all computations are handled by a decentralized network of node operators for trust-minimization. zkMe is a solution designed to fulfill standards put forth by various stakeholders, be it technical (W3C DID standards) or regulatory (FATF AML6 and EU MiCA among others).

In this paper introduction, we present an overview of zero-knowledge proof-based credential networks in *Section 1.1* and then our four key areas of innovation in *Section 1.2*. We describe the organization of the rest of this paper in *Section 1.3*.

1.1 State of research on ZKP-based credentials

The Web3 ecosystem necessitates a decentralized identity system that delivers privacy, security, and interoperability. The solution should empower users to exert complete control over their identity while enabling selective data sharing with third parties. Numerous research papers have proposed decentralized identity systems utilizing zero-knowledge proofs (ZKPs) to ensure privacy and security. ZKPs facilitate credential verification without disclosing extraneous information beyond what is required.

Yang and Li (2020) introduce a novel approach to digital identity management in blockchain networks, designed to bolster privacy and security by allowing users to authenticate their

identity without exposing personal data. ZKPs offer an effective means of preserving user privacy and security while maintaining the transparency and integrity of the blockchain system.

Microsoft (2020) presents an extensive overview of zero-knowledge proof (ZKP) technology and its potential to enhance privacy in digital identity management. The article elucidates how ZKPs allow users to authenticate their identity without revealing personally identifiable information through cryptographic protocols that enable authentication without disclosing sensitive data. The advantages of ZKPs, such as improved user privacy, diminished dependence on centralized authorities, and increased control over personal data, are explored, along with potential applications in online authentication, digital signatures, and decentralized identity systems.

Schanzenbach et al. (2019) propose ZKlaims, a privacy-preserving attribute-based credential scheme that employs non-interactive zero-knowledge proofs, enabling users to demonstrate possession of specific attributes without divulging any additional information about those attributes or their identity.

*Pieter (2021) and Pauwels (2022) p*ut forth the notion of zkKYC, an innovative solution addressing the long-standing challenge of Know Your Customer (KYC) compliance. By leveraging self-sovereign identity (SSI) and zero-knowledge proofs (ZKP), zkKYC enhances privacsy and security while alleviating the compliance burden on businesses. The zkKYC process does not mandate businesses to possess knowledge of the customer's identity; rather, it relies on a trusted third party to verify the customer's identity using SSI and ZKP, granting customers control over their identity information.

Luong and Park (2023) present a meticulously crafted and groundbreaking approach to privacy-preserving identity management on the blockchain. The employment of zk-SNARKs represents a substantial contribution to blockchain-based identity management, addressing privacy and security challenges while maintaining high levels of functionality and usability.

ZKP-based solutions, as proposed in various research papers, tackle these challenges by enabling credential verification without revealing superfluous information. These solutions grant users full control over their identity while also permitting selective data sharing with third parties. ZKP-based identity solutions hold the potential to unlock the full capacity of the Web3 ecosystem by providing a secure, privacy-preserving method for managing identities.

1.2 Four key design goals

The following four key design goals are what determine the quality and success of a ZKP-based credential protocol and are the values by which the solution put forth in this paper was designed to fulfill.

1.2.1 Privacy-by-Design

- **End-to-End Zero-Knowledge Architecture**

Personal data is processed entirely automatically, directly on the end user's device or within a decentralized oracle network. Throughout the due diligence process, no third party, including regulators, companies, or data processors, including the protocol/infrastructure provider, can access any personally identifying information (PII). Personal data is not shared and is not stored on centralized servers. End-to-end zero-knowledge architecture guarantees the highest level of data security and privacy.

- **Selective Disclosure**

Selective disclosure of personal information in a secure manner is possible. Personal information is strictly anonymized in the form of yes/no responses to preselected demographic questions (e.g. through the use of zero-knowledge algorithms) and even once anonymized, only shared when and to whom strictly necessary. For example, an identity management solution discloses and verify that a user is over 18 years old to a verifier, rather than disclosing the user's actual birthdate. The demographic questions verified by the identity management solution are carefully designed to prevent any indirect inference of a single user's identity; ensures that each possible answer combination is expected to be shared by a population of at least 50.000 people, thus making identification of a single user virtually impossible. Providing the highest degree of anonymity while allowing access to the required information for legitimate purposes.

- **Self-Sovereign Identity**

Self-Sovereign Identity (SSI) frameworks are used in which the credential holder has full control over data sharing of all it's data, including it's anonymized individual information. The credential holder can easily ammend, update, and revoke verification permissions for single verifiers from their end user device, ensuring complete control over their personal data. The solution eliminates the need for cumbersome email processes to delete data or for

sharing data without explicit consent. The SSI framework ensures that the user retains full control over their personal data, and enhances the privacy and security of our data processing system.

1.2.2 Decentralization

- **Multi-Party Computation Identity Oracle**

The decentralized identity solution employs a Multi-Party Computation (MPC) approach for all trust-building determinations and computations. This includes verification of user credentials, generation of zero-knowledge proofs, and encryption and decryption of users' data. The system relies on a network of node operators, eliminating the need for central computation and preventing any potential protocol manipulation. The approach ensures that trust is built upon a foundation of decentralization, security, and cryptographic techniques, with no reliance on proxy verifications.

- **Trust Minimization**

The solution establishes a highly trustworthy layer for decentralized identity systems through decentralization, strong anchoring in high-security blockchains, cryptographic techniques, and cryptoeconomic guarantees. By minimizing the need for trust, our system enhances the security and privacy of user data.

- **Party-Agnostic Design**

The solution employs a party-agnostic design, meaning that no single entity, including the zkMe network, controls any role within the infrastructure. The roles of "Issuer", "Holder", "Verifier", "Node Operator", and even "Regulator" are context-specific, and changes or removals can be made with the consent of the governing DAO. The design ensures a high degree of flexibility and adaptability, enhancing the system's resilience and security.

1.2.3 Compliance

- **Regulatory Requirements**

The solution satisfies the customer due diligence requirements of the Financial Action Task Force (FATF) recommendations (incl. crypto travel rule requirements), the EU's 6th Anti-Money Laundering (AML) directive, and forthcoming EU MiCA and US Lummis-Gilibrand

bill directives, among others. The solution employs protocols to certify that due diligence checks have been processed. With threshold cryptography, the user's actual identity remains unknown until a regulator initiates "bad actor" proceedings, prompting the Regulator, Verifier, and Issuer to come together decrypt the Holder provided identity documentation using their combined. This design ensures verifiable anonymity until proven guilty.

- **Technological Standards**

The solution is built in compliance with the World Wide Web Consortium's (W3C) Decentralized Identifiers (DIDs), Verifiable Credential (VC), and Verifiable Presentation (VP) standards. This ensures interoperability with other decentralized identity systems that also comply with these standards.

1.2.4 Transparency

- **Open Source and Composable**

The algorithms required to run the solution infrastructure, including user credential verification and zero-knowledge proof generation, are or will be open-sourced, regularly audited, and available for the ecosystem to expand and build additional use cases. This design promotes transparency, collaboration, and innovation within the identity ecosystem.

- **Cross-Silo & Multi-Chain Identity**

The solution can process and cross-pollinate credentials across all identity silos, allowing web3 identities, such as those in Metamask or Trust wallets, to anonymously benefit from credentials in real-life or web2 identities, such as FICO credit scores or social media followings. This design promotes greater convenience and flexibility for users seeking to leverage their credentials across multiple domains.

1.3 Organization of this paper

The remainder of this paper is structured as follows:

- Chapter 2 offers an exhaustive background review of current identity management solutions, their limitations, and recent technological breakthroughs with the potential to transform the field.

- Chapter 3 illustrates the drawbacks of existing identity management solutions through a case study of KYC procedures in the financial sector.

- Chapter 4 presents the zkMe protocol as a Pareto-optimal identity solution, which offers enhanced privacy while imposing lower entry barriers compared to current alternatives. This chapter also delves into zkMe's provable security.
- Chapter 5 furnishes in-depth technical implementation information concerning the primitives utilized by zkMe and the proof of the protocol's security.
- Chapter 6 applies zkMe to KYC processes, concentrating on KYC procedures for Web3 decentralized applications.
- Chapter 7 showcases a range of zkMe use cases, emphasizing its application in Permissioned Decentralized Finance (DeFi).
- Chapter 8 pinpoints areas for future research and enhancement of zkMe.
- Chapter 9 concludes the whitepaper by summarizing the principal findings and contributions of this study.

2. Background

This chapter offers a comprehensive overview of the current identity authentication market, delving deeper and singling out identification services for financial products for Know-Your-Customer (KYC) / Anti Money Laundering (AML) purposuses through a case study. Subsequently, the following sections delve into the limitations and challenges of these existing solutions. We examine the recent advent of decentralized identity (DID) solutions, highlighting their unaddressed issues and challenges. We review recent research on Self-Sovereign Identities (SSI), emphasizing its benefits over Federated Identities (FI). Finally, we discuss the potential of SSI with Zero-Knowledge Proofs (ZKPs) in identity and credential management.

Overall, this paper aims to provide a comprehensive solution for handling identities with the highest degree of Holder privacy without imposing prohibitive hurdles for Verifiers, allowing for effective strategies for managing and protecting digital identities in a trust-less and decentralized world.

2.1 Identity & Credentials

Identities are complex and multifaceted constructs that are shaped by a variety of factors, including social, cultural, and personal experiences. They are not fixed, but are constantly evolving and negotiated through interactions with others and with society as a whole. Identities are arrived at by the way in which the person faces and uses his experiences. Credentials are used to proof experiences. We own credentials for our demographic details, our education, and our social or financial status. We benefit from these credentials. We use them as trust-building tokens for improved access and leverage in a very wide variety of complex services we consume.

Economically speaking, credentials are valuable in two dimensions. They are more valuable the more they protect the privacy (incl. anonymity, convenience, and purpose-driven disclosure) of the credential holder's information and in transparency (incl. efficiency, atomicity, and compliance) in proving eligibility to the credential verifier. Traditionally, these goals of privacy and transparency were in conflict with each other. Holders (i.e. the entity wanting to prove a credential) either have to disclose much more than they needed to Verifiers (i.e. the entity wanting to check a credential), or the Verifiers have to lower eligibility criteria below acceptable quality standards to keep onboarding hurdles low.

Identities and identity management have become critical issues in today's digital age, where the use of online services and digital platforms has become a fundamental part of our daily lives. As we increasingly rely on digital technologies for communication, commerce, and social interaction, we also need to manage and protect our digital identities. Identity management refers to the process of creating, maintaining, and controlling the identity (and the credentials it is composed of) in the digital world. This process involves identification, authentication, authorization, and access control, and is crucial for ensuring that only authorized individuals or entities access restricted services and/or information.

2.2 The identity management process

The first step in the identity management process is **identification**, which involves collecting and verifying information about an individual or entity to establish their identity. This information can include name, date of birth, address, phone number, and other personal information. Identification is often done through the use of credentials, such as usernames and passwords, or through biometric data, such as fingerprints or facial recognition. In some cases, multiple forms of identification may be required to establish an individual's identity.

Once an individual's identity has been established, the next step is **authentication**. Authentication is the process of verifying that an individual is who they claim to be. This is typically done by requiring the individual to provide credentials, such as a password or biometric data, that match the information collected during the identification process. Authentication can be done in several ways, including single-factor authentication, where only one credential is required, or multi-factor authentication, where multiple credentials are required. Multi-factor authentication is typically more secure, as it provides an additional layer of protection against unauthorized access.

After an individual's identity has been authenticated, the next step is **authorization**. Authorization is the process of granting or denying access to specific resources or services based on an individual's identity and permissions. Authorization is typically based on a set of rules or policies that determine what actions an individual is authorized to perform. These rules can be based on a variety of factors, including an individual's job role, level of clearance, or other criteria.

The final step in the identity management process is **access control**. Access control is the process of controlling and monitoring access to resources or services based on an individual's identity and permissions. Access control is typically done through the use of centralized access control lists (ACLs), which define who is allowed to access specific resources and what actions they are authorized to perform. Access control can also be enhanced through the use of other security measures, such as encryption, firewalls, and intrusion detection systems.

Most solutions use a Federated Identity (FI) system is a single digital identity that is shared across multiple systems and applications. This identity is usually established and maintained by a central identity provider (IdP) that manages authentication and authorization on behalf of multiple Verifiers. Federated Identities only work based on the idea of trust in one central all-controlling entity. They are complex, non-interoperable and expose a single point of trust and failure.

2.3 The rise of decentralized identities (DID)

DIDs are unique identifiers that are not tied to any centralized authority or identity provider. They are designed to be globally unique and resolvable, meaning that they can be used to reference a specific entity or object, such as a person, organization, or device.

DIDs enable individuals to own and control their own identities and personal data, by providing a way to create and manage decentralized digital identities. This is achieved by allowing individuals to create their own DIDs, which can be associated with their personal data and credential, and stored securely on a decentralized network, such as a blockchain or other distributed ledger technology.

One of the key benefits of DIDs is that they provide a way to establish trust between different parties in a decentralized network. This is achieved through the use of cryptographic proofs, which enable parties to verify the authenticity and integrity of the data associated with a particular DID.

"*Decentralized Identifiers (DIDs) v1.0*" *(2020)* is a W3C specification that defines the technical requirements for DIDs and how they can be used in decentralized identity solutions. A Decentralized Identifier (DID) is a text string consisting of three components, namely the DID URI scheme identifier, the DID method identifier, and the DID method-specific identifier. Through the use of a DID method-specific resolver, a DID can be resolved, and its corresponding DID document can be retrieved from a Verifiable Data Registry. The DID document expresses cryptographic material, verification methods, and services, which enable a DID controller to prove control of the DID and facilitate trusted interactions associated with the DID subject.

The use of DIDs allows entities to maintain separation between their identities, personas, and interactions by enabling them to have as many DIDs as necessary. These identifiers can be appropriately scoped to different contexts, enhancing privacy as a single identifier no longer needs to correlate all interactions. An entity may possess one or multiple public DIDs, which are stored on a public Verifiable Data Registry, and/or private DIDs, which resolve to a DID document stored on a private registry.

Dib and Toumi (2020) provides a comprehensive overview of Decentralized Identity Systems, its architecture, challenges, solutions, and future directions. Decentralized Identity Systems has the potential to provide users with control over their digital identities, and to provide a platform for secure and private interactions between users. The paper also suggests the use of standardized protocols such as W3C's DID and Verifiable Credentials.

In summary, this paper demonstrate the benefits of DID over FI. Firstly, DID eliminate the need for a central authority to manage identity verification and authorization, which means that users have more control over their personal data. This reduces the risk of data breaches

and identity theft. Secondly, DID provides greater flexibility and interoperability than federated identities, allowing users to easily access resources across different systems and applications without relying on a single identity provider. Finally, DID has the potential to enable new business models and use cases that are not possible with federated identities, such as secure peer-to-peer transactions and privacy-preserving authentication. In summary, the use of DIDs provides a powerful means of enabling decentralized, trust-enhancing interactions between entities by allowing them to maintain control over their identity and associated data.

2.4 Issues with DID solutions: the quality gap

As above, decentralized identity and credential management solutions have gained popularity in recent years due to their potential to provide greater privacy, security, and control over personal data compared to traditional centralized systems. However, some issues with existing decentralized identity solutions remain unaddressed. Here are some of the key challenges:

One of these issues is **identity cloning**, where attackers can easily create fake identities that are identical to real ones, posing a serious security threat.

Another issue is the **bundling problem**, where multiple identities are linked together, making it difficult to manage them separately.

Scalability is another significant challenge for many decentralized identity solutions, requiring significant computational resources to operate.

Compatibility with legacy systems is also essential for decentralized identity solutions to be widely adopted. Achieving identity interoperability becomes crucial to ensure that identities can be seamlessly shared and used across different platforms.

Sybil attacks are a significant challenge for many decentralized identity solutions, allowing attackers to create multiple fake identities to carry out malicious activities.

Ensuring **accountability** is critical for decentralized identity solutions, allowing users to hold other parties accountable for their actions. However, many solutions lack the ability to enforce accountability.

There are several rules that must be followed when issuing and verifying credentials in a decentralized identity system to ensure that credentials issued and verified in a decentralized identity system are secure, trustworthy, and privacy-preserving:

- **Identity Verification:** Before issuing a credential, the issuer must verify the identity of the individual to whom the credential will be issued. This can be done through a variety of methods, such as in-person verification, document verification, or digital identity verification.

- **Credential Issuance:** Once the issuer has verified the individual's identity, they can issue a verifiable credential that includes the necessary claims and metadata. The credential must conform to the Verifiable Credential Data Model standard and any custom extensions specified by the network.

- **Cryptographic Proofs:** The credential must include a cryptographic proof, such as a digital signature or zero-knowledge proof, that allows the recipient of the credential to verify its authenticity and integrity.

- **Revocation:** The issuer must have the ability to revoke a credential if it is no longer valid or if the individual to whom the credential was issued no longer has the right to use it. Revocation must be done in a way that does not compromise the privacy or security of the individual.

- **Verification:** When verifying a credential, the recipient must use the cryptographic proof included with the credential to ensure its authenticity and integrity. The recipient must also verify that the credential was issued by a trusted party and that it has not been revoked.

In conclusion, addressing the issues of identity cloning, bundling, scalability, interoperability, sybil-resistance, and accountability will be critical for the widespread adoption of decentralized identity and credential management solutions in the web3 ecosystem. The proposed solutions discussed in literature are steps towards addressing these issues and provide insights for future research in this field.

2.5 Self-Sovereign Identities (SSI) to empower Holders

A Self-Sovereign Identity (SSI) is a specialized DID model that enables individuals to own, control, and share their personal data and credentials in a secure and private manner. SSI

has the potential to transform traditional identity and credential management by providing a user-centric, secure, and privacy-preserving solution.

Verifiable Credentials (VC) are a key feature of SSI, which are cryptographically secure digital representations of a person's or organization's information issued by trusted entities. These credentials can be stored and managed in a decentralized manner, using blockchain or other distributed ledger technologies, which ensures their security and tamper-resistance.

One of the key benefits of SSI for identity and credential management is that it eliminates the need for intermediaries, which can be expensive and time-consuming. With SSI, individuals can directly present their VC to the entities that require them, without the need for intermediaries. This also reduces the risk of identity theft and fraud, as the individual is in control of their own data.

Another benefit of SSI is that it enables greater privacy and security for personal data and credentials. With SSI, personal data and credentials are encrypted and stored locally on the individual's device, rather than in a centralized database that can be vulnerable to hacking and data breaches. This gives individuals greater control over who has access to their personal information, and reduces the risk of identity theft and fraud.

Several papers have explored the potential applications of SSI in identity and credential management.

Mukta et al. (2021) presents a novel approach to trust management in SSI systems based on credential-based trust evaluation. The proposed system has the potential to provide a more flexible and adaptable approach to trust management in SSI systems, although further research and testing are needed to evaluate its feasibility and practicality. Siddiqui et al. (2021) presents a novel approach to providing SSI as a cloud service using TEEs, which addresses the scalability and security limitations of current SSI systems. The proposed CaaS system has the potential to provide users with a more user-friendly and accessible identity management system that can be accessed from any device. However, the feasibility and practicality of the proposed system needs to be further evaluated and tested in real-world scenarios. Schardong and Custódio (2022) presents a mapping and taxonomy of SSI systems, categorizing them based on the type of blockchain technology they use, the level of user control they provide, and the level of privacy they offer. The authors also provide a detailed analysis of each category, discussing the advantages and limitations of each type of SSI system. The review highlights the key research themes in the field, including the technical

challenges of SSI systems, the regulatory and legal considerations, and the potential use cases for SSI systems in various industries.

In the SSI model, there are four key roles and one central tool:

- **Issuer:** The Issuer creates and issues digital credentials, such as driver's licenses or passports, to the Holder. Issuers are trusted by the sheer fact that they are typically trusted organizations, such as government or educational institutions with a reputation for accuracy and reliability in their domain. They issue VCs to Holders and write reference proofs to verifiable data registries.

- **Holder:** The Holder stores and controls access to their own digital credentials in form of VCs in self-custody. They present credentials to Verifiers.

- **Verifier:** The Verifier verifies the authenticity of the VCs presented by the Holder and controls access to restricted services and/or information based on the verification of said credentials. They comply with law enforcement and can access verifiable data registries to verify the authenticity of the credentials presented by the Holder.

- **Regulator:** The Regulator ensures that the rules and regulations regarding the issuance, storage and use of digital credentials are followed by the Issuer, Verifier, and Holder by interacting with the verifiable data registry and requesting information from the Verifier.

- **Verifiable Data Registry:** A database (either centralized or decentralized) used to store, manage and share data. By the fact that only trusted Issuers are enabled the data registry they are the main trust building mechanism that allows the Verifier to trust the Holder VC without contacting the Issuer directly.

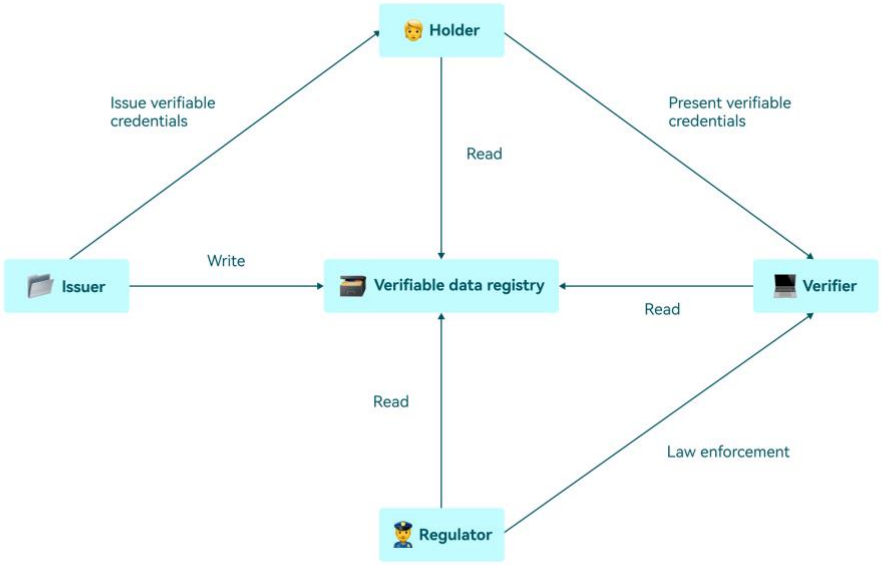

***Figure 1*. SSI role overview**

In summary, SSI offers a decentralized, secure, and privacy-preserving solution for managing personal data and credentials. By giving individuals control over their own data, SSI enables greater efficiency, security, and trust in digital transactions.

2.6 Soulbound Token (SBT) to anchor Identities on-chain

The Soulbound Token (SBT) is a novel concept developed in *May 2022* by a team consisting of Ethereum co-founder *Vitalik Buterin*, lawyer *Puja Ohlhaver*, and economist and social technologist *E. Glen Weyl*. SBTs are non-transferable tokens that serve as identity and credentialing tools, representing a person or entity on the blockchain. SBTs are created by wallets or blockchain accounts called "Souls," and users can tokenize their achievements or traits using them. Multiple Souls can be held by users, each containing different credentials for various aspects of their lives. For instance, a person may possess an SBT to track professional credentials for job interviews, an SBT for tracking health records, or an SBT to track their gaming achievements.

SBTs are exceptional in that each token has a unique identifier and metadata, such as education, ownership, credit scores, criminal records, affiliation, and more. Similar to achievements, a person can have an unlimited number of SBTs. SBTs also possess verifiability, allowing individuals to verify their information using blockchain technology,

including authentication of eligibility and ownership, membership activities, education discounts, and other KYC scenarios.

The most significant characteristic of SBTs is their non-transferability, as they can only be granted to users by wallet authorities. Unlike other NFTs, they can not be sold or transferred, making impersonation very difficult, which is a good way to prevent theft and fraud, much like in DeFi. SBTs are expected to become a valuable tool for future use in digital identity verification.

3. A case study on identity management in KYC

Identity Management in Know Your Customer (KYC) procedures are used to verify the identity of customers or clients for regulatory compliance purposes. KYC procedures are critical for preventing money laundering, fraud, and other financial crimes, and identity management plays a key role in ensuring the accuracy and reliability of KYC data. Identity management technologies, such as biometrics, blockchain, and artificial intelligence, are increasingly being used to enhance KYC processes and improve customer experience.

3.1 Regulatory framework

Know Your Customer (KYC) regulations are a critical component of anti-money laundering (AML) and counter-terrorism financing (CTF) efforts globally. The regulatory framework for KYC varies depending on the jurisdiction, with different countries and regions having their own laws and guidelines. For example, in the United States, KYC requirements are set forth by the Bank Secrecy Act (BSA) and its implementing regulations. Similarly, the European Union has implemented KYC regulations through the Fourth Anti-Money Laundering Directive (AMLD6), which requires financial institutions to identify their customers, assess the risks of money laundering and terrorist financing, and establish risk-based procedures for ongoing due diligence. The regulatory framework for KYC is constantly evolving, with new laws and guidelines being introduced to combat emerging threats such as cybercrime and the financing of terrorism. Compliance with these regulations is critical for financial institutions and other regulated entities to avoid legal and reputational risks.

Regulations for KYC in web3 are still a developing. Selected countries have started to implement regulations specific to web3 technologies, while others have issued guidance or are in the process of developing regulations. Most notably, the European Commission has

passed a number of EU Directives and Regulations, including **MiCA, TRF and AMLD7**. These regulations will require all Virtual Asset Service Providers (VASPs) to undergo customer due diligence and comply with FATF requirements (incl. travel rules) just like any other financial service provider. This applies to all for-profit projects, including DeFi, NFT, DecSoc, GameFi, Wallets, Advisory, or Consulting. The legislation also requires protection of users' personal data according to GDPR, putting the industry in a

Similarly, in the United States, regulators have taken an interest in web3 technologies, and there have been several proposed bills and guidance issued by the SEC and CFTC. For example, the SEC has issued guidance on whether certain digital assets are securities and therefore subject to securities laws. Additionally, the Digital Asset Securities Bill, currently in draft form, seeks to provide regulatory clarity and protect investors by requiring digital asset platforms to register with the SEC. Other countries, such as Switzerland, the United Kingdom, Hong Kong, Singapore, and Japan, have or are about to implement regulations specific to web3.

3.2 The surge of electronic KYC (eKYC)

KYC procedures have been in place in the financial industry for decades, but the increasing use of digital technologies has made identity management a more complex and challenging process. In the past, KYC procedures relied primarily on paper-based documentation, such as passports and driver's licenses, to verify the identity of customers. Verifiers collect, verify, and assess user information to ensure that their identity is genuine and that they pose an acceptable level of risk to the organization within a trade-off of information transparency and user privacy. A higher degree of transparency makes risk profiling easier but imposes user entry hurdles in the form of excessive data sharing requirements.

However, the rise of online banking and digital transactions has made it easier for criminals to impersonate others and commit financial crimes. As a result, regulators have increased their scrutiny of KYC procedures, and financial institutions have had to adopt more robust identity management technologies to comply with regulations and protect against fraud, giving rise to an extensive market of electronic credential verification solutions or eKYC.

The current end-to-end process for eKYC involves a linear and sequential transfer of personal information from one entity to the next. Unfortunately, this often results in an uncontrolled and unproportionate sharing of more personal information than required to

more unrelated third-parties than required. *Figure 2* provides a simplified overview of the different stakeholder roles of a typical eKYC process and how information is typically shared. The **Issuer** issues credential documentation (in physical or digital form), which are held by the **Holder**.

When onboarding at a **Verifier**, the Holder presents these identity documents. To ensure the authenticity and integrity of the presented information, the Verifier either verifies it themselves or (unbeknownst to the Holder) relies on third-party eKYC providers (which in turn rely on additional third-parties) to process verification. In such a process, it is thus common that at least two parties that are not part of the original service relationship between Holder and Verifier have unrestricted access to the Holder's private information.

Once successfully verified, the Holder is onboarded as a customer to the Verifier services. For certain industries and use cases, a **Regulator** may define process requirements and may impose future data sharing obligations for both service usage and personal data collected on the Holder following the verification, especially once the Holder is suspected to be involved in illegal activities such as money laundering.

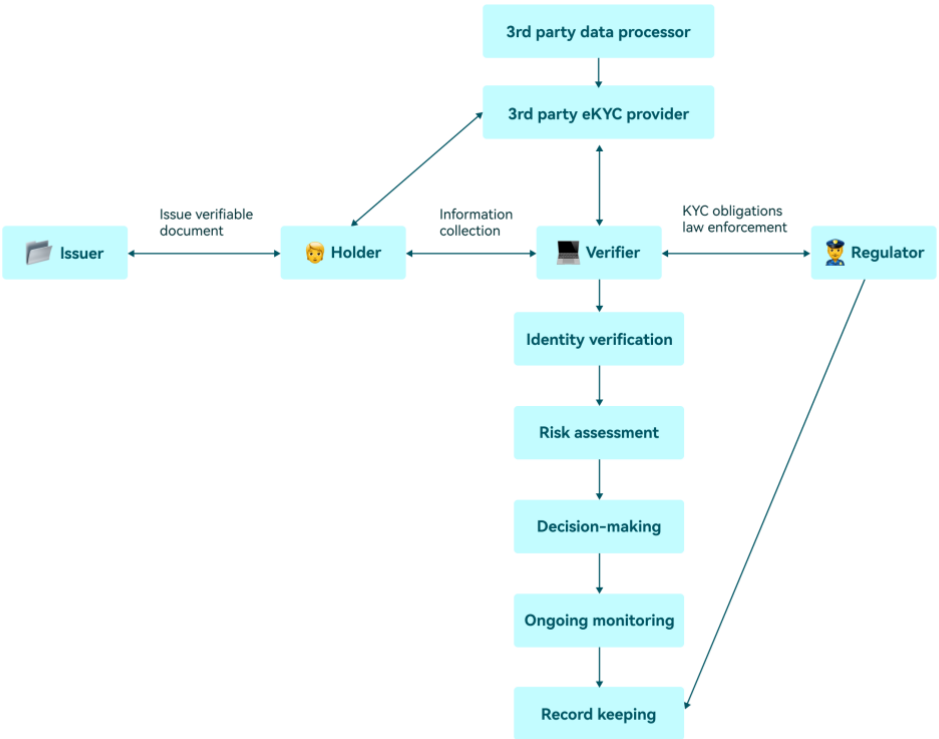

***Figure 2*. Typical eKYC information flow**

Figure 3 is a sequence diagram that depicts the typical eKYC process, which includes several steps. The first step involves the collection of personal information, such as name, date of birth, address, and government-issued ID (passport, driving license, *et.al.*). After this, the holder's identity is verified by checking their ID against public databases or conducting a facial recognition check using biometric technology.

The goal of this step is to ensure that the customer's identity is genuine and matches the information provided. Following identity verification, the level of risk associated with the customer is assessed by analyzing factors such as their financial history, employment status, and credit score. Based on this assessment, the Verifier decides whether to approve or reject the holder's application, sometimes requiring additional documentation or further verification.

The Verifier must then keep records of the KYC process and any relevant documentation for compliance purposes, storing the holder's personal information securely and ensuring that it is not shared with unauthorized parties. Finally, ongoing monitoring of the holder's activities is necessary to ensure compliance with relevant regulations and to detect any suspicious activity that may indicate fraudulent behavior.

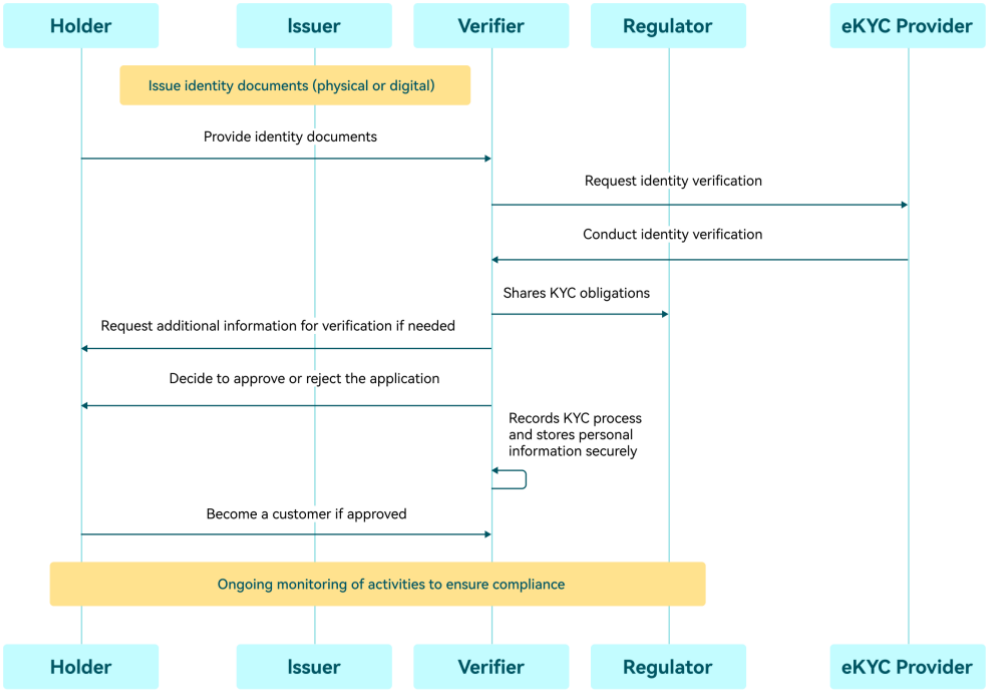

***Figure 3*. Typical eKYC sequence diagram**

3.3 Issues with eKYC: The trust gap

Identity and credential management solutions are essential in securing and managing users' identities, credentials, and access control to various systems, applications, and services. The current state of these solutions is rapidly evolving, driven by advances in technology and increasing demand for privacy, secure and efficient verification processes. However, these solutions are facing various issues, including data security and privacy, lack of user control, vulnerabilities to cyber attacks, regulatory compliance, interoperability, and structured transparency.

Studies have explored the different approaches to improve eKYC process **efficiency and effectiveness**. A significant challenge in the KYC process is the need to balance compliance requirements with customer experience.

Another significant challenge in the eKYC process is that it does **not ensure the security and privacy** of customers' personal information. Many studies have proposed different approaches to enhance the security and privacy of the KYC process.

Data security and privacy are critical concerns for eKYC solutions. These solutions store sensitive information, such as personal identification information, login credentials, and access control policies. Therefore, any breach in the security of these solutions can lead to severe consequences, such as identity theft, data loss, and financial fraud.

eKYC solutions often **lack user control**, which can lead to frustration and mistrust among users. Users have little control over the collection, storage, and use of their personal information, which can violate their privacy rights.

eKYC solutions are **vulnerable to cyber attacks**, which can lead to severe consequences, such as data breaches and identity theft.

eKYC solutions frequently **do not comply with various regulatory requirements**, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA) by sharing personal data to unrelated third-party providers.

Lack of interoperability is widespread in eKYC, which can lead to inefficiencies and increased costs. Different systems and applications use different identity management solutions, which can create silos and hinder collaboration.

eKYC solutions often lack **structured transparency**, which can lead to a lack of trust among users. Users have little visibility into the collection, storage, and use of their personal information, which can violate their privacy rights.

While some jurisdictions, such as the European Union with its *General Data Protection Regulation (GDPR)*, have taken steps to address the privacy implications of mandated and voluntary data sharing, much work remains to be done. Some institutions and regulators are experimenting with *Privacy Enhancing Technologies (PET)* to avoid accessing personal information for transparency purposes, but these initiatives are limited in scale and maturity. The *Australian Transaction Reports and Analysis Centre (AUSTRAC)* is working on a privacy-preserving encryption system that could allow them to examine patterns in financial transaction data without accessing the underlying information. This project attempts to tackle the recursive oversight problem to achieve structured transparency.

3.4 A KYC solution based on SSI, SBT and ZKP - zkKYC

A novel KYC solution based on an SSI model that employs SBT and ZKP technologies, would enable individuals to manage their own identities securely and privately, while also allowing institutions to verify their identities effortlessly. Given that such a solution would be private-by-design, it would allow for a true zero-knowledge know-your-customer (zkKYC) process and address the quality gap of existing DID solutions and the trust gap of existing eKYC solutions.

The solution would involve individuals creating a digital identity that is stored on a blockchain. The identity would contain only verified, ZKP-anonymized information. When an individual needs to prove their eligibility to a service provider, they would only share proof of fulfilling a certain eligibility criteria (i.e. a VP of the relevant VC) rather than sharing detailed credentials. This would reduce the risk of their information being misused or stolen. To ensure privacy and security, the individual would control access to their identity, and institutions would need to request access and be approved by the individual before being able to view any information. This would be accomplished using a combination of digital signatures and encryption.

Institutions would also have their own identities, and individuals could verify the authenticity of these institutions before sharing any information. This would be achieved using a web of trust, where trusted entities vouch for the authenticity of other entities.

Overall, the zkKYC model, which includes a regulator role, provides a more efficient and secure solution for KYC processes, which enables individuals to maintain control over their digital identities while providing institutions with a more trustworthy and streamlined KYC process.

4. zkMe - The web3 credential network

In this chapter, we introduce zkMe, our innovative implementation of a Credential Network protocol that enhances the SSI model. By integrating VCs, VPs, DIDs), and SBTs, zkMe offers a comprehensive solution. Its primary design objectives emphasize privacy-by-design, decentralization, regulatory compliance, and transparency.

4.1 zkMe's design philosophy

One of the notable features of zkMe is its **privacy-by-design** approach, which ensures that personal data is processed fully automatically and directly on the end device or in a decentralized oracle network. This means that no party has access to personally identifying information, and personal information is only shared when strictly necessary. Users are in full control of their data, including the ability to revoke verification permissions on a project-by-project basis from their mobile phone.

Another important aspect of zkMe is its **decentralization**, which ensures that all trust-building determinations and computations are handled by a decentralized network of node operators. This approach minimizes the risk of protocol manipulation and ensures that there is no reliance on proxy verifications. zkMe is also party-agnostic, meaning that no role in the infrastructure is fixed and controlled by a single entity.

zkMe is designed to **comply with regulations**, including KYC/AML FATF recommendations (incl. travel rule requirements), EU 6AML and TRF directives, and even upcoming EU MiCA and US Lummis-Gilibrand bill requirements. The platform is built with compliance to W3C DIDs, VC, and VP standards in mind.

Finally, zkMe is **transparent and open source**, which means that all algorithms required to run the infrastructure are audited regularly and provided to the ecosystem to expand and build additional credential use cases on. The platform is also able to process and cross-polinate credentials across all identity silos, enabling users to benefit from credentials in their web3 identities as well as their real-life or Web2 identities.

Overall, zkMe is a comprehensive solution to address the issues of privacy, decentralization, compliance, and transparency in credential networks. Its emphasis on end-to-end zero-knowledge processing, selective disclosure, and SSI provides users with a high degree of control over their personal data. The platform's compliance with existing and upcoming regulations ensures that it can be used in all industries, ranging finance, to gaming, travel, or even social media. *Figure 4* shows the sequence diagram for an identification process using zkMe.

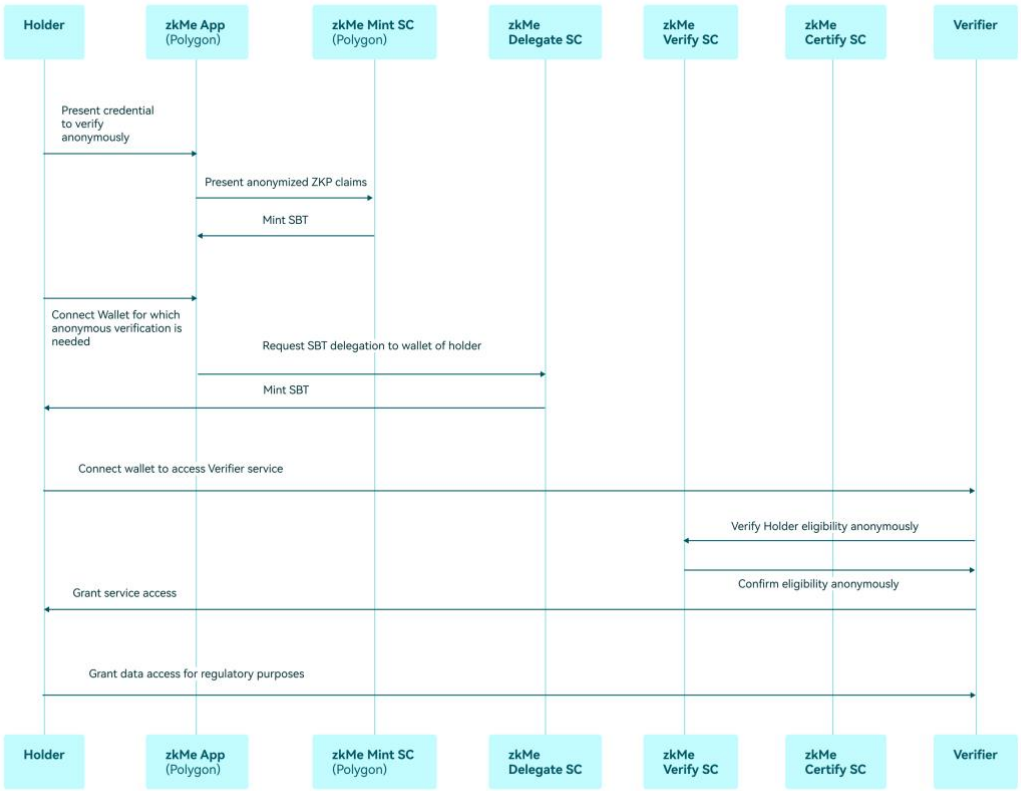

***Figure 4*. zkMe sequence diagram**

4.2 Architectural overview

The following diagram (*Figure 5*) presented below offers a high-level view of the proposed implementation of the zkMe protocol. It takes into account the key observations, requirements, and design considerations previously discussed and depicts the identified solution components that are specific to the web3 ecosystem, as well as their interactions. Each of the components in this architectural overview are described in detail in this chapter.

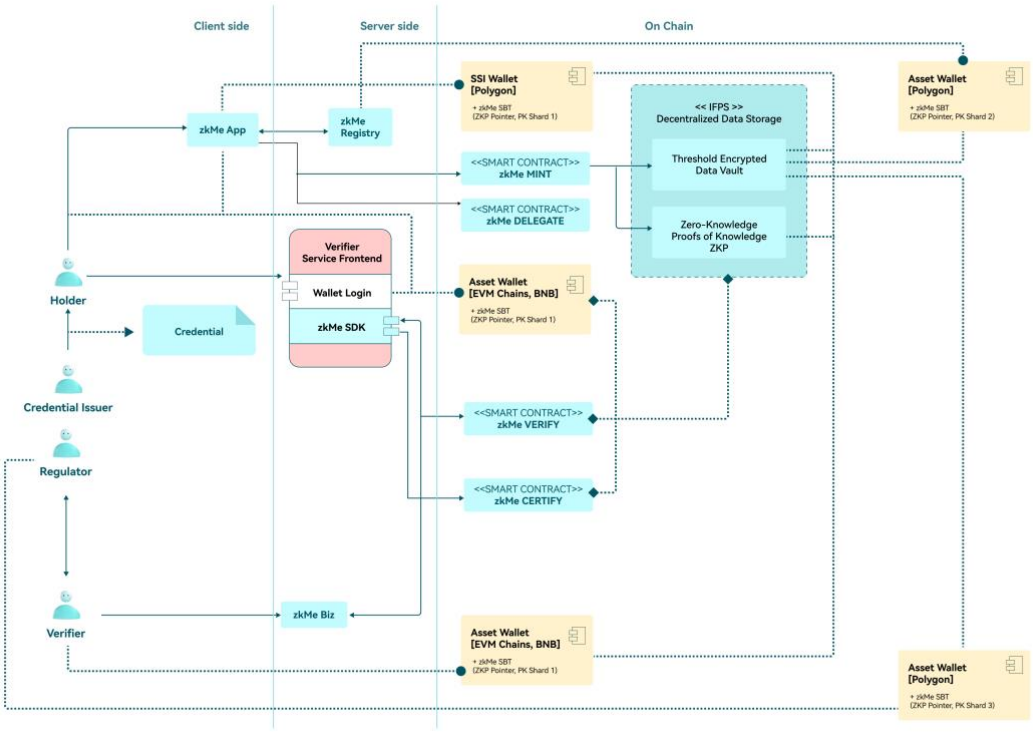

***Figure 5*. zkMe architectural overview**

4.2.1 zkMe Issuer kit

- **zkMe App**

The zkMe App is a secure and decentralized SSI wallet based on MPC cryptography. Users can store and manage their identity information in this wallet, including VC issued by trusted entities like government agencies or financial institutions. These credentials allow individuals to prove their identity or attributes without revealing unnecessary personal information. MPC enables the private keys used to sign and encrypt users' data to be distributed across multiple devices or nodes, eliminating the need for a single device to hold them. This reduces the risk of unauthorized access or data breaches.

The zkMe App also includes OCR and Facial Recognition components for identity verification on mobile devices. OCR technology extracts information from ID documents such as name, date of birth, and ID number, reducing the need for manual input. Facial recognition technology verifies that the person presenting the ID is the same as the photo in the document. These components enhance security and ensure that ID information is not shared beyond the mobile device. In addition, the user is screened against lists of known

criminals, terrorists, or politically exposed persons (PEPs) to identify potential risks on the mobile-end.

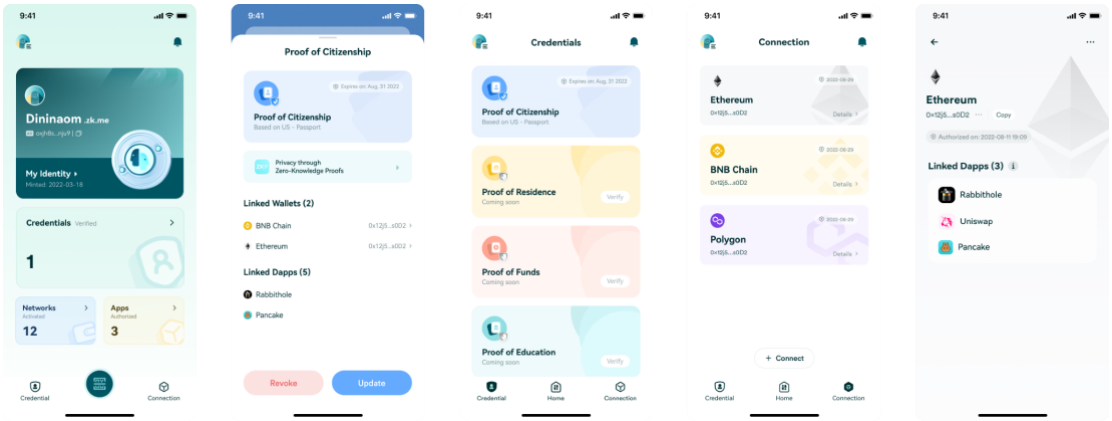

***Figure 6*. The user interface of zkMe App**

- **zkMe SDK**

zkMe SDK is a software development kit that enables web3 developers to integrate SSI and verifiable credential functionality into their applications. It provides a set of libraries and APIs for interacting with zkMe's zk-credentail network, including the ability to create and manage VC, verify their authenticity, and enable secure and private identity verification. The SDK is designed to be easy to use and customizable, allowing developers to tailor the integration to their specific needs and use cases. Additionally, the SDK provides support for standards-based protocols such as DIDs, VC, and ZKPs, enabling interoperability with other SSI solutions in the ecosystem.

The zkMe SDK Pop-up is a feature-rich JS component that has been specifically designed to work seamlessly with desktop browsers and can be integrated as a Pop-up window of services that require user verification. It offers access to all the functionalities of the zkMe App (through mobile QR codes) in a convenient and easily accessible interface. This integration of the zkMe app with desktop browsers makes it easier for users to access and interact with web3 services securely and conveniently.

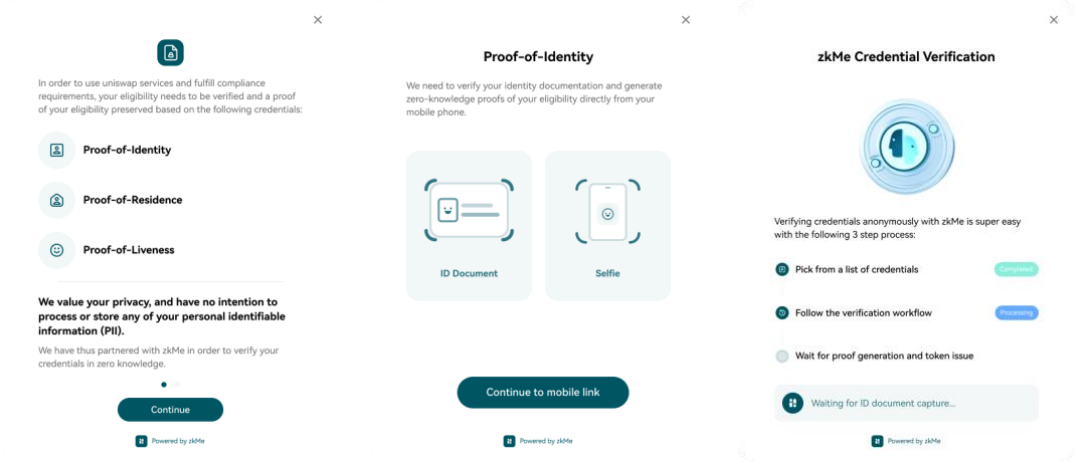

***Figure 7*. The user interface of zkMe SDK Pop-up**

- **zkMe Biz**

zkMe Biz is a powerful dashboard designed to cater to the identification requirements of web3 projects. With zkMe Biz, Verifiers can easily customize preset "verification profiles" based on their specific business needs. For example, they may choose to restrict access to users from certain countries where cryptocurrencies are banned, or they can set up that all users must be over 18 years accessing their services.

zkMe Biz provides a comprehensive set of tools and features to help developers streamline their KYC processes. From customizable verification criteria to detailed analytics and reporting, zkMe Biz makes it easy for developers to ensure that their dApps are compliant with KYC regulations while still delivering a seamless user experience. One of the key features of zkMe Biz is the ability for developers to configure their developer keys. This allows developers to integrate their web3 dApps with zkMe seamlessly, providing a seamless user experience for their customers. One of the key features of zkMe Biz is the ability for developers to configure their API keys, granting them the ability to integrate their web3 dApps with the zkMe Credential Network seamlessly.

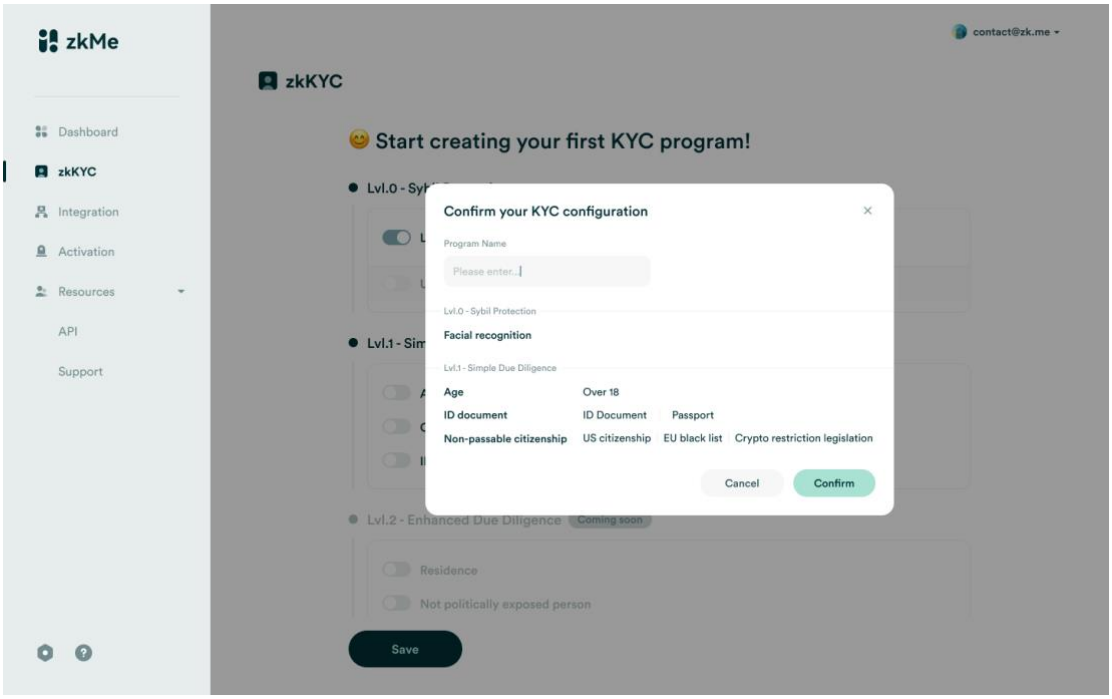

***Figure 8*. The user interface of zkMe Biz**

4.2.2 Asset wallet

The asset wallet is the digital wallet that the Holder uses to interact with web3 dApps. This wallet is typically self-custodial, meaning that users have full control over their assets and private keys. These private keys enable users to transfer digital assets on the blockchain or prove ownership of a particular wallet address associated with the public key.

When using zkMe, the Holder's representation of verification proofs (the SBT) are minted from the SSI wallet accessed through zkMe App (or Pop-up) onto this asset wallet. If a Holder is active in more than one chain ecosystem, a SBT delegated copy is minted for each chain ecosystem in which the user is active in. Once a SBT is minted onto the Holder's asset wallet, the Holder dApp user experience remains unchanged and seamless.

4.2.3 Verifier user interface

The Verifier's user interface is a key component that serves as the primary mean to regulate access control to the Verifier's services. Through the zkMe SDK, zkMe provides a user-friendly way to verify eligibility here. The Verifier can verify presented proofs (i.e. SBT in the Holder's asset wallet) quickly and efficiently through either conventional zkMe web API or smart contracts, or prompt the user to provide new proofs through a user friendly Pop-up window in case no proof was presented.

4.2.4 Cryptographic component

zkMe uses Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) to generate ZKPs that represent the Holder's identity information without disclosing any sensitive data as they run directly on the Holder's end device and only expose the anonymized ZKP results (yes/no statements). These ZKPs are then presented as VPs to the Verifier, who can verify the identity without accessing the underlying data. This approach enhances privacy and security by ensuring that only the necessary information is shared between the Holder and Verifier, and the sensitive data remains protected.

Moreover, the encryption of raw identity documents is also an essential step. To protect the privacy and security of the user's identity documents, threshold encryption is employed. This encryption method utilizes multiple parties to jointly encrypt and decrypt the user's data, ensuring that no single entity has access to the complete data. This technique adds an extra layer of security to the process and reduces the risk of data breaches or unauthorized access to sensitive information.

4.2.5 Decentralized storage

zkMe uses IFPS based decentralized storage a a verifiable data registry. This storage mechanism provides a decentralized and distributed architecture vital to hedge against single points of failure and unauthorized access. Specifically, zkMe uses IFPS to store the single ZKPs that a Holder's SBT points to. This ensures that these proofs are fully reusable, can not be tampered with, or accessed by unauthorized parties.

Furthermore, in cases where the recovery of the original identity documentation (such a photo of the Holder's passport) that were used to derive VPs from is required by law, such will also be stored in decentralized storage and protected through 3-out-of-3 threshold cryptography with split private keys. As described in the implementation details, this solution provides the highest possible degree of Holder privacy while complying with regulatory requirements.

4.2.6 Smart contracts

To enable the verification of Holders in a fully decentralized manner, zkMe developed a suite of smart contracts to allow the protocol itself and the Verifiers to process verifications

in a fully decentralized manner. Please see the zkMe implementation details chapter for more information.

Using the **zkMe Mint** smart contract, zkMe mints the original DID (in form of an SBT) onto the Holder's SSI wallet as soon as the first level of verification is passed. This smart contract is deployed on Polygon. Using the **zkMe Delegate** smart contract, zkMe mints delegate copies of the DID onto the asset wallet and chain selected by the user. This smart contract is currently available for most EVM compatible chains, Solana, Aptos and Sui.

Using the **zkMe Verify** smart contract, Verifiers integrate with the zkMe network by providing a public key address. If the holder has an SBT copy on the asset wallet he/she used to interact with the dAPP, the Verifier can verify the proofs. If the Verifier needs to fulfill legal data access requirements, it can further use the **zkMe Certify** smart contract to request the minting of a special SBT copy, that grants the Verifier a key shard, enabling future data recovery in case bad-actor proceedings are initiated against the Holder.

4.3 zkMe credential schema

This section discusses the credential schema on zkMe network. It is designed to be flexible and extensible, allowing issuers to customize the schema to meet their specific needs while maintaining compatibility with the W3C VC and VP data models; enabling Issuers to issue a wide range of credentials, from educational degrees to professional certifications, while ensuring that these credentials are interoperable and ease of use across different networks and applications.

zkMe specifies the following set of properties that its ZKPs (as VPs) must include as:

1. **Issuer**: The entity that issued the verified.
2. **Subject:** The DID to whom the credential pertains.
3. **Type:** The type of credential being issued (e.g. Proof-of-Citizenship, Proof-of-Residence).
4. **Issuance Date:** The date on which the credential was issued.
5. **Expiration Date:** The date on which the credential expires.
6. **Claims:** The specific eligibility the VP is attesting to (e.g. Adulthood - Is the holder over 18 years old?).

7. **Proof:** A cryptographic proof that the credential was issued by the specified issuer and has not been tampered with since issuance.

In addition to these standard properties, VCs and VPs on zkMe may also include other properties or custom extensions, depending on the needs of the Issuer or the network's requirements as a whole. It's important to note that the content of VCs and VPs issued on zkMe is determined by the Issuer, and may vary depending on the type of credential being verified and the specific claims being attested.

In the zkMe zkKYC solution, the following information elements and descriptions are included:

Issuer DIDs: The public decentralized identifiers (DIDs) of the issuer are published in the Verifiable Data Registry. This information enables the holder to verify the authenticity and trustworthiness of the issuer.

Holder DIDs: The private DIDs of the Holder towards a particular Issuer are known only to the Holder and the Issuer. These DIDs enable the Holder to authenticate themselves to the Issuer and provide proof of their Identity.

4.4 zkMe's security goals

The zkMe protocol is designed to fulfill the highest degree of privacy for the Holder. In order to prove the security of private information in an end-to-end manner, we define a security proof around zkMe's ideal functionality (represented by $\mathcal{F}$) that captures the following security goals:

1. **Anonymity/Unlinkability:** Anonymity means that the proposed system should prevent the probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ from finding any identity information of the Holder.

2. **Unforgeability:** Any PPT adversary $\mathcal{A}$ cannot forge a fake SBT with a fake document, and his SBT will never be transferred from SSI wallet to another SSI wallet.

3. **Traceability:** When bad actor proceedings are initiated to trace back the ID data of a Holder, a PPT adversary $\mathcal{A}$ can not stop this process or cheat it to trace back to another user.

Chapter 5.7 describes the security proofs based on three theorems derived from the ideal functionality in detail.

5. zkMe technical details

This chapter elucidates the intricacies of each constituent within the zkMe Network, encompassing the zkMe App, facial recognition, Optical Character Recognition (OCR), threshold encryption, and smart contracts. Additionally, the security model is delineated at the conclusion to showcase the robust security underpinning zkMe.

5.1 zkMe App: MPC-based SSI wallet

The zkMe App is the front-end access to a decentralized and secure SSI wallet that uses MPC cryptography. It allows users to manage and store their identity information, either as VCs or anonymized VPs (in form of SBT). With MPC technology, private keys used for data encryption and signing are distributed across multiple nodes or devices, removing the need for a single device to store them. This significantly reduces the risk of data breaches or unauthorized access. The following sequence diagram (*Figure 9*) shows the sequence of a holder creating their SSI wallet.

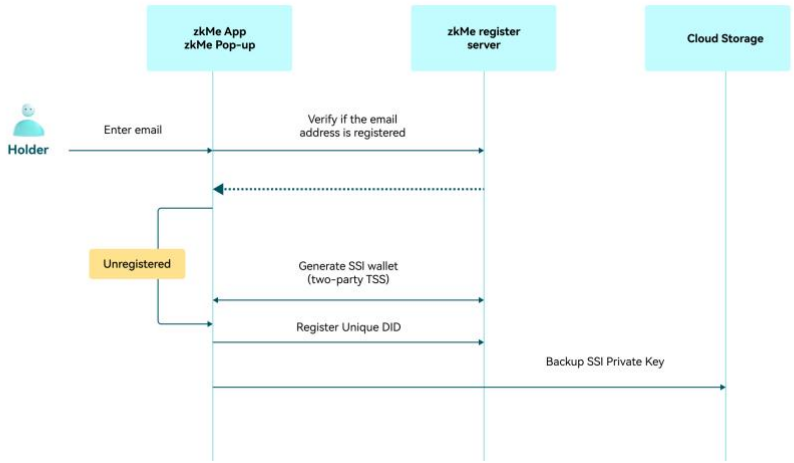

***Figure 9*. Sequence diagram SSI wallet creation**

5.1.1 Notation

This section shows the notation for zkMe's MPC-based SSI wallet creation.

Symbol	Notion	Symbol	Notion

P	Elliptic curve base point	x	Global private key(no one knows it)(type: scalar)
p	Order of the base point	h	Global public key(type: ecpoint)
Z_n	Operation field of the elliptic curve	x_i	party-i 's private key (key share of)(type: scalar)
+	Numerical addition	h_i	party-i 's public key (key share of)(type: ecpoint)
*	Numerical multiplication	c_i	party-i 's commiment(type: scalar)
$\oplus$	Elliptic curve point addition operation	r_i	Random number(type: scalar)
$\otimes$	Elliptic curve point doubling operation	q_i	party-i 's private share of q(type: scalar)
$\boxplus$	Paillier addition operation (addends can be plaintext or ciphertext encrypted with the same homomorphic public key)	Q_i	party-i 's share of Q(type: ecpoint)(if its x-coordinate does not equal 0, R=Q)
$\boxtimes$	Paillier multiplication operation (multiplier must be plaintext)	R	First party of signature(type: scalar)
H	keccak256	S	Second party of signature(type: scalar)

5.1.2 Phase 1: Public key negotiation

The following steps are processed during the SSI wallet public key negotiation:

1. Generate the keypair(x_1, h_1) for party-1 regarding h, and make a commitment $c_1 = H(h_1, r_1)$ for h_1; Generate the keypair (x_2, h_2) for party-2 regarding h, and make a commitment $c_2 = H(h_2, r_2)$ for h_2.

Function	**Operation**
generate_key_share(m, n) at party-i	$x_i \xleftarrow{R} [m, n], h_i = x_i \otimes P$
rand(p) at party-i	$r \xleftarrow{R} [1, p]$
generate_commitment(m, n) at party-i	$c = H(m \,\|\| n)$
verify_commitment(c, m, n) at party-i	$c' = H(m \,\|\| n)$, check $c == c'$

2. Party-1 sends c_1 to Party-2.

3. Party-2 sends c_2 and the preimage (h_2, r_2) ofc_2 to Party-1.

4. Party-1 verifies $c_2 = H(h_2, r_2)$and then sends the preimage (h_1, r_1) of$c_1$to Party-2.

5. Party-2 verifies $c_1 = H(h_1, r_1)$.

6. Party-1 and Party-2 each calculate $h = h_1 + h_2$, confirm that the results are the same, and jointly announce the global public key as h.

Function	**Operation**
compute_global_pubkey(m,n) at party-i	$h = m \oplus n$

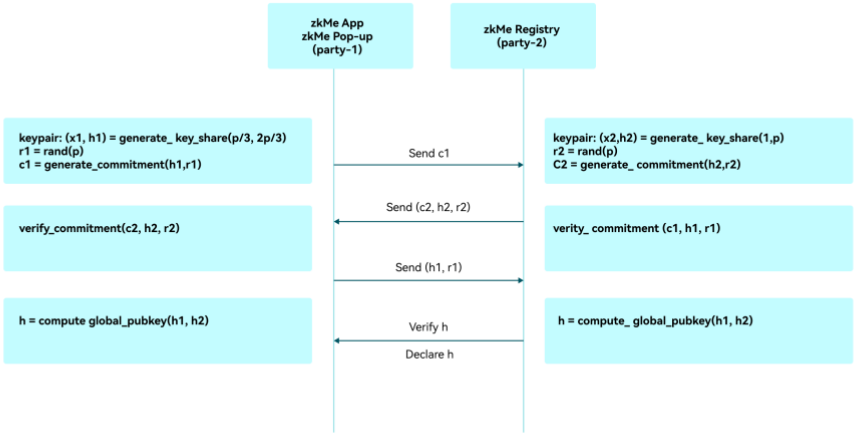

***Figure 10*. SSI wallet public key negotiation**

5.1.3 Phase 2: Signature

The following steps are processed for SSI wallet signature:

1. Negotiation of a Consistent Random Number R

2. Party-1 generates a Paillier keypair $(psk, ppk)$and encrypts its private key x_1 with the Paillier public key ppk： $c_{key} = ppk.encrypt(x_1)$. Then, party-1 sends c_{key} to party-2.

3. Party-1 generates a keypair (q_1, Q_1) regarding$Q$ and makes a commitment $c_3 = H(Q_1, r_3)$ for Q_1； Party-2 generates a keypair (q_2, Q_2) regarding Q and makes a commitment $c_4 = H(Q_2, r_4)$ for Q_2.

4. The joint declaration R

   a. Party-1 sends c_3 to party-2.

   b. Party-2 sends c_4 and the preimage (Q_2, r_4) of c_4 to party-1.

   c. Party-1 verifies $c_4 = H(Q_2, r_4)$ and sends the preimage (Q_1, r_3) of c_3 to party-2.

   d. Party-2 verifies$c_3 = H(Q_1, r_3)$.

   e. Party-1 computes$Q = q_1 * Q2$and party-2 computes $Q = q_2 * Q1$. Both parties confirm $Q.x \ mod \ p \ != \ 0$ and the results are the same, then they jointly declare $R = Q.x$, otherwise, they restart the phase.

Function	Operation
compute_Q() at party-i	$Q = q_i \otimes Q_{3-i}$
check_Q() at party-i	$Q.x == 0$? True: repeat phase 2.1.2 False: declare $R = Q.x$

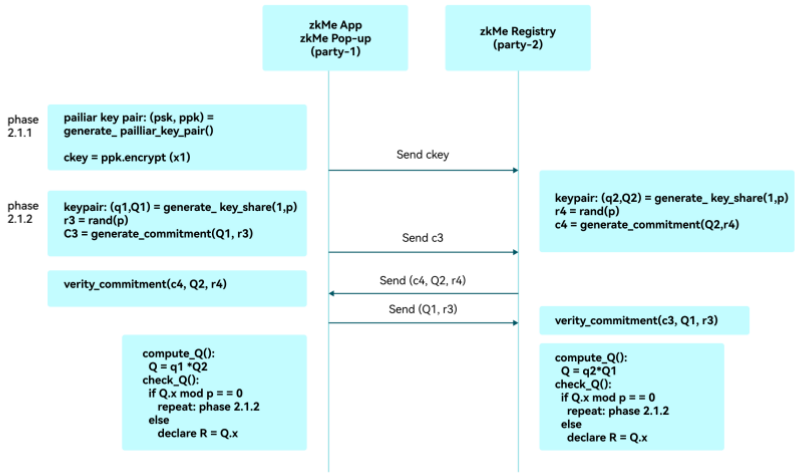

***Figure 11*. SSI wallet signature joint declaration R**

5. The joint signature S

a. Party-2 calculates the hash digest of the computation data as z, and uses the ckey obtained in Phase 2.1.1 and its own private key to calculate the intermediate ciphertext itm_enc. Finally, Party-2 sends itm_enc to Party-1.

Function	Operation
compute_intermediate_enc() at party-2	$\rho \xleftarrow{R} [1, p^2]$, $q_2^{-1} = invert(q_2, p)$, $tmp = \rho \times p + ((q_2^{-1} \times z) \bmod p)$, $enc_1 = ppk.encrypt(tmp)$, $enc_2 = ppk.encrypt(x_2)$, $enc_3 = c_{key} \boxplus enc_2$, $enc_4 = (R * q_2^{-1}) \boxtimes enc_3$, $enc_5 = enc_1 \boxplus enc_4$, itm_enc = enc_5

b. Party-2 sends itm_enc to Party-1, and Party-1 calculates the second part of the signature$S$. Finally, Party-1 announces.

Function	Operation
compute_S() at party-1	$S' = psk.decrypt(c_5)$, $q_1^{-1} = invert(q_1, p)$, $S'' = q_1^{-1} \times S' \bmod p$, $S = min(S'', p - S'')$

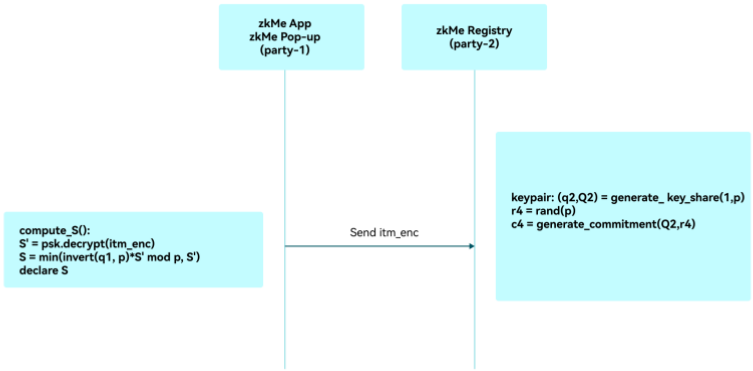

***Figure 12*. SSI wallet joint signature S**

5.2 Facial recognition

zkMe App and zkMe SDK rely on advanced Artificial Intelligence (AI) technologies to perform Facial Recognition for the purpose of unique credential holder identification. The system leverages computer vision, employing both deep learning and traditional image processing techniques to effectively process and analyze image data. Such a combination of technologies enables the automatic detection, identification, and verification of faces, thus accomplishing critical functions including identity authentication and security monitoring. The cutting-edge nature of zkMe's AI technologies and their significant contributions to credential identification and security make it an important development in the field.

In contrast to conventional template matching-based algorithms, the zkMe system leverages advanced artificial intelligence technology to achieve accurate and efficient facial recognition (*Figure 13*). The underlying objective of facial recognition is to match an input face image

with known identity images or databases. This involves the extraction of relevant facial features from the input image, which are then compared with the features of the target image or database to ascertain the identity of the input face image. The zkMe approach utilizes a four-stage process that is both effective and robust.

***Figure 13*. zkMe App facial recognition flowchart**

5.2.1 Face detection

According to *Li and Liang (2021)*, facial detection serves as the first module in the facial recognition process to localize a face within an image using specific algorithms to output the coordinates of the relevant region of interest. By accurately identifying the face, downstream tasks can isolate and analyze the facial features, which have significant value in facial recognition systems. The facial detection approach primarily employs regression-based techniques to locate the facial landmarks in the image, unlike optical character recognition (OCR) which relies on a set of pre-defined reference patterns for detection. Notably, facial detection methods can leverage existing object detection algorithms such as the Single Shot Detector (SSD) proposed by *Girshick et al. (2014)* to improve their performance in recognizing faces within an image.

With the advent of convolutional neural networks (CNNs) for universal object detection, facial detection has advanced into a new era. CNNs offer a promising approach to automatically learn facial features, thereby achieving improved accuracy and faster facial detection. Facial detection networks can be broadly classified into two categories: single-stage methods and multi-stage methods. These two methods use different approaches for face detection, and have shown success in various applications. As noted by *Liu et al. (2016)*, single-stage methods directly predict the position and size of faces from images, exemplified by the YOLO *(Redmon and Farhadi, 2018)* and SSD *(Liu et al., 2016)* series algorithms. On the other hand, multi-stage methods employ a two-stage strategy, first generating candidate regions and then performing classification and regression on each region, such as the MTCNN *(Zhang et al., 2018)* and RCNN *(Ren et al., 2015)* series. The SSD algorithm, a representative single-stage object detection method, achieves target detection by one

forward pass after feature extraction with various receptive fields. Many current fast facial recognition algorithms are inspired by and modified based on the SSD algorithm, resulting in high speed and accuracy.

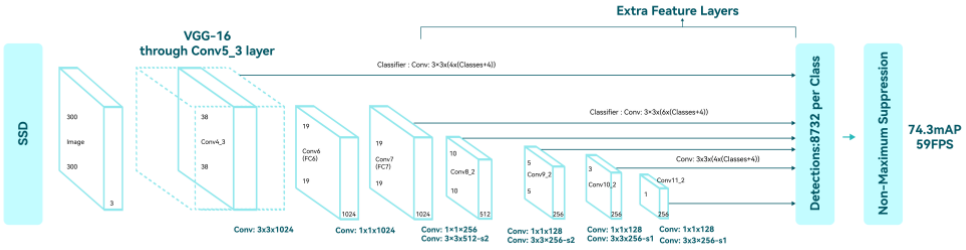

Figure 14*. SSD network architecture diagram, *Liu et al., (2016)

zkMe employs a regression-based object detection network with multi-task fusion to facilitate face detection. The model is designed to simultaneously detect the position of the face and locate the facial landmarks. The use of multi-task learning in training the model enables it to better capture the underlying details and remove interference that may arise from single-task learning. To cater to different scenarios on both the client and server sides, zkMe leverages two solutions with varying model sizes. In terms of network architecture, zkMe incorporates the Separable Convolution module and utilizes the Bottleneck structure to reduce model parameters while retaining critical information.

In addition, for face recognition scenarios, zkMe's model only selects several layers of features that are more suitable for faces, instead of a feature pyramid model with as many layers as SSD. According to experimental results, zkMe's model can adapt well to current scenarios, achieving an average precision (AP) of 98.61% on the test set and millisecond-level runtime on modern i5 servers. To ensure the practicality of the application, zkMe applies post-processing techniques, such as statistical outlier removal, non-maximum suppression, and key point tracking. These techniques improve the application's performance by ensuring fast, accurate, and stable operation.

5.2.2 Faceprint creation

Facial keypoint recognition is an essential sub-task in the field of computer vision, which follows the initial step of facial detection. The primary objective of this task is to analyze the area detected by facial detection and extract the positional information of the predefined keypoints, including eyes, nose, mouth, and eyebrows. The collection of these keypoint positions is typically referred to as the "Faceprint." The accuracy of facial keypoint

recognition is critical for a range of applications, including facial expression analysis, head pose estimation, and facial recognition.

In the field of facial keypoint detection, traditional methods typically involve manually extracting features such as Histograms of Oriented Gradients (HOG) and Local Binary Patterns (LBP) from facial regions, and then using classifiers or regressors to predict keypoint locations. This approach was proposed by *Dalal and Triggs (2015)* in "Histograms of Oriented Gradients for Human Detection," and further developed by *Ojala et al. (2002)*.

However, with the application of deep learning, the speed and accuracy of facial keypoint detection have significantly improved. Deep learning models can be categorized into different types, one of which is the single-task cascaded model. An example of this model is the Deep Alignment Network (DAN), proposed by *Kowalski et al. (2017)*. The DAN model employs convolutional neural networks to regress keypoint locations from coarse to fine levels, similar to the boosting method used in traditional machine learning. A more common type of deep learning model for facial keypoint detection is the multi-task joint model, which combines tasks such as face detection and keypoint detection in the same network. This approach enables the lower-level modules to learn more rich and important features through different tasks, leading to improved robustness and generalization ability. Examples of multi-task joint models include the Multi-Task Cascaded Convolutional Networks (MTCNN) proposed by *Zhang et al. (2014, 2015).*

Grishchenko et al. (2020) present a workflow for the mediapipe face mesh model, as depicted in *Figure 15*. This model utilizes a face detector based on BlazeFace *(Bazarevsky & Kartynnik, 2019)* to obtain the corresponding facial region. Subsequently, a convolutional neural network extracts the corresponding facial features and converts them into regression positions. To achieve more accurate detection, attention modules are added to specific regions such as the eyes, eyebrows, and mouth, ultimately leading to dense keypoint detection of the face.

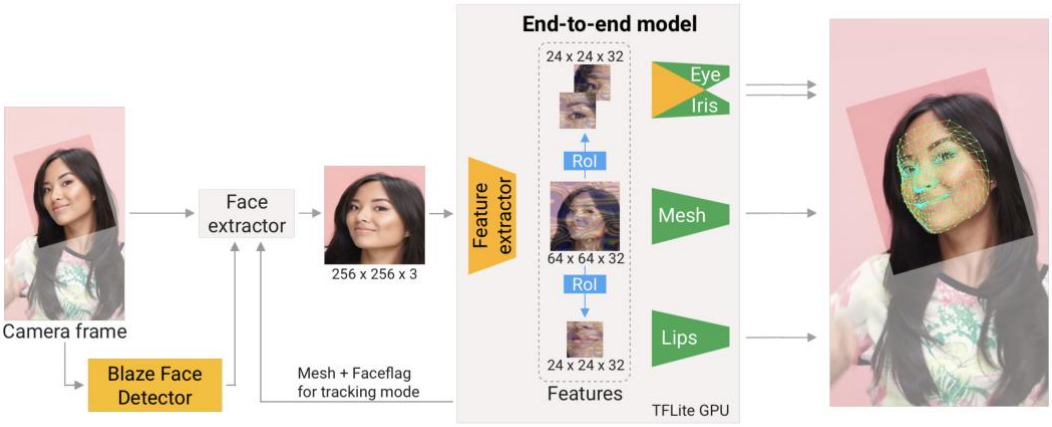

Figure 15*. Mediapipe face mesh model workflow, *Grishchenko et al. (2020)

The facial landmark model developed by zkMe relies on a dense keypoint detection algorithm that enables the extraction of the corresponding facial region through face detection. zkMe's face detection model is a multitasking model that is capable of obtaining the target facial region and the corresponding facial key points simultaneously. This allows for facial alignment at the corresponding positions prior to processing by the facial keypoint model. However, it has been observed that keypoint drift occurs during overall facial movement in facial keypoint detection models. To address this issue, post-processing methods such as smoothing, outlier removal, and Euro filter have been incorporated into the model to ensure system stability. Additionally, a coordinate point tracking method has been introduced to improve the overall processing speed and avoid running the face detection module repeatedly. The method exploits the fact that facial positions in face detection scenarios are unlikely to rapidly deviate significantly from the previous frame's position. Consequently, previous position data can be used for quick screening to bypass the detection module, thus speeding up the overall model's speed. In the event of sudden changes in facial position or occlusion, the face detector is introduced to enable more precise detection and to address the issue of facial loss.

5.2.3 Faceprint alignment

Facial alignment is a critical preprocessing step in many computer vision applications, particularly those involving face recognition and analysis. The primary objective of facial alignment is to aid algorithms in accurately estimating the pose and position of faces, thereby facilitating downstream tasks that require feature extraction. Traditional approaches to facial alignment typically rely on manually designed features and algorithms, such as

facial contour lines and eye positions, to estimate facial rotation and translation. However, such methods are less effective when dealing with changes in pose and expression. In contrast, deep learning techniques employ deep neural networks to learn features from images and perform more precise facial alignment. Facial alignment methods based on facial landmarks can be broadly categorized into regression-based and alignment-based methods. Regression-based methods train deep neural networks to map input facial images to a set of specific facial landmarks, such as the eyes, nose, and mouth. Once these landmark positions are determined, facial alignment can be achieved through a simple affine transformation.

According to *Kowalski et al. (2017),* regression-based methods have the advantage of high precision, but they necessitate a substantial number of accurately labeled landmarks for training. In contrast, alignment-based methods directly utilize deep neural networks to align input facial images to a fixed template. This approach does not mandate pre-labeled facial landmarks, thereby facilitating expansion to new datasets and tasks. Although alignment-based methods offer higher efficiency, they may compromise accuracy.

The proposed method, zkMe, utilizes a regression-based approach for face alignment, which is iteratively applied through multiple detections to improve the robustness of the overall task. In the face detection module of zkMe, the method not only identifies the facial region but also extracts the positions of the key points associated with the five facial features, providing the initial key point information. Our experiments revealed that applying affine transformation and other conventional methods can cause distortion and deformation of the face, leading to challenges in key point recognition and face comparison tasks. Additionally, the key point positions of inclined faces may not be accurately captured, thus negatively affecting face region cropping. In response, zkMe addresses this issue by rotating the entire image based on the key points of the facial features following the initial detection to correct for large inclination angles. Subsequently, face detection is repeated on the rotated image to obtain more accurate position and key point information and reduce the impact of affine transformation. Finally, based on the second face position, zkMe performs the final alignment, resulting in a clear, frontal face region that is suitable for downstream tasks.

5.2.4 **Faceprint comparison**

Facial feature comparison is a vital task in the field of facial recognition, which involves comparing two or more facial images to determine if they belong to the same person. This task can be divided into two categories, namely one-to-one and one-to-many comparisons, depending on the context. To accomplish this, the first step is to convert the aligned face into a feature vector through a convolutional neural network. This feature vector is then compared with the target image or the image in the database to make a comparison judgment based on the feature distance. The primary objective of this study is to explore the efficacy of facial feature comparison in different scenarios. *Figure 16* is provided to illustrate the differences between various facial comparison tasks.

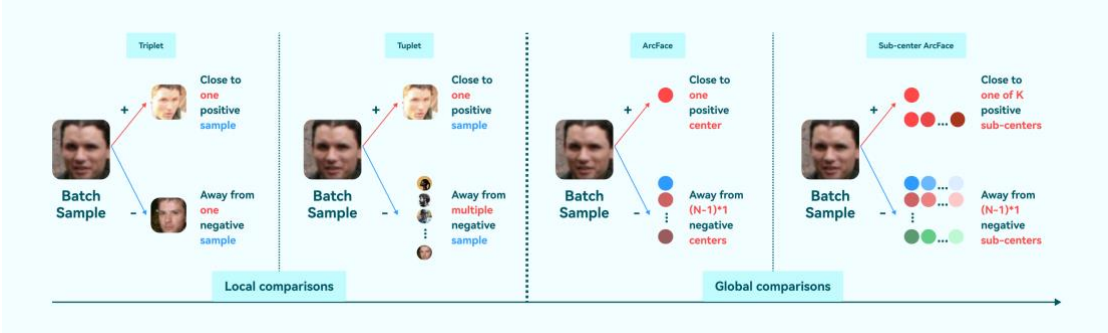

Figure 16. **Illustration of different face comparison tasks,** ***Deng et al. (2019)***

Facial feature comparison can be conducted through two distinct approaches, namely traditional and deep learning methods. The traditional methods rely on the combination of feature extraction and similarity measurement to accomplish facial feature comparison tasks. The feature extraction process involves the utilization of conventional computer vision techniques such as Local Binary Pattern (LBP), Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA), among others, to extract features from facial images. On the other hand, the similarity measurement process entails the use of various distance metrics such as Euclidean distance, cosine similarity, and Mahalanobis distance to compare the similarities between two features. The combination of these methods has been widely applied in the field of facial recognition technology to improve the accuracy and efficiency of facial feature comparison.

According to the work of *Deng et al. (2019)*, the zkMe approach primarily depends on the ArcFace loss function. As illustrated in figure below, this method differs from the Center Loss technique proposed by *Wen et al. (2016)*, which aims to increase intra-class compactness by penalizing the distances between the features of the posterior layers in the Euclidean space and the corresponding class centers. However, this method faces challenges when

updating actual centers during training due to the sharp increase in the number of facial categories. In contrast, the ArcFace loss function achieves the separation of different faces by adding margins to the loss function. Specifically, facial images are projected into a feature space with small intra-class distances and large inter-class distances. The approach involves calculating the angles between different features using the inverse cosine function and adding additive angle margins to the target angles. Finally, the cosine function and feature norm are utilized to scale all feature results, enhancing intra-class compactness while also strengthening inter-class differences.

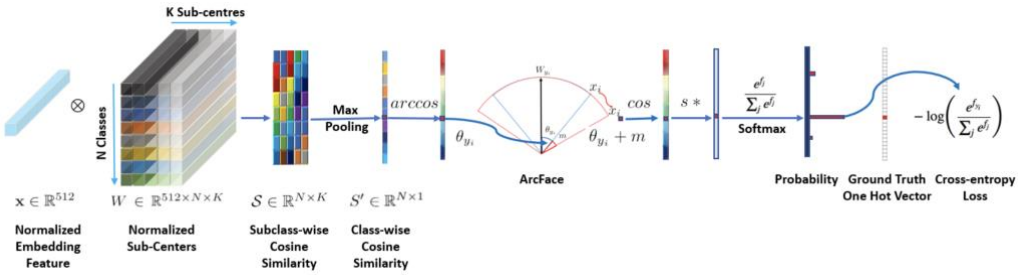

Figure 17*. The implementation process of ArcFace loss, *Deng et al. (2019)

5.3 Optical character recognition

The document recognition approach employed by zkMe is rooted in the well-established Optical Character Recognition (OCR) technology widely utilized in the industry. In conjunction with traditional template matching techniques, the zkMe methodology accomplishes rapid and precise identification of diverse document types and from various countries. The existing mature OCR process in industry conventionally involves two core stages: text position localization and character recognition.

***Figure 18*. zkMe OCR flowchart**

5.3.1 Text localization

Text localization algorithms are primarily designed to identify and extract character regions from input images, which are then used to facilitate downstream text recognition tasks. These algorithms can be broadly classified into two categories: target detection and image segmentation. Target detection algorithms are focused on identifying regions in the input image that contain text, while image segmentation algorithms segment the input image into regions that contain text and those that do not. Both of these approaches have shown promising results in text localization, and the choice of algorithm depends on the specific application requirements and characteristics of the input data.

According to *Liao, Shi, and Bai (2018)*, target detection methods, similar to traditional general object detection algorithms, utilize preset anchor detection boxes and direct pixel regression to acquire the coordinates of the target regions through regression-based approaches. However, unlike general object detection, character detection algorithms in optical character recognition (OCR) typically modify the anchor pre-selection box and convolution kernel size according to the text scene characteristics. These methods perform well for regular-shaped text, but their performance is relatively poor for detecting irregular-shaped text. *Figure 20* illustrates the recognition process of Textboxes++, which is a typical model based on target detection.

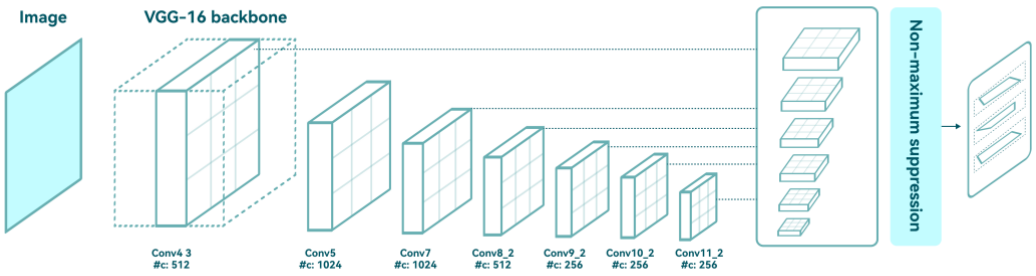

Figure 19*. The TextBoxes++ model, *Liao, Shi, and Bai (2018)

According to *Liao et al. (2020)*, the approach of position localization through image segmentation involves direct mapping of the original image to a probability map of identical size, followed by obtaining the final text regions via threshold filtering. This method effectively addresses the issue of curved text and performs well in detecting normal text. As segmentation is a relatively simpler task than target detection, the feature extraction model

is more efficient. The authors illustrate this process with a threshold feature map in the DBNet, which represents a typical image positioning process based on segmentation.

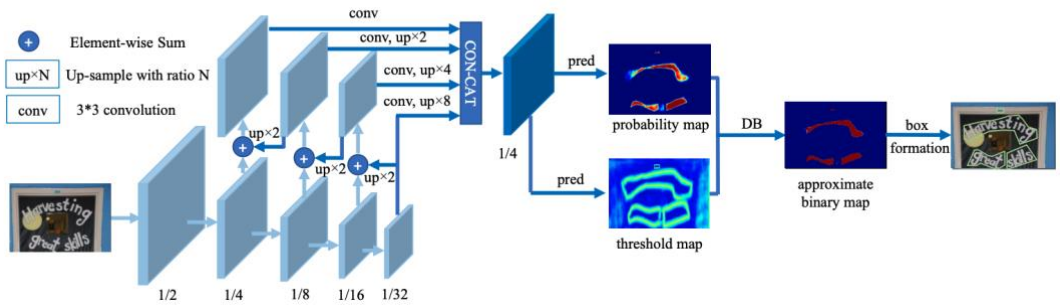

Figure 20*. The DBNet model, *Liao et al. (2020)

The zkMe detection model relies on segmentation, albeit with a departure from conventional segmentation models, wherein threshold binarization and time-consuming traditional image processing techniques are employed in the post-processing stage. The DBNet model is employed in this study, as depicted in the figure below, to enable the network to automatically learn the segmentation threshold, without necessitating any additional binarization or post-processing steps. This is achieved through the use of a differentiable binarization function that approximates a step function, thus enabling the network to learn different text segmentation thresholds during the end-to-end training process. The automatic threshold adaptation method offers not only an improvement in accuracy, but also simplifies post-processing, thereby leading to state-of-the-art results in text detection.

5.3.2 Character recognition

In the realm of character recognition, deep learning algorithms have emerged as a promising approach. In the context of such algorithms, the process of character recognition can be broadly decomposed into several stages. The initial stage involves rectification of the image to correct for any slanting or curvature, which in turn aids in feature extraction in subsequent steps. Subsequently, a convolutional neural network (CNN) is deployed to extract features from the rectified image. The CNN's ability to learn and generalize patterns in the data makes it a popular choice for feature extraction in character recognition tasks. In the field of natural language processing, various sequence models have been developed to improve the ability of capturing complex features of text sequences. For instance, Long Short-Term Memory (LSTM) model introduced by *Hochreiter and Schmidhuber (1997)* and Transformer model proposed by *Vaswani et al. (2017)* are widely used sequence models that

have shown superior performance in many NLP tasks. These models are designed to handle sequential information and capture long-term dependencies in text data, which allows them to make accurate predictions. Ultimately, the target character is predicted based on the learned sequence features.

The present paper discusses two prominent approaches to text recognition, namely the Connectionist Temporal Classification (CTC) and Sequence-to-Sequence (Seq2Seq) methods. These approaches have become increasingly popular and are currently considered mainstream in the field of text recognition. The primary distinction between these two approaches lies in the decoding stage. While the CTC-based algorithm employs a sequence generated by encoding and inputs it into CTC for decoding, the Seq2Seq-based method utilizes a Recurrent Neural Network (RNN) module for decoding. Both methods have demonstrated their effectiveness in practice, and thus have gained widespread adoption among researchers and practitioners alike.

According to *Shi et al. (2017)*, the Convolutional Recurrent Neural Network (CRNN) is the most common algorithm used for image-based sequence recognition and text recognition. The feature extraction component of the CRNN utilizes popular convolutional structures such as ResNet *(He et al., 2016)* and MobileNet *(Howard et al., 2017)*. However, due to the unique characteristics of text recognition tasks, there is a wealth of contextual information in the input data, which convolutional neural networks tend to overlook, as they focus on local information and lack the capacity to model long dependencies. Consequently, exploring contextual relationships between text using only a convolutional neural network becomes problematic. To overcome this challenge, the CRNN algorithm introduces bidirectional Long Short-Term Memory (LSTM) to improve contextual modeling. Experimental results demonstrate that the bidirectional LSTM module can effectively extract contextual information from images. Finally, the output feature sequence is input into the Connectionist Temporal Classification (CTC) module for direct sequence decoding.

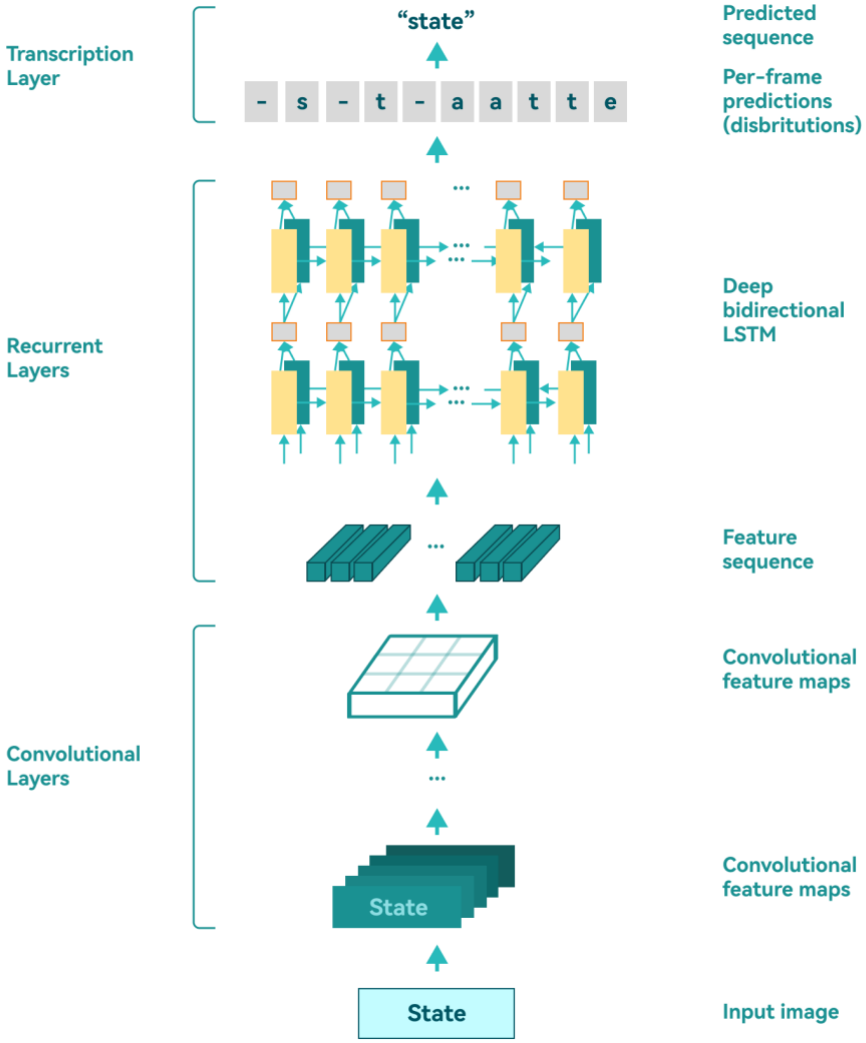

Figure 21*. The CRNN model, *Shi et al. (2017)

In order to achieve stable and more robust performance, the zkMe system must be able to recognize multiple languages. To this end, an industry-standard model based on Connectionist Temporal Classification (CTC) is employed. The overall model architecture consists of a backbone network, recursive layers, and transcription layers. First, the bottom-level backbone network extracts the feature sequence from the input image. The recursive network in the recursive layer then converts the image features into sequence features and predicts the feature distribution for each part. Finally, the transcription layer employs a full-connection module and a softmax module to convert the final sequence results. The end-to-end training prediction is accomplished through CTC Loss, which does not require sequence alignment. Overall, the zkMe system achieves high performance and stability through the use of this model architecture.

5.4 Zero-Knowledge Proofs

SNARK JS and Circom are the core tools for implementing zkMe technology. Snarkjs is a JavaScript library for zk-SNARKs that is used to generate and verify ZKPs, supporting multiple zero-knowledge proof systems such as Groth16 and PLONK. Circom, on the other hand, is a zk-SNARKs circuit DSL (domain-specific language) used to write circuit constraints. Developers can write circuit constraints in a more readable code using circom and then use snarkjs to generate and verify ZKPs.

In zkMe, the Groth16 algorithm is used, which requires a trusted setup to generate a parameter set known as a common reference string (CRS) for each statement (also known as circuit). Once this CRS is generated, it can be used for proving different statement instances throughout the system's entire lifetime. The Groth16 setup is performed through a multi-party computation setup ceremony that ensures the security of the ZKPs system as long as at least one party is honest.

5.4.1 zkMe's trusted setup

In zk-SNARK systems, trust is established through the use of cryptographic techniques that ensure the integrity and security of the process. There are two phases for zkMe's trusted setup.

Phase 1 is the Perpetual Powers of Tau Ceremony that can be forever on-going and used by all circuits. zkMe uses the publicly available Ptau parameters from Snarkjs as part of its zk-SNARK implementation. These parameters will form the base for Phase 2 and are generated through a process known as the "toxic waste" process, which involves using a random number generator to generate some initial parameters, and then performing a series of computations to generate the parameters for zk-SNARK.

Phase 2 is the Circuit/Statement-specific Setup Ceremony that needs to be done for each circuit in zkMe. A circuit is a representation of the computation to be performed in zk-SNARK. Before zkMe goes into production, zkMe will customize and publish all circuits in protocol to ensure that circuits correctly represent computation and do not contain any vulnerabilities that could be exploited by an attacker. Then, based on circuit files and the

publicly available Ptau parameters in Phase 1, zkMe generates the initial keys to start Phase 2 and contributes it with some randomness. Phase 2 can continue for as long as there are participants willing to contribute with their resource of entropy. And we will open the entry to welcome every potential participant to contribute to the latest keys. Once enough participants have contributed, we will terminate this relay process and publish the protocol keys so that the protocol can be used in production after that. To learn about the overall progress, please visit our website.

In summary, each computation verification problem in the zkMe is converted into an R1CS/circuit file, corresponding to a pair of proving/verification keys.

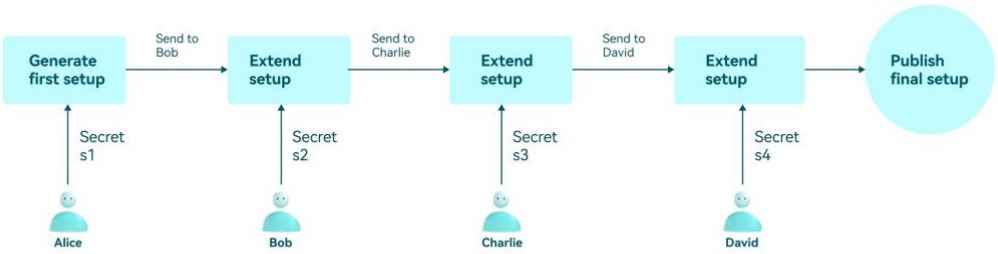

***Figure 22*. Multiple participants contribute with secrets in relay**

5.4.2 Proving and verification

zkMe uses the Groth16 algorithm as its underlying zk-SNARK construction. The Groth16 algorithm is a variant of the pairing-based SNARK construction, which allows for efficient verification of complex computations in zero-knowledge. It uses a structured reference string to generate zk-SNARK proofs. The structured reference string consists of a set of public parameters that are used to represent the computation being performed, and it has been generated in 5.4.1 Trusted Setup Ceremony.

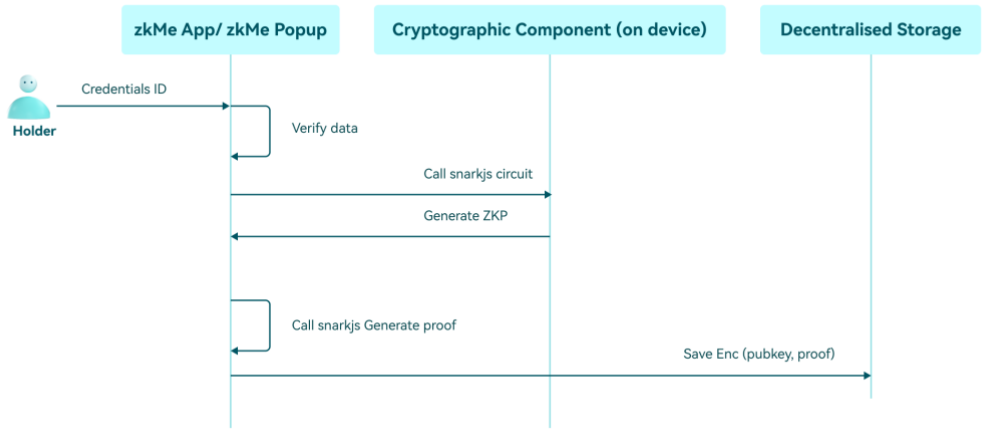

***Figure 23*. zkMe ZKP generation sequence diagram**

The proving and verification process of the Groth16 algorithm through the following steps:

Proving process:

1. The prover queries the circuit file and proving key corresponding to statement.
2. The prover calculates the intermediate and output signals of this circuit as witness based on its own inputs. Note that the witness is as important as private input and should not be disclosed to anyone.
3. The prover calculates the proof based on witness and proving key.
4. Submit public inputs and proof to the Verifier.

Verification process:

1. The Verifier receives the proof and verification key of CRS.
2. With the proof and verification key known, the checking equation can be verified to hold by the pairing function.
3. If the checking equation holds, the proof is valid; otherwise, it is invalid.

5.4.3 zk-SNARK example

Let's assume a scenario where Alice wants to go to a bar for a drink and the bar staff needs to verify that Alice is over 18 years old. In the interests of privacy, Alice does not want to reveal her real age. This is a good scenario for a real application of ZKPs, where Alice can prove her age meets the established requirements of zk-SNARKs.

First, before that, Alice already had a verifiable credential (VC) issued by the federal government regarding the time of birth. It is stored in blockchain and is directly linked to the federal government's public key. Essentially, the VC is a hash value and is calculated according to the following rules.

$$VC_{A,age} = hash(pk_A, birthdayGMTts_A, r)$$

where pk_A is Alice's public key in her digital wallet, $birthdayGMTts_A$ is the GMT timestamp of Alice's birth, and r is a random salt value. The latter two are stored in Alice's digital wallet.

First, before Alice generates the proof, the staff will calculate the interval between today's timestamp and the timestamp of the day 18 years ago, e.g., today is 2022-10-29 00:00:00, the timestamp is 1666972800, and the timestamp of 2004-10-29 00:00:00 eighteen years ago is 1098979200. The interval between the two timestamps is 567993600, which is denoted as $intervalThreshold$. Alice then generates a proof in her wallet that she is indeed older than 18 years old in the following steps: Alice defines $(sk_A, birthdayGMTts_A, r)$ as the private input and defines $(pk_A, VC_{A,age}, todayGMTts_A, intervalThreshold)$ as the public input to generate the proof as follows:

$$zk - SNRKK(sk_A, pk_A, VC_{A,age}, birthdayGMTts_A, todayGMTts_A, intervalThreshold, r)$$

The proof has the following steps:

- check that (sk_A, pk_A) is a correct key pair
- check that $hash(pk_A, birthdayGMTts_A, r)$equals $VC_{A,age}$
- check that $todayGMTts_A - birthdayGMTts_A > intervalThreshold$

This proof will be used to call the validation contract specified by the staff, which will return the validated boolean value.

Lastly, since both the and its signature signed by the federal government are stored in Smart Contact, it could be checked that this verifiable credential is indeed issued by a specific authority. Note that the contract will generate a nullifier based on the proof that will be verified to prevent this proof from being replayed.

5.5 Threshold encrypted decentralized storage

The use of decentralized storage combined with threshold ensures that only authorized parties can access these documents under strict predetermined conditions and strict collaboration between all involved stakeholders. At no point in time is a single stakeholder able to unlock the private data of the Holder. In threshold encryption, a group of n participants collaboratively generate a public key, while the decryption key is shared among them. The public key can be used to encrypt messages directly, but decryption requires the participation of a minimum number of t participants among the n participants to obtain the correct plaintext. A cryptosystem that requires at least t participants to decrypt is called a (t, n) threshold cryptosystem.

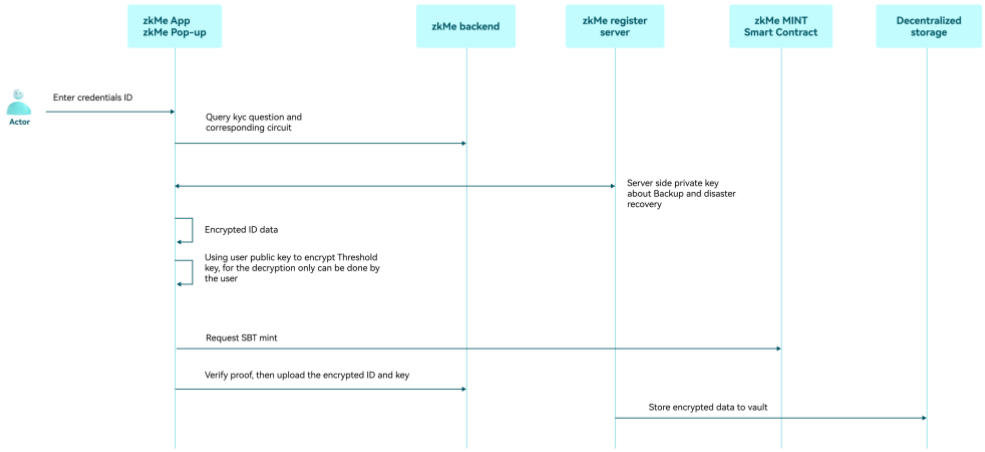

***Figure 24*. zkMe threshold encryption sequence diagram**

The zkMe protocol implements a (2, 2) threshold cryptosystem (to be expanded to 3,3 in future iterations). Here, two-party EC-ElGamal scheme: Two-party computation of ciphertexts, the global decryption key is given by: $x = x\_1 + x\_2 \bmod p$, in additive key share form. The global encryption key is given by $h = x * P$:

5.5.1 Notation

Symbol	Notion	Symbol	Notion
P	Elliptic curve base point	x	Global private key(no one knows it)(type: scalar)
p	Order of the base point	h	Global public key(type:

			ecpoint)
Z_n	Field of operations for elliptic curves	x_i	party-i 's private key (key share of)(type: scalar)
+	Addition operation in numerical terms	h_i	party-i 's public key (key share of)(type: ecpoint)
*	Multiplication operation in numerical terms	c_i	party-i 's commitment(type: scalar)
$\oplus$	Point addition operation on elliptic curves	r_i	Random number(type: scalar)
$\otimes$	Point doubling operation on elliptic curves	m	message
H	keccak256	ciphertext	ciphertext of m under AES with symmetric key
k_{point}	Point can derive the symmetric key	sym_key	symmetric key k

5.5.2 Phase 1: Global public key negotiation

The threshold encryption public key negotiation goes through the following steps.

1. Generate the keypair (x_1, h_1) for party-1 regarding h and make a commitment $c_1 = H(h_1, r_1)$ for h_1. Generate keypair (x_2, h_2) for party-2 regarding h and make a commitment $c_2 = H(h_2, r_2)$ for $h2$.

Function	**Operation**
generate_key_share(m, n) at	$x_i \xleftarrow{R} [m, n], h_i = x_i \otimes P$

party-i	
rand(p) at party-i	$r \xleftarrow{R} [1, p]$
generate_commitment(m, n) at party-i	$c = H(m \,||n)$
verify_commitment(c, m, n) at party-i	$c' = H(m \,||n)$, check $c == c'$

2. Party-1 sends c_1 to party-2.

3. Party-2 sends c_2 and the preimage (h_2, r_2) of c_2 to party-1.

4. Party-1 verifies $c_2 = H(h_2, r_2)$ and then sends the preimage (h_1, r_1) of c_1 to party-2.

5. Party-2 verifies $c_1 = H(h_1, r_1)$.

6. Party-1 and party-2 each compute $h = h_1 + h_2$, confirm that the results are the same, and jointly announce the global encryption key as h .

Function	**Operation**
compute_global_pubkey(m,n) at party-i	$h = m \oplus n$

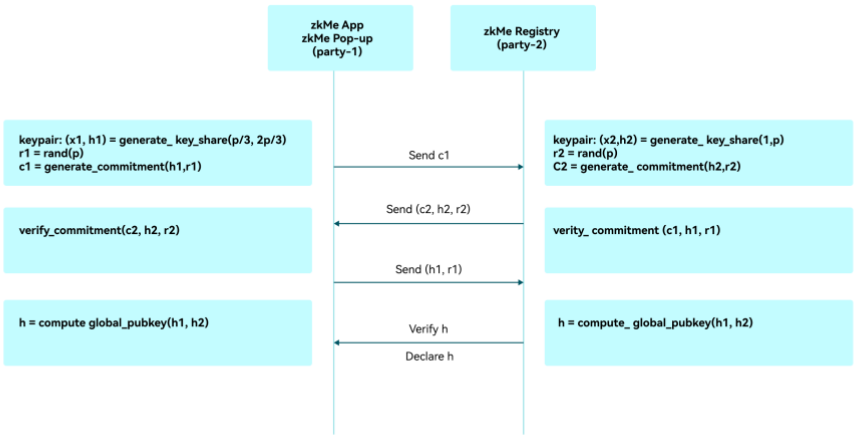

***Figure 25*. zkMe threshold encryption public key negotiation**

5.5.3 Phase 2: Encryption

The following process is standard hybrid encryption using EC-ElGamal, assuming that the encryption party has already obtained the global encryption key h through the following steps:

1. The encrypting party calls generate_sym_key(p) to generate a random k_{point}, and then calls compute_sym_key(k_{point}) to compute the symmetric key pair sym_key.

Function	**Operation**
generate_key_point(p) at party-i	$k \xleftarrow{R} [1, p], k_{point} = k \otimes P$
compute_sym_key(k_{point}) at party-i	$sym\_key = H\left(point2bytes\left(k_{point}\right)\right)$

2. The encrypting party calls the AES algorithm to encrypt the message m using the symmetric key sym_key to obtain the symmetric ciphertext enc, and then uses EC-ElGamal to encrypt by calling elgamal_encrypt(k_{point}, h) to obtain ($C1, C2$).

Function	**Operation**
elgamal_encrypt(k_{point} , h) at encrypt-party	$r \xleftarrow{R} [1, p]$, $C_1 = r \otimes P$, $C_2 = k_{point} \oplus (r \otimes h)$

3. The ciphertext (ciphertext, $C1, C2$) is made public.

5.5.4 Phase 3: Threshold decryption

In case regulators initiate bad actor proceedings, the threshold cryptography protecting the raw data of the user can be recovered using the following steps:

1. Each party-i calculates the partial decryption D_i with respect to C_1.

Function	**Operation**

compute_partial_decryption(xi, C1) at party-i	$D_i = x_i \otimes C_1$

2. Party-i sends D_i to party-3-i.

3. Party-i locally calls elgamal_decrypt(D1, D2, C2) to obtain , and then calls compute_sym_key(k_{point}) to compute the symmetric key pair sym_key.

Function	**Operation**
elgamal_decrypt(D1, D2, C2) at party-i	$D = D_1 \oplus D_2, k_{point} = C_2 \oplus (-D)$

4. Party-i calls the AES algorithm to decrypt the symmetric ciphertext enc using the symmetric key sym_key to obtain the message m.

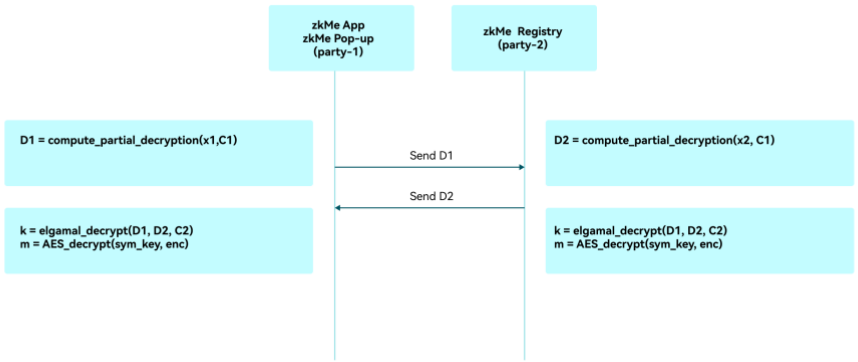

***Figure 26*. zkMe threshold decryption**

5.6 Smart contracts (SC)

In the following, a short overview of the Smart Contracts (SC) developed by zkMe for the processing of the zkMe network. Details for these SCs can be found on the zkMe website documentation. All functionalities available through the zkMe SCs are also available on zkMe APIs for non-web3 native Verifiers.

5.6.1 zkMe mint

The following sequence diagram shows the process of registering an SSI wallet, completing KYC verification, and minting an SBT token. The three participants involved are Holder, zkMe App, and Polygon (MATIC). This SC for minting SBT is deployed on the MATIC blockchain.

The process starts with the Holder requesting to register an SSI wallet through the zkMe App. Once the wallet is created, the Holder presents their credentials to the zkMe App. The zkMe App generates the relevant ZKP and triggers the minting request to the zkMe Mint Polygon smart contract.

The zkMe Mint SC receives the location pointers for the ZKP, minting an SBT asset directly to the Holder SSI wallet, The zkMe SBT, which contains the holder's DIDs, a key share, and most importantly the pointer to the verified ZKP. This process ensures that the Holder is able to securely own their Identity on-chain.

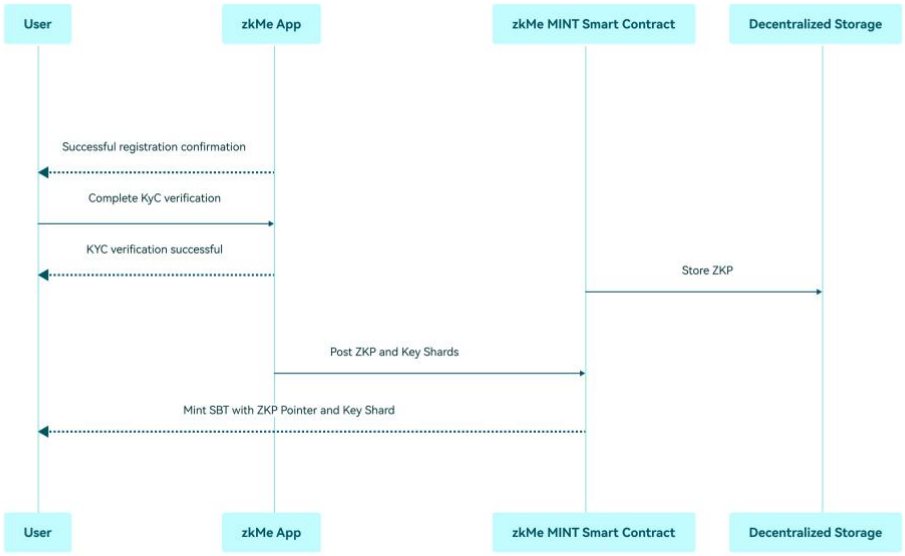

***Figure 27*. zkMe SBT mint sequence diagram**

5.6.2 zkMe delegate

The zkMe Delegate SC comes into play when a Holder wishes to perform verifications for dApps across chain ecosystems. Holders need to first connect their asset wallet to the zkMe App and sign a transaction requesting a delegate copy of SBT.

The zkMe infrastructure and zkMe Delegate SC complete the cross-chain data transfer to the Holder's connected asset wallet and issue a delegate copy of SBT. Currently, zkMe Delegate supports the ETH, MATIC and BNB chains. Support for additional EVM-compatible chains is achievable with minimal efforts.

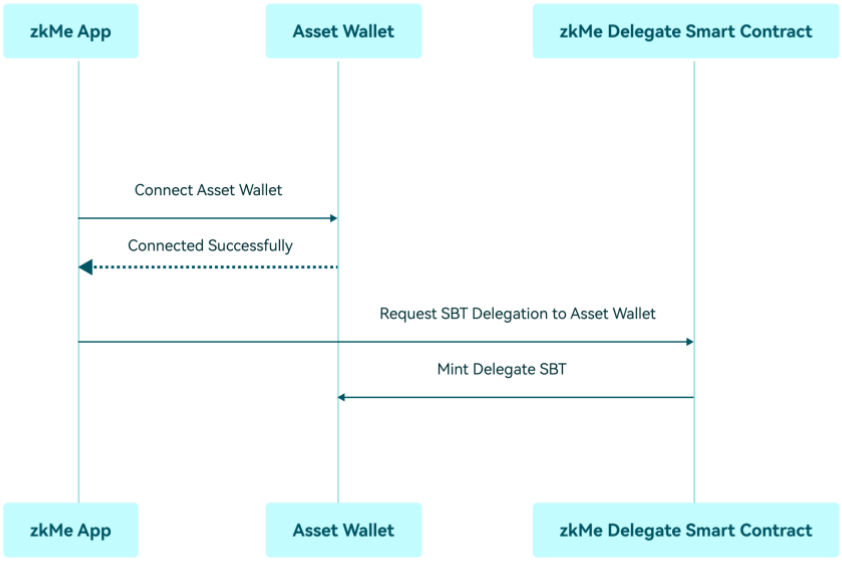

***Figure 28*. zkMe SBT delegate sequence diagram**

5.6.3 zkMe verify

The zkMe Verify SC comes into play when a Verifier wishes to perform verifications on incoming Holders' eligibility for using their services and is triggered once a dApp recognizes a SBT asset within a Holder's wallet.

The zkMe Verify SC exposes yes/no answers to predetermined eligibility questions to each of the credentials verified through zkMe. A full list of eligibility questions is provided through the zkMe documentation available through the zkMe website. Currently, zkMe Verify supports the ETH, MATIC and BNB chains. Support for additional EVM-compatible chains is achievable with minimal efforts.

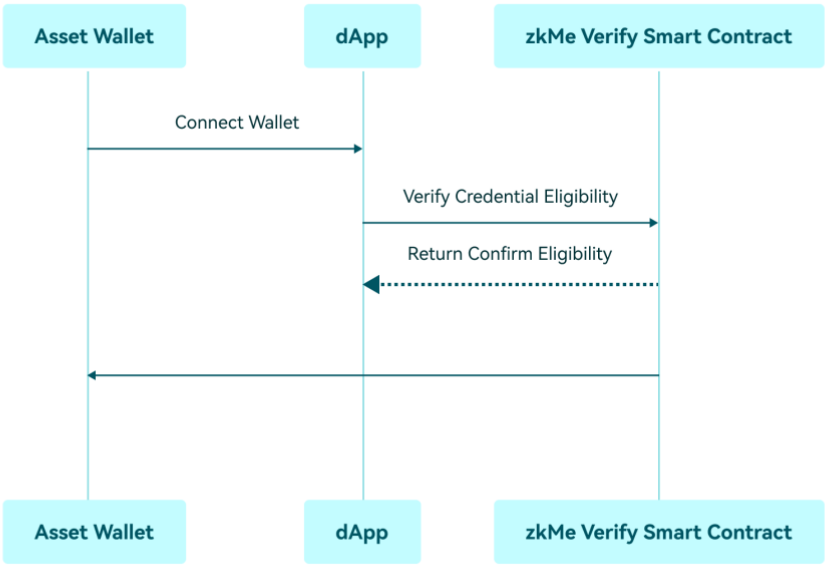

***Figure 29*. zkMe verify sequence diagram**

5.6.4 zkMe certify

The zkMe Certify SC comes into play when a Verifier needs to fulfill data storage requirements as part of their compliance with KYC/AML requirements. It is triggered by the dApp (optionally) once they verified the eligibility of the Holder through the zkMe Verify SC and requires the explicit Holder approval (through transaction signature).

The zkMe Certify SC creates a Verifier specific copy of the Holder SBT to a Verifier specified asset wallet; this SBT copy includes the Holder's private key shard, allowing the Verifier to recover the Identity of the Holder when (and only when) a regulator initiates bad-actor proceedings even without the Holders approval. Currently, zkMe Certify supports the ETH, MATIC and BNB chains. Support for additional EVM-compatible chains is achievable with minimal efforts.

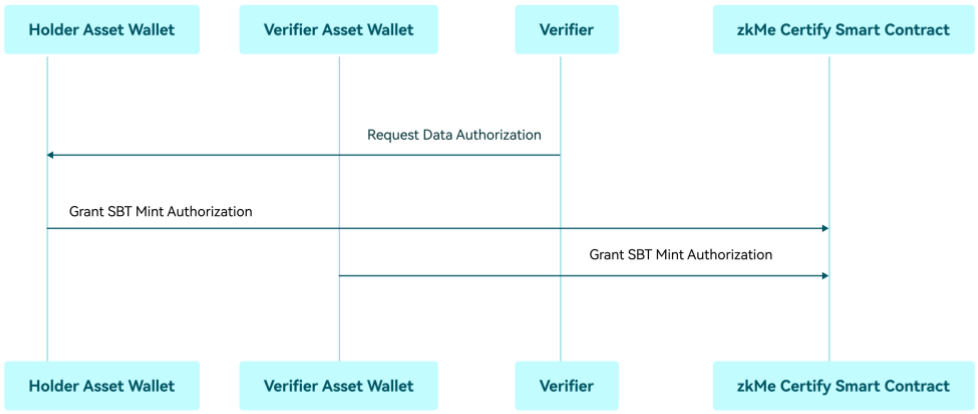

***Figure 30*. zkMe vertify sequence**

5.7 zkMe's security model

The security model constitutes a crucial component of any cryptographic protocol. In this section, we expound upon the robust security framework that underlies zkMe.

5.7.1 Notations

- **zk-SNARK**

Let $R = (c, x)$ be a polynomial relation of statements c and witness x. A zk-SNARK Π for R is composed of the following 3 polynomial time algorithms: $Setup, Prove, Verify$ and works as follows. It satisfies completeness, succinctness, computational zero knowledge and simulation extractability.

$Setup(1^{\lambda}, C) \rightarrow crs$: It takes the security parameter λ and a circuit C as input, and generates a common reference string crs .

$Prove(crs, c, x) \rightarrow \pi$: It takes a public input (statement) c and a private input (witness) x and generates and returns proof π, where $(c, x) \in R_C$ and R_C is a relation defined by C.

$Verify(crs, c, \pi) \rightarrow \{0/1\}$: This algorithm verifies proof π with regard to public input c. It returns 1 if the proof is verified successfully and 0 otherwise.

- **Completeness**: An honest prover with a valid witness can always convince an honest Verifier for a given security parameter λ and arithmetic circuit C. i.e:

$$\Pr\left[\ \mathsf{Verify}(\mathsf{crs},\pi,c)=1\ \middle|\ \begin{array}{l}\mathsf{crs}\leftarrow\mathsf{Setup}\left(1^{\lambda},C\right)\\ \pi\leftarrow\mathsf{Prove}(\mathsf{crs},c,x)\end{array}\ \right]=1\,.$$

- **Succinctness**: The size of a zk-SNARK proof π about C is short, unrelated to the complexity of C. In addition, the $Verify$ algorithm is a deterministic polynomial time algorithm, also unrelated to the complexity of C.

- **Computational Zero Knowledge**: A valid proof π is computationally zero knowledge if it does not leak any information about the witness to PPT adversary $\mathcal{A}$. More formally, for any PPT $\mathcal{A}$, a simulated algorithm $\widehat{Setup}$ can produce a common reference string crs and a trapdoor τ,which is used in $\widehat{Prove}$ to produce a simulated proof $\hat{\pi}$. This simulated proof $\hat{\pi}$ is indistinguishable from a honestly generated proof. Computational Zero Knowledge means the honestly generated proof is zero-knowledge since the simulated proof does not contain any information about the witness.

$$\Pr\left[\ \mathcal{A}(\mathsf{crs},\pi,c)=1\ \middle|\ \begin{array}{l}\mathsf{crs}\leftarrow\mathsf{Setup}\left(1^{\lambda},C\right)\\ \pi\leftarrow\mathsf{Prove}(\mathsf{crs},c,x)\end{array}\ \right]\simeq\Pr\left[\ \mathcal{A}(\widehat{\mathsf{crs}},\hat{\pi},c)=1\ \middle|\ \begin{array}{l}(\widehat{\mathsf{crs}},\tau)\leftarrow\widehat{\mathsf{Setup}}\left(1^{\lambda},C\right)\\ \hat{\pi}\leftarrow\widehat{\mathsf{Prove}}(\widehat{\mathsf{crs}},\tau,c)\end{array}\ \right]$$

- **Simulation Extractability**: The simulator can extract a witness from a proof generated by PPT adversary $\mathcal{A}$ even after $\mathcal{A}$ is allowed to use the simulation oracle and has seen many simulated proofs, which imply knowledge soundness. This property ensures that the prover knows a valid witness if he can produce a valid proof. For every PPT adversary $\mathcal{A}$, there exist simulator algorithms $\widehat{Setup}$ and $\widehat{Prove}$ a probabilistic polynomial-time witness extractor $\mathcal{X}$, such that the following probability is negligible:

$$\Pr\left[\begin{array}{l}(c,x)\notin\mathcal{R}_C\\ (c,\pi)\notin\mathcal{Q}\\ \mathsf{Verify}(\widehat{\mathsf{crs}},\pi,c)=1\end{array}\ \middle|\ \begin{array}{l}(\widehat{\mathsf{crs}},\tau)\leftarrow\widehat{\mathsf{Setup}}\left(1^{\lambda},C\right)\\ (c,\pi)\leftarrow\mathcal{A}^{\widehat{\mathsf{Prove}}}(\widehat{\mathsf{crs}},\tau,\cdot)(\widehat{\mathsf{crs}})\\ x\leftarrow\mathcal{X}\left(\mathsf{trans}_{\mathcal{A}}\right)\end{array}\right]$$

- **2-party ECDSA**

The signing private key is split into key shares: one for the Issuer, one for the Holder, and one for the Regulator (where applicable). Only the three of them working together can sign the message. The signature on the 2-ECDSA cannot be forged by any PPT adversary $\mathcal{A}$. The decryption private key is split into two key shares, one for the zk-SNARK and one for the Holder.

5.7.2 The ideal functionality $\mathcal{F}$ for zkMe

We design the zkMe network by combining all the preliminaries (see sequence diagram above) and describe the ideal functionality as follows:

- $\mathcal{F}.setup(\cdot)$

  Initialize the public parameters pp

  Decide on kyc template $\mathcal{T} := (\mathcal{Q}, c, add_{kyc})$

  Publish $(pp, \mathcal{T})$

- $\mathcal{F}.registry(pp, sk_1, sk_2, tesk_1, tesk_2) \rightarrow (pk_E, pk_{SSI}, Add_{SSI}, pk_{TE})$

  Generate key shares

  User selects random $sk_1, tesk_1$. Zkme selects random $sk_2, tesk_2$.

  Two parties runs $ComputeGlobalPubKey(sk_1, sk_2) \rightarrow pk_{SSI}$

  User runs $ComputePubKey(sk_1) \rightarrow pk_E$

  Two parties runs $ComputeGlobalPubKey(tesk_1, tesk_2) \rightarrow pk_{TE}$

  Publish $(pk_{SSI}, pk_E, pk_{TE})$

- $\mathcal{F}.mintSSISBT(\cdot)$

  Client scans the document into IDdata

  queries the kyc question set $\mathcal{C}$ and proving keys

  runs

  $$ThresholdEncrypt(pk_{TE}, IDdata) \rightarrow enc_{id}$$

  $$Encrypt(pk_E, tesk_1) \rightarrow enc_{t1}$$

  $$ProveAll(IDdata, \mathcal{C}, pk) \rightarrow \{\pi_i\}_{i \in [\mathcal{C}]}$$

  Sends $(\mathcal{C}, \{\pi_i\}_{i \in [\mathcal{C}]}, enc_{id}, enc_{t1})$ to server

- $\mathcal{F}.transferSBT(\cdot)$

  Client downloads enc_{t1} from SBT_{ssi}, and runs

  $$Decrypt(sk_1, enc_{t1}) \rightarrow tesk_1$$

$$ComputeCapsule(sk_{assert}, tesk_1) \rightarrow capsule$$

Sends $capsule$ to server

Server calls $mint(enc_{\mathcal{C}}, enc_{\pi}, capsule) \rightarrow SBT_{assert}$

- $\mathcal{F}.showSBT(\cdot)$

Client downloads $capsule$ from SBT_{asset}, and runs

$$ComputeRK(sk_{asset}, pk_{project}) \rightarrow rk$$

Sends rk to server.

Server calls $ReEncrypt(SBT_{asset}, rk, targetProject) \rightarrow rk$

- $\mathcal{F}.verifyShow(\cdot)$

DApp watches the event and gets rk , and runs

$$PDecrypt(sk_{project}, rk) \rightarrow tesk_1$$

$$ThresholdDecrypt(tesk_1, enc_{\mathcal{C}}, enc_{\pi}) \rightarrow (\{ques_i\}_{i\in[\mathcal{C}]}, \{\pi_i\}_{i\in[\mathcal{C}]})$$

If $VerifyAll(\mathcal{C}, \{\pi_i\}_{i\in[\mathcal{C}]}) == 1$, authorization checking

5.7.8 Security Proof

As quickly highlighted in chapter 4.4, the zkMe protocol shall be deemed secure if the following three theorems are proven:

- **Theorem 1:**

If the zk-SNARK scheme Π satisfies Completeness and Computational Zero Knowledge, the 2-party threshold encryption scheme is secure against chosen-plaintext attacks, then the proposed zk-SNARK provides zk-SNARK.

- **Proof of Theorem 1:**

We adopt the standard hybrid argument to prove Theorem 1 by showing that our zkMe network securely realizes the ideal functionality $\mathcal{F}.mintSSISBT(\cdot)$. Let $\mathcal{R}eal$ be a random variable representing the joint view of zkme server and nodes in blockchain. If there exists a PPT simulator $\mathcal{S}im$ such that the output of $\mathcal{S}im$ is computationally indistinguishable from the output of $\mathcal{R}eal$.

Hybrid_0: Hybrid_0 is a real view where PPT adversary $\mathcal{A}$ tries to find the identity information of wid.

Hybrid_1: Hybrid_1 is identical to Hybrid_0, except that the simulated wid runs $ThresholdEncrypt(pk_{TE}, r) \rightarrow \mathbb{P}$ where r is uniformly random. Since only the content of enc_{id} is changed, the ThresholdEncrypt function is secure against chose-plaintext attacks, Hybrid_1 is indistinguishable from Hybrid_0.

Hybrid_2: Hybrid_2 is identical to Hybrid_1, except that the simulated wid runs $ProveAll(IDdata', \mathcal{B}, pk) \rightarrow \{\pi_i'\}_{i\in[\mathcal{B}]}$ where$\mathcal{B}$ has the same amount of questions as $\mathcal{C}$ and the distribution of $\mathcal{B}$ is uniformly random and every π_i' is correctly generated against the question in $\mathcal{B}$. Since only the content of $\mathcal{C}$ and $\{\pi_i\}_{i\in[\mathcal{C}]}$ is changed, the zk-SNARK satisfies Completeness and Computational Zero Knowledge, all $\{\pi_i'\}_{i\in[\mathcal{B}]}$ are verified valid, Hybrid_2 is indistinguishable from Hybrid_1.

Hybrid_3: Hybrid_3 is identical to Hybrid_2, except that the simulated zkme server runs $ThresholdEncryptAll(pk_{TE}, \{ques_i\}_{i\in[\mathcal{B}]}) \rightarrow enc_{\mathcal{B}}$. Since only the content of $\mathcal{C}$ and $enc_{\mathcal{C}}$ is changed, the ThresholdEncrypt function is secure against chose-plaintext attacks, Hybrid_3 is indistinguishable from Hybrid_2.

Hybrid_4: Hybrid_3 is identical to Hybrid_3, except that the simulated zkme server runs $ThresholdEncryptAll(pk_{TE}, \{\pi_i'\}_{i\in[\mathcal{B}]}) \rightarrow enc_{\pi}'$, where $\{\pi_i'\}_{i\in[\mathcal{B}]}$ are independent with $\{\pi_i\}_{i\in[\mathcal{C}]}$. Since only the content of enc_{π} is changed, the ThresholdEncrypt function is secure against chose-plaintext attacks, Hybrid_4 is indistinguishable from Hybrid_3.

The argument proves that there is a simulator $\mathcal{S}im$ sampled from the distribution described above so that its output is computationally indistinguishable from the output of $\mathcal{R}eal$. Hence, our scheme can provide anonymity.

- **Theorem 2:**

If the zkMe client provides a trusted execution environment, the public blockchain is immutable, the commitment scheme is binding, then the proposed zk-SNARK provides unforgeability.

- **Proof of Theorem 2:**

We also adopt the standard hybrid argument to prove Theorem 2, where the probability that PPT adversary $\mathcal{A}$ succeeds in breaking unforgeability becomes negligible.

Hybrid_0: Hybrid_0 is a real view where PPT adversary $\mathcal{A}$ tries to mint SBT with fake document and transfer others' SBT_ssi to his/her asset wallet.

Hybrid_1: Hybrid_1 is identical to Hybrid_0, except that $\mathcal{A}$ constructs the fake document and shows it to zkme client. Since the zkme client provides a trusted execution environment, any forged document can not pass validation. Hybrid_1 is indistinguishable from Hybrid_0.

Hybrid_2: Hybrid_2 is identical to Hybrid_1, except that $\mathcal{A}$ constructs the ciphertext and proof $(enc_{\mathcal{C}'}, enc_{\pi'})$ based on fake question set and modifies the data in SSI blockchain. Since the public blockchain is immutable. $\mathcal{A}$ cannot change the verified ciphertext data $(enc_{\mathcal{C}}, enc_{\pi})$ of SBT_{ssi} in SSI blockchain. Hybrid_2 is indistinguishable from Hybrid_1.

Hybrid_3: Hybrid_3 is identical to Hybrid_2, except that colluder constructs the $capsule'$ based on $sk_{\mathcal{A}}$ and tries to transfer his/her SBT to asset wallet $\mathcal{A}$. Since the commitment scheme is binding. It can be detected by mint server if the private key of asset wallet is not bound to public key of SSI wallet. The asset wallet of $\mathcal{A}$ can not receive the SBT from other ones. Hybrid_3 is indistinguishable from Hybrid_2.

Hence, under these assumptions of Theorem 2, the proposed system provides the unforgeability property.

- **Theorem 3:**

If the zkMe client provides a trusted execution environment, the (k, n)-threshold SSS is correct, the decentralized storage is immutable, then the designed zkMe network provides traceability.

- **Proof of Theorem 3**

We also adopt the standard hybrid argument to prove Theorem 3, where the probability that PPT adversary $\mathcal{A}$ succeeds in breaking traceability becomes negligible.

Hybrid_0: Hybrid_0 is a real view where PPT adversary $\mathcal{A}$ tries to stop the regulator committee to trace back the real IDdata.

Hybrid_1: Hybrid_1 is identical to Hybrid_0, except that $\mathcal{A}$ constructs the fake ciphertext $enc_{id'}$ where plaintext is fake identity data $IDdata'$. Since the zkme client provides a trusted execution environment, any forged document can not pass validation. Hybrid_1 is indistinguishable from Hybrid_0.

Hybrid_2: Hybrid_2 is identical to Hybrid_1, except that $\mathcal{A}$ generates the ciphertext $enc_{id'}$ and tries to replace the data enc_{id} in IPFS. Since the decentralized storage is immutable. $\mathcal{A}$ cannot change the data enc_{id} in IPFS. Hybrid_2 is indistinguishable from Hybrid_1.

Hybrid_3: Hybrid_3 is identical to Hybrid_2, except that $\mathcal{A}$ tries to break the correctness property of the (k, n)-threshold SSS.. Since the (k, n)-threshold SSS is correct. Hybrid_3 is indistinguishable from Hybrid_2.

Hence, under these assumptions of Theorem 3, the proposed system provides the traceability property.

6. A case study on the application of zkMe for web3 KYC

Web3 has emerged as a critical technology that has the potential to revolutionize several industries. As the web3 ecosystem continues to evolve, the regulatory landscape surrounding it is still developing, and utilizing traditional KYC solutions can run counter to the core principles of web3. To address this issue, this section presents a case study on the application of zkMe in KYC within the web3 ecosystem.

The use of zkMe in KYC can help ensure compliance with regulatory requirements while also maintaining the privacy and security of user information, which is a fundamental principle of the web3 ethos. By leveraging the advanced cryptographic techniques of zkMe, it is possible to establish a trustless system that enables secure and efficient identity verification without requiring users to divulge their sensitive personal information. The case study presented in this section offers valuable insights into the potential of zkMe in addressing the unique challenges posed by KYC within web3.

6.1 Success criteria for web3 compatible KYC

The core spirits of web3 are decentralization and data autonomy, which can make the implementation of traditional KYC processes challenging, as they often require the collection and storage of personal data, which goes against the core principles of web3. However, ZKPs-based KYC offers a solution to this challenge, providing a way to verify users' identities while still maintaining data autonomy and decentralization.

Here are some of the key business requirements for implementing ZKPs-based KYC in the web3 ecosystem:

- **Privacy:** With ZKPs-based KYC, businesses can verify users' identities without requiring them to disclose their personal information. This can help to protect users' privacy, as their data is not stored on a centralized server or shared with third parties.

- **Regulatory Compliance:** Many businesses operating in the web3 ecosystem are subject to regulatory requirements, such as anti-money laundering (AML) and know-your-customer (KYC) regulations, incl. Identity recovery capabilities for at least five years after the completion of a service relationship given reasonable suspicion and regulatory intervention, and the need for travel rule of KYC data among financial institutions. ZKPs-based KYC can help businesses comply with these regulations while still maintaining the decentralized and autonomous nature of the web3 ecosystem.

- **Security:** By implementing ZKPs-based KYC, businesses can enhance security and reduce the risk of fraud, identity theft, and other malicious activities. The use of ZKPs allows for secure identity verification without the need for centralized identity repositories, which can be a target for attackers.

- **Efficiency:** Traditional KYC processes can be time-consuming and expensive, which can create a barrier to entry for some businesses. ZKPs-based KYC can improve efficiency by reducing the time and cost associated with verifying user identities.

- **User Experience**: With ZKPs-based KYC, users can enjoy a more seamless and user-friendly experience when accessing web3 applications and services. The process of identity verification is simplified, reducing the friction that can sometimes exist with traditional KYC processes.

Referring to Chapter *3.3*, the typical process of existing eKYC, we how zkMe can fulfill these criteria and provide a superior KYC solution for web3.

6.2 zkMe's zkKYC solution

As discussed in the above sections, KYC is a crucial step for businesses operating in the web3 ecosystem to ensure compliance with regulatory requirements and prevent illicit activities. It is an important way for web3 mass adoption to occur. This section discusses how zkMe evolves the Roles of SSI verifications and the eKYC process to address aforementioned success criteria.

6.2.1 The SSI roles evolved

As an evolution of the roles needed for SSI (Chapter 2.5), zkMe defines Roles with new interactions.

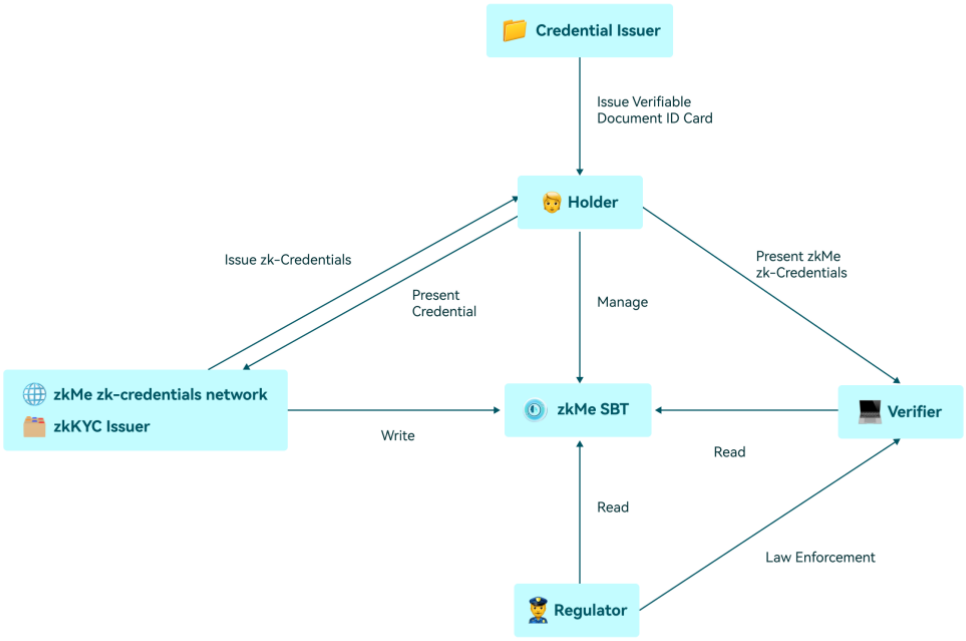

***Figure 31*. zkMe zkKYC role concept**

- **Issuer:** In contrast to the traditional SSI role concept, zkMe splits the Issuer role into a trusted Issuer to Issue the verified credentials and the verified presentations.
    - **Credential Issuer:** Refers to governmental or financial entities or organizations that issue physical or digital credentials (such as Passports or ID cards) to individual Holders. This role is equivalent to the role of the Issuer within the traditional SSI concept.

- **ZKP Issuer:** zkKYC specific concept that refers not to a person or entity, but instead to a trusted Issuer program running directly on the Holder's end device (through the use of Trusted Cryptographic Setups and open-sourced and audited algorithms) uses ZKP technology to process the credentials provided by the Holder and generate VPs in the form of ZKP. zkMe enables eligibility proofs, which are ZKPs that the Holder meets the criteria set out by the Verifier to provide access to the requested service. By leveraging the information in VC, eligibility proofs allow for authentication without disclosing the actual information itself to anyone. For example, a proof can demonstrate that the Holder is of a certain age, is a domestic resident, not on a sanctions list, or not a politically exposed person. This innovative approach to identity verification not only improves privacy and security, but also increases efficiency and convenience for businesses and users alike.

  The security of a cryptographic protocol is of paramount importance, especially in the case of zkMe network which is based on zero-knowledge proofs (ZKPs). A robust security model is essential to ensure the protocol's resistance against potential attacks. This chapter presents the ideal functionality for zkMe along with its security goals and security proofs.

- **Holder:** This role remains mostly unchanged to the one proposed in the SSI concept. Refers to individuals that hold VC that can be used for various purposes such as accessing services, proving identity, or providing proof of qualifications or certifications. Holder can use their VC to access various services without the need for repeated identity verification. In zkKYC processes, a Holder can trust that their credentials are proven to a Verifier without disclosing private details.

- **Verifier:** Verifiers in zkKYC check the authenticity and correctness of a VP claim without the need for the Holder of the credential to reveal sensitive personal information. The Verifier checks the proof against a set of rules or criteria, such as checking that the proof is cryptographically secure and that it matches the information stored on the blockchain. If the proof is valid, the Verifier can be confident that the information provided by the holder of the credential is accurate without having knowledge of the underlying information itself. The Verifier can check the proof against the set of rules or criteria, such as checking that the user is of legal age or is a resident of a particular jurisdiction, without actually processing the Holder's personal information

- **Regulator:** The goal of the regulator role in a zkKYC process remains unchanged; materially, it is, however, given a direct role in keeping the data of the Holder private. As the Regulator holds one of three key shards required to uncover a Holder's identity, none of the stakeholders involved (incl. the regulator itself) can remove the veil of Holder anonymity on their own.

- **zkMe SBT:** In zkKYC, the verifiable data registry is replaced with an SBT asset stored on public distributed ledgers, that point towards the decentralized storage that contains the ZKP. In contrast to typical SSI implementations, only anonymized VP claims are stored, claims are explicitly designed to not allow for indirect Holder identification, and are only accessible to authorized stakeholders. The zkMe SBT token revolutionizes the way we handle identity and credentials in the web3 ecosystem. The zkMe SBT is the on-chain representation of a Holder's Identity. It contains their DID, ZKP, and one of three key shards used to encrypt and protect the raw data of the holders.

6.2.2 The KYC process restructured

zkMe zkKYC enables users to prove their identity to a service provider without revealing their personal information, improving privacy and security over existing eKYC solutions. The process can also help service providers comply with regulatory requirements for KYC while reducing the risk of data breaches, identity theft and verification costs in general. The restructured process of zkKYC involves the following steps:

1. **Credential Verification**: The Holder submits their identity documentation digitally to the zkKYC Issuer for verification. This step involves the traditional process of providing personal information and documents, such as a passport or driver's license. The Holder's Identity documentation and likeness is verified through OCR and Facial Recognition checks. The zkKYC Issuer algorithm is able to parse the machine-readable identity documents in a structured way. No need for any human interaction or third-party processing.

2. **Screening & Risk Assessment:** The Holder Identity is screened against lists of known criminals, terrorists, or politically exposed persons (PEPs), transaction history and other relevant information to identify potential risks. This check is processed in real time, no personal data is stored at any time. On basis of the check the zkKYC Issuer generates a risk profile for the Holder Identity and actively scrubs all private user data from memory.

3. **ZKP Generation:** Once the zkKYC Issuer has verified the holder's identity, it issues an anonymous VP claim (in form of SBT and ZKPs) for each of the preselected eligibility questions. ZKPs provide a mechanism to express traditional credentials digitally, cryptographically secure, privacy-respecting, and machine-verifiable. SBTs are stored on-chain and ZKPs are stored in decentralized storage.

4. **SBT Mint:** Creation of an encrypted data object to the Holder's SSI wallet that contains their DID and respective ZKP pointers required to prove a Holder's eligibility to Verifiers repeatedly.

5. **Proof Verification:** When a Holder wants to access a service that requires KYC, they receive a request to allow for verification of proofs from the Verifier. Once authorized, the Verifier checks the Holder's ZKP against their internal eligibility criteria, such as age or residency. If the proof is valid and the ZKP answers fulfill the service requirements, the user is granted access to the service.

6. **Proof Revocation:** ZKP VP claims have a natural If the user's verifiable credential is compromised or revoked, the identity issuer can update or revoke the credential, preventing its use for future authentication and verification.

7. **Ongoing Monitoring:** Verifiers may process continuous on-chain transaction monitoring to ensure compliance with relevant regulations and to detect any suspicious activity that may indicate fraudulent behavior. Additionally, every time a ZKP is reissued upon expiration or revocation, screening and risk assessment procedures are repeated.

8. **(Data Recovery)**: In case, and only in case, the regulator initiates formal bad actor proceedings against a Holder. Upon substantial suspicion, the Regulator, Credential Issuer and Verifier combine their key shards, creating the private key required to unlock the original identity document proof stored in threshold encrypted decentralized storage.

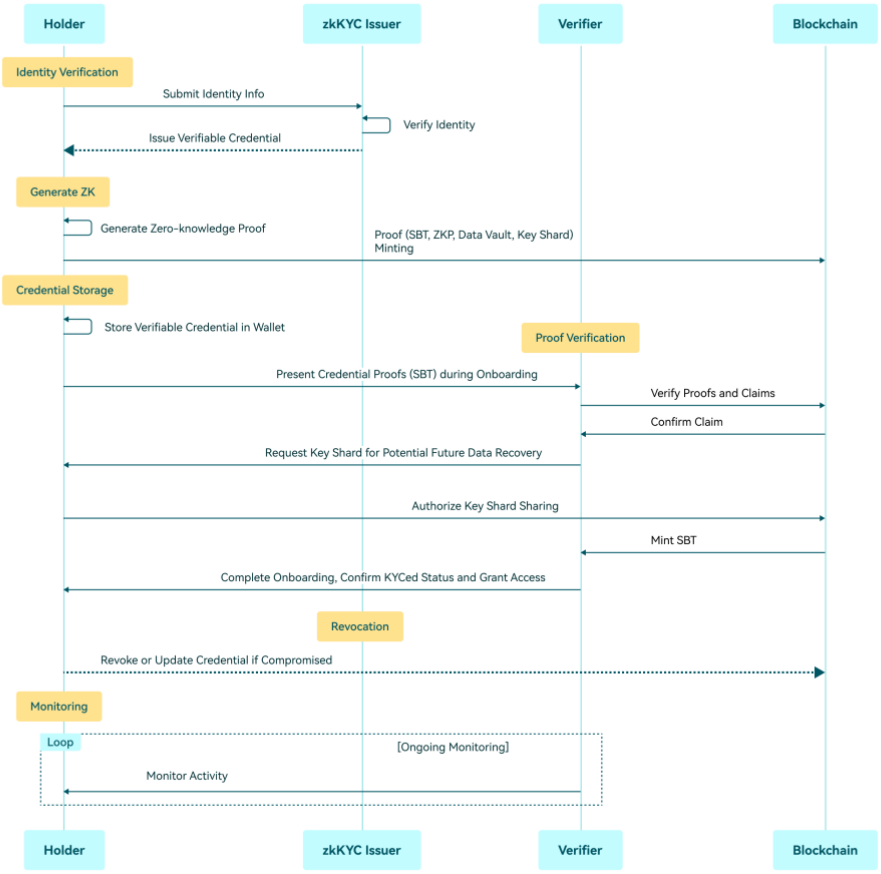

***Figure 32*. zkMe's zkKYC high level sequence diagram**

6.2.3 The zkKYC benefits

zkMe's zkKYC is a privacy-enhancing approach that allows businesses to verify the identity of their customers without collecting and storing sensitive personal information. This approach offers several benefits, including:

- For Holders:
  - Ultimate privacy protection
  - Ultimate Identity theft protection
  - Reusable verifications
  - Cross-chain verifications
- For Verifiers:
  - Increased solution trust without the loss of regulatory compliance

- Ultimate protection from user data breaches
- Increased user conversion through lowered user verification barriers
- Fully decentralized user verification (dApp compatibility)
- Reduced user verification costs
  - Marginal costs of single verification is zero
  - No centralized data storage requirements
  - No human process costs for verification

- For Regulators:
  - Creation of new, regulated financial markets in web3
- For Credential Issuers:
  - Removal of trust-intermediaries when issuing VCs and VPs directly to the zkMe network

Overall, zero-knowledge KYC offers a more privacy-preserving and efficient way to verify customer identities, benefiting all stakeholders.

7. Additional zkMe use cases

The applications of zkMe are not limited to the traditional KYC use case described above. Extending beyond traditional identity verification to include educational achievements, medical records, credit scores, and even social media status, zkMe can have significant impact in various domains, including identity and credential verification, permissioned DeFi, DeFi credit loans, and DAO management. This chapter discusses a non-extensive selection of potential use cases of applying zkMe in various domains.

7.1 zkMe for permissioned DeFi

Permissioned DeFi refers to a type of decentralized financial where all participants are KYC/KYB verified and AML compliant. Before participating in DeFi protocols, participants must undergo whitelisting procedures and background checks on their identity through KYC and KYB verification. This allows regulated institutions to access DeFi easily and securely, providing assurance that all participants have undergone a background check.

Permissioned DeFi is seen as a nascent innovation of DeFi and is designed to address compliance and regulatory concerns that have prevented regulated institutions from accessing traditional DeFi protocols built on public blockchain networks.

In the context of Permissioned DeFi, zkMe can provide a secure and privacy-preserving platform for credential issuance and verification, enabling users to participate in DeFi activities without compromising on their privacy while allowing regulated financial institutions to benefit from For example, zkMe can be used to verify the credit scores of borrowers on DeFi credit loans, enabling lenders to make informed decisions without accessing sensitive information about the borrower.

By using zkMe, Permissioned DeFi networks can ensure that participants are authorized to access the network and its services without revealing any sensitive information to unauthorized parties. Without a decentralized, zero-knowledge KYC solution, Permissioned DeFi markets (Pools) have to be segmented for each single centralized KYB provider as their access control mechanisms to the pools do not exist on-chain. This also means that the underlying DeFi assets have to be artificially wrapped by a centralized entity, mooting the original ethos of Decentralized Finance and forcing Permissioned DeFi providers to create new security products rather than just providing access to decentralized commodities markets.

Such segmented, so-called, Permissioned DeFi markets are highly capital inefficient. True permissioned DeFi is only possible when using a truly end-to-end zero-knowledge, decentralized Credential Network such as zkMe.

7.2 zkMe for undercollaterized crypto lending

DeFi credit loan applications involve borrowers providing collateral or other forms of guarantees to secure loans from lenders on decentralized finance platforms. In such applications, it is important to protect the privacy and security of the borrowers' sensitive information, such as their credit history, income, and other financial data.

By using zkMe, DeFi credit loan applications can ensure that borrowers' sensitive information remains private and secure while still allowing lenders to assess the borrowers' creditworthiness. This can be achieved by allowing borrowers to prove their creditworthiness without revealing any sensitive information to the lenders. For example, a borrower could provide a zk-credential that proves their credit score or income level,

without revealing the actual credit score or income level to the lender. The lender can then use this credential to assess the borrower's creditworthiness and make a loan decision, without accessing or storing the borrower's sensitive information.

In DeFi credit loans, collateral is typically required to secure the loan. ZKPs-based credential networks like zkMe can enable users to prove ownership of collateral without revealing sensitive information about the underlying asset. In addition to enhancing the privacy and security of borrowers' sensitive information, zkMe can also improve the efficiency and speed of DeFi credit loan applications. This is because the verification of credentials can be done automatically and quickly, without requiring manual review or input from lenders or other parties.

Overall, zkMe can be a valuable tool for improving the privacy, security, and efficiency of DeFi credit loan applications, making them more accessible to borrowers and lenders while ensuring compliance with data privacy and security regulations.

7.3 zkMe for loyalty programs

In a ecommerce loyalty programs, customers earn rewards or points for making purchases or engaging with a particular brand or business. These rewards are often linked to personal information, such as the customer's name, email address, or purchase history. ZKPs-based credential networks like zkMe can provide several features that could be applied to loyalty programs to enhance their security, privacy, and efficiency.

- **Private user data management**: Loyalty programs often require users to provide personal information, which can be vulnerable to hacking and data breaches. zkMe can provide more secure and private methods for managing user data, allowing users to control their own data and protect it from unauthorized access.

- **Secure and private transactions**: Loyalty programs often involve transactions between users and merchants, which can be vulnerable to fraud and manipulation. zkMe can provide secure and private transaction processing, enabling users to redeem rewards without revealing sensitive information about their identity or transaction history.

- **Cross-platform interoperability**: Loyalty programs often operate across different platforms and merchants, which can create challenges for interoperability. zkMe can provide efficient cross-platform transactions, enabling users to transfer rewards and redeem them across different loyalty programs.

- **Secure and anonymized market analytics**: Loyalty programs often collect user data for analytics and marketing purposes, which can raise privacy concerns. zkMe can provide more secure and private methods for data analytics, allowing users to control their own data and protect it from unauthorized access.

Overall, the features of zkMe could provide significant benefits to loyalty programs by enhancing their security, privacy, and efficiency. By integrating these features into loyalty program systems, it will be possible to create more secure, private, and user-centric loyalty programs that better meet the needs of users and merchants.

7.4 zkMe for decentralized social networks

In a decentralized social networks, users share information and interact with each other without relying on a central authority or platform. Privacy and security are important considerations in such networks, as users may want to control the information they share and with whom they share it.

By applying zkMe, decentralized social networks can **enhance the privacy and security** of their users' sensitive information **while increasing and trust in the information shared** (and overall data quality). This can be achieved by allowing users to prove certain information about themselves, such as their identity or interests, without revealing any sensitive information to other users or the network.

For example, a user could provide a ZKPs-based credential that proves their age or interests, without revealing their actual birthdate or specific interests to other users or the network. Other users can then verify the credential and determine if they want to connect or interact with the user, without accessing or storing the user's sensitive information.

In addition, social networks often rely on centralized content moderation, which can be vulnerable to bias and censorship. zkMe can provide decentralized methods for content moderation, allowing users to control the content they see and share without relying on centralized authorities.

7.5 zkMe for DAO managment

Decentralized Autonomous Organizations (DAOs) are self-governed organizations that use blockchain technology to enable decentralized decision-making and management. DAOs are typically open, transparent, and trustless, meaning that members can participate in decision-

making without relying on centralized intermediaries. However, DAOs also face challenges related to privacy, security, and identity verification.

ZKPs-based credential networks, such as zkMe, have the potential to address some of these challenges by enabling secure, privacy-preserving, and verifiable identity and credential management. By using a ZKPs-based credential network, DAOs could manage member identities and credentials without relying on centralized authorities, while still ensuring that members are verified and authenticated in a secure and private manner.

Here are a few examples:

- **Secure and private voting:** DAOs often rely on voting to make decisions, which can be vulnerable to fraud and manipulation. zkMe can provide secure and private methods for voting, enabling participants to cast their votes anonymously without revealing sensitive information.

- **Identity management:** DAOs often require participants to prove their identity to participate in decision-making. zkMe can provide decentralized methods for managing participant identity, enabling secure and private authentication and access control.

Overall, by integrating these features into DAO management systems, it may be possible to create more secure, transparent, and efficient decision-making processes for decentralized organizations.

8. Future work

As with all new product introductions, there is room for improvement in the current architecture of zkMe. These are areas that need to be optimized in future work. In this chapter we highlight two important Roadmap items for the zkMe network, outlining rough plans for how it plans to tackle these improvement areas.

A continuously updated protocol roadmap is provided through the website: www.zk.me.

8.1 zkMe MPC-based identity oracle

Many credentials exist only in form of website data. Such credentials are needed on-chain for many decentralized applications. Accessing website data that is locked behind HTTPS/TLS encryption in a decentralized and bridging them on-chain in a trustless manner is a significant challenge.

In this regard, we propose an identity oracle solution based on the DECO framework, as described in the paper "*DECO: Liberating Web Data Using Decentralized Oracles for TLS*" publis*hed at the* 2018 USENIX Security Symposium.

This proposed zkMe web data oracle to blockchain architecture is to use the DECO method to provide secure and decentralized access to web data. By using a "three-party TLS handshake" and a node operator network, the system can ensure that the data access is secure and trustworthy. Additionally, the use of smart contracts ensures that the data access rules are enforced in a transparent and automated manner. An overview of the most relevant elements is provided here:

1. **zkMe ID SC:** A smart contract to manage the relationship between the data provider and the data consumer, as well as enforcing access rules (fees, formats, frequencies, node operator incentives, among other). It allows data providers to register their data for on-chain consumption.

2. **Oracle nodes:** Implement a set of decentralized oracle nodes that collect and verify TLS certificates from the data providers and transmit them to the clients. The decentralized oracles can be incentivized with tokens or other rewards for their participation.

3. **Client-side library:** Client-side library that will verify the TLS certificates and ensure secure access to the web data. The library should be integrated with the client's blockchain wallet to enable seamless access to the data. When a consumer requests access to a data provider's data, the zkMe ID SC will notify the decentralized oracles to collect the TLS certificate from the data provider and transmit it to the client. The client-side library will verify the TLS certificate and allow secure access to the web data.

Symbol	**Notion**	**Symbol**	**Notion**
P	Prover (individuals)	enc_key	The key of the encrypted data in TLS
V	Verifier (Business)	mac_key	The MAC key in TLS
S	TLS server	t	Digest

Q	P-initiated query	Q_s	The private parameters of Q to tls server
Q'	Encrypted Q	k_p	The key to Prover
R	Data replied by S	k_v	The key to Verifier
R'	Encrypted R	k	The tls session keys

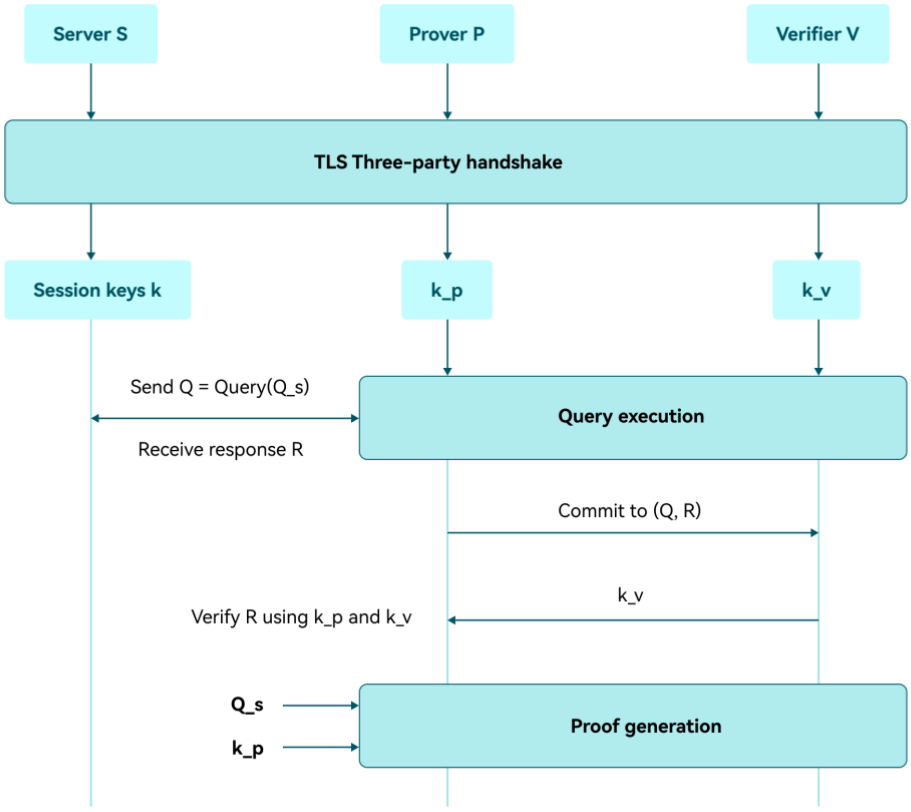

***Figure 33*. zkMe Identity oracle concept**

8.2 zkMe SBT zk-Bridge

In the current architecture of zkMe, holders of zkMe SBT must authorize the replication of their zkMe SBT on the Verifier's chain. The current architecture incurs significant transmission costs for delegating SBT across different ecosystems and is dependent on a central service for bridging between chains.

zkMe is working to establish asset specific zk-Bridge, as efficient tools to bridge SBT across chains. Ensuring strong security without relying on external trust assumptions. Through concise zero-knowledge proofs, zkMe Bridge guarantees correctness and significantly reduces on-chain validation costs. With a modular design, zkMe's zk-Bridge may in future support a wide range of use cases and functions, including messaging, token transfer, and computational logic for manipulating state changes from different chains. Apart from the security of the underlying blockchain, we do not require additional security requirements. Using specially constructed zk-SNARKs, proof verification requirements are kept to a minimum.

Overall, zkMe zk-Bridge are a powerful solution to address the challenges of cross-chain transmission while maintaining high security and efficiency.

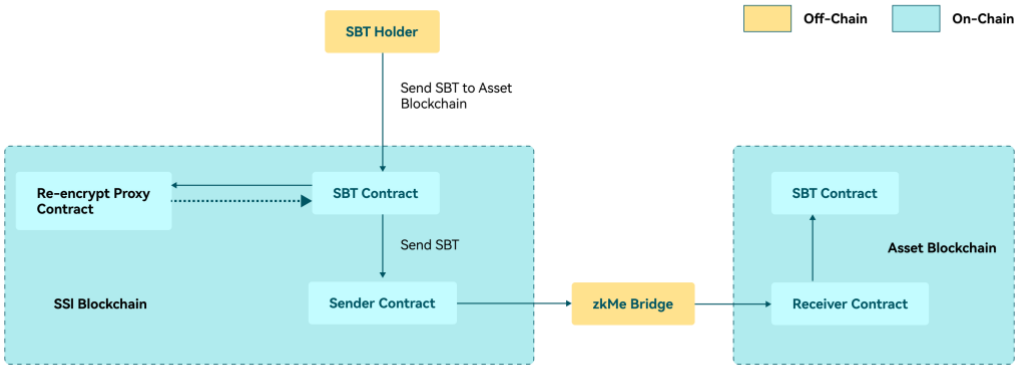

Figure 34. **zkMe zk-Bridge concept**

9. Conclusion

In conclusion, this chapter underscores the benefits of ZKP-based verifications, which is a key feature of the zkMe network. Moreover, it identifies the drivers that will fuel the adoption of the platform. Lastly, a call to action is made to encourage the use of zkMe for secure, privacy-preserving verifications.

9.1 Benefits of ZKP-based verifications

This section explores the potential impact of the ZKP-based verifications on the web3 ecosystem and beyond; delving into benefits and challenges.

One of the main advantages of ZKP-based verifications is its **privacy-preserving** nature. Credential holders can prove their identity or possession of credentials without revealing any sensitive information. This is achieved through the use of cryptographic protocols that allow for the verification of information without revealing its content.

Another benefit of zkMe network is its **decentralized** nature. With zkMe network, users can manage their identities and credentials without relying on any centralized authority. This can provide greater control over personal data and reduce the risk of data breaches and identity theft. Moreover, the zkMe network can enable the creation of new types of digital identities that are not tied to any specific platform or service, but rather to the user themselves.

Holders have **greater control** over their personal data, **reduce the risk of identity theft**, and enable the creation of new types of digital identities. This can pave the way for **new use cases and applications**, in decentralized finance, decentralized social networks, and more.

Moreover, ZKP-based verifications can also provide benefits beyond the web3 ecosystem. For instance, they can be used in traditional industries such as healthcare, where privacy and security are critical. ZKP-based verifications can enable patients to share their medical records with healthcare providers without revealing any sensitive information, thus improving the quality of care while preserving privacy.

Moving forward, it will be important for developers, researchers, and industry leaders to work together to address the challenges and build upon the potential of ZKP-based verifications. This can include developing new standards and protocols for interoperability, improving the scalability of technology, and creating user-friendly interfaces that make it easy for non-experts to use and understand.

9.2 Adoption drivers

Increasing adoption of ZKP-based verifications such as the ones provided by zkMe will not be straightforward and there will be several challenges that will need to be addressed. This section explores strategies for increasing its adoption.

Improving Usability: One of the main barriers to adoption of ZKP-based verifications is usability. The underlying cryptographic protocols can be complex and difficult to understand for non-experts. To increase adoption, it is crucial to improve the usability of the ZKP-based verifications. This can include developing user-friendly interfaces, providing

clear documentation, tutorials, and offering support for developers who are implementing the technology.

Developing Interoperability Standards: As zkMe network are still relatively new, there are no widely accepted standards or protocols for interoperability with existing systems. To increase adoption, it is essential to develop interoperability standards that enable ZKP-based verifications to work seamlessly with existing identity solutions, be it DID or FID in either web2 or web3. Collaboration among regulators (global eID solutions), industry leaders, and standards organizations (such as the W3C) to develop and promote interoperability standards.

Building Community Support: Building community awareness and support for truly anonymous verifications is a critical factor for widespread adoption. End-to-end zero-knowledge identitiy solutions are more difficult to build than traditional FID solutions; adoption will thus depend on the broader market recognizing the value of decentralized, trustless and open verifications. To build communities, it is essential to engage with broader developer and user communities. Higher degrees of collaboration, knowledge & best practices sharing, and support offered, foster opportunities for networking and collaboration, and encourage the development of additional use cases. The early adopter community will be the foundation for a self-reinforcing cycle for additional developers, users, and other stakeholders to adopt ZKP-based verifications as the new paradigm for digital verifications.

Creating Sustainable Incentives: Creating incentives for adoption is crucial for increasing adoption of ZKP-based verifications. This can include short-term incentives, offering grants, prizes, and funding opportunities for developers, or awareness, educational marketing campaigns to highlight the benefits of anonymous verifications to users and stakeholders. In the long term, however, the ability to create a new, inclusive identity economy that benefits all stakeholders and not just credential issuers and verifiers, are the clearest sustainable form of adoption incentive. Once credential holder are directly financially incentivitized to use privacy preserving verifications, such solutions are pareto-optimal, removing any reason to continue using legacy identity solutions.

By addressing these challenges and implementing these strategies, we can unlock the full potential of the ZKP-based verifications and pave the way for a more privacy-preserving and secure digital future.

9.3 Call for action

This section presents a call to action for these stakeholders to get involved and help advance the development and adoption of the ZKP-based verifications.

Credential Holders:

Credential Hodlers are the key stakeholder when it comes to adoption; being the ones that ultimately benefit the most from ZKP-based verifications by keeping their identities private, valuable, and self-sovereign, them championing the solution is crucial. To advance the adoption of zkMe network, users shall:

- Educate themselves about the potential benefits of ZKP-based verifications,
- advocate for the adoption of ZKP-based verifications by their employers or service providers,
- use applications and systems that leverage ZKP-based verifications to demonstrate demand and encourage further adoption, and
- provide feedback to developers and service providers to help improve the usability and functionality of ZKP-based verifications.

Developers:

Developers play a critical role in the development and adoption of the zkMe network. They are responsible for building the underlying infrastructure, creating user-friendly interfaces, and developing applications that leverage ZKP-based verifications. To advance their development and adoption, developers shall:

- Engage with the broader developer community to share knowledge and best practices,
- contribute to open-source projects that leverage ZKP-based verifications,
- work with industry leaders and standards organizations to develop interoperability standards,
- advocate for the adoption of ZKP-based verifications by highlighting their potential benefits.

Other Stakeholders:

Other stakeholders, such as regulators, standards organizations, and industry leaders, all play a crucial role in the development and adoption of the ZKP-based verifications. To advance their development and adoption, these stakeholders shall:

- Collaborate with developers and users to develop interoperability standards,
- provide funding opportunities for the development of the ZKP-based verifications,
- advocate for the adoption of ZKP-based verifications by highlighting their potential benefits,
- support research and development efforts to improve the scalability, security, and usability of ZKP-based verifications.

To advance the development and adoption of ZKP-based verifications, developers, users, and other stakeholders must collaborate and get involved. By working together, we can create a more privacy-preserving and secure digital future.

Acknowledgements

We would like to express our gratitude to all the individuals and organizations who have contributed to the development of zkMe. Firstly, we would like to thank our team members for their hard work and dedication towards this project. Their expertise and commitment have been instrumental in bringing zkMe to life. We would also like to extend our appreciation to the academic community and researchers in the field of ZKPs for their groundbreaking work and insights that have inspired and influenced our protocol. Additionally, we would like to acknowledge the invaluable feedback and support received from our advisors, industry peers, and early adopters who have provided us with valuable insights, feedback, and encouragement throughout the development process. Finally, we are grateful for the financial support provided by our investors, without whom we would not have been able to realize our vision for zkMe.

Thank you all for your contributions and support.

We look forward to continuing our collaboration and building a brighter future for digital identity and privacy with zkMe.

References

Bazarevsky, Valentin, and Yury Kartynnik. "BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs." 2019.

Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005, pp. 886-893.

Deng, Jiankang, Jia Guo, and Niannan Xue. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690-4699.

Dib, A., & Toumi, N. "Decentralized Identity Systems: Architecture, Challenges, Solutions and Future Directions." 2020 International Conference on Cybersecurity and Artificial Intelligence (ICCAI), 2020, pp. 1-6.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587.

Grishchenko, Ivan, Virat Mahajan Patel, and Rui Gomez. "Medipipe Face Mesh: Real-time facial landmark detection." 2020.

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural Computation, vol. 9, no. 8, 1997, pp. 1735-1780.

Kowalski, Mateusz, Jan Naruniec, and Tomasz Trzcinski. "Deep Alignment Network: A Convolutional Neural Network for Robust Face Alignment." CVPRW, 2017.

Kowalski, Mateusz, Jan Naruniec, and Tomasz Trzcinski. "Deep alignment network: A convolutional neural network for robust face alignment." Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 2009-2017.

Li, J., & Liang, J. "Research on Face Recognition Based on Deep Learning." IEEE Access, vol. 9, 2021, pp. 44530-44544.

Liao, Minghui, et al. "Real-time Scene Text Detection with Differentiable Binarization." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, 2020, pp. 11949-11956.

Liao, Minghui, et al. "TextBoxes++: A Single-Shot Oriented Scene Text Detector." IEEE Transactions on Image Processing, vol. 27, no. 8, 2018, pp. 3676-3690.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. "SSD: Single Shot Multibox Detector." European Conference on Computer Vision, 2016, pp. 21-37.

Luong, T., & Park, Y. "Privacy-Preserving Identity Management on the Blockchain with Zk-SNARKs." IEEE Transactions on Information Forensics and Security, vol. 18, no. 1, 2023, pp. 169-184.

Microsoft. "Zero-Knowledge Proofs in Identity and Access Management." Microsoft, 29 Oct. 2020, https://www.microsoft.com/security/blog/2020/10/29/zero-knowledge-proofs-in-identity-and-access-management/.

Mukta, S., Islam, M. R., Islam, M. R., Islam, M. R., & Rahman, M. M. "A Credential-Based Trust Evaluation Approach for Self-Sovereign Identity Systems." Future Generation Computer Systems, vol. 118, 2021, pp. 744-753.

Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002, pp. 971-987.

Pieter, M. "zkKYC: A Privacy-Preserving Solution to the Regulatory Compliance Problem." IEEE Security & Privacy, vol. 19, no. 4, 2021, pp. 62-69.

Pauwels, Pieter, et al. "zkKYC in DeFi: An approach for implementing the zkKYC solution concept in Decentralized Finance." *Cryptology ePrint Archive*, 2022.

Redmon, J., & Farhadi, A. "YOLOv3: An Incremental Improvement." arXiv preprint arXiv:1804.02767, 2018.

Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." Advances in neural information processing systems, 2015, pp. 91-99.

Schanzenbach, M., Hettwer, F., & Döttling, N. "ZKlaims: A Privacy-Preserving Attribute-Based Credential Scheme Using Non-Interactive Zero-Knowledge Proofs." Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1623-1639.

Schardong, A. C., & Custódio, R. "A Mapping and Taxonomy of Self-Sovereign Identity Systems." Journal of Network and Computer Applications, vol. 194, 2022, article 103085.

Shi, Baoguang, Xiang Bai, and Cong Yao. "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 39, no. 11, 2017, pp. 2298-2304.

Siddiqui, M. A., Lee, C., Lee, H. J., & Kim, H. K. "Cloud-Based Self-Sovereign Identity as a Service Using Trusted Execution Environments." Future Generation Computer Systems, vol. 117, 2021, pp. 34-44.

Vaswani, Ashish, et al. "Attention Is All You Need." Proceedings of the 31st Conference on Neural Information Processing Systems, 2017, pp. 5998-6008.

Weyl, Eric Glen and Ohlhaver, Puja and Buterin, Vitalik, Decentralized Society: Finding Web3's Soul, May 10, 2022, Available at SSRN: http://dx.doi.org/10.2139/ssrn.4105763

Yang, X., & Li, J. "Privacy-Preserving Digital Identity Management in Blockchain Networks Using Zero Knowledge Proof." Journal of Network and Computer Applications, vol. 166, 2020, article 102738.

Zhang, Kaipeng, et al. "Joint face detection and alignment using multitask cascaded convolutional networks." IEEE Signal Processing Letters, vol. 25, no. 10, 2018, pp. 1495-1499.

Zhang, Kaipeng, et al. "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks." IEEE Signal Processing Letters, vol. 23, no. 10, 2015, pp. 1499-1503.

Zhang, Zheng, et al. "Facial Landmark Detection by Deep Multi-task Learning." European Conference on Computer Vision (ECCV), 2014, pp. 94-108.

Glossary

This glossary is intended to define important terms in the body of the paper. Square brackets denote the section of the main body of the paper (if any) in which a given term receives treatment. Boldface terms within definitions indicate a corresponding glossary entry.

- **Anti-Money Laundering Act (AMLA)**: A US federal law that requires financial institutions to detect and prevent money laundering and terrorist financing activities.
- **Decentralized Autonomous Organization (DAO)**: An organization that operates on a blockchain and is governed by smart contracts and voting systems, without the need for centralized authority or management.
- **Decentralized Finance (DeFi)**: System of financial products built on blockchain technology that aim to provide users with a more open, transparent, and accessible financial system without intermediaries.
- **Decentralized identifier (DID)**: Unique digital identifier that is used to represent a person, organization, or thing in a decentralized digital identity system.
- **EU Markets in Crypto-Assets (MiCA)**: MiCA is the upcoming regulatory framework by the European Union for cryptocurrencies and related digital assets.
- **Financial Action Task Force (FATF)**: intergovernmental organization that sets international standards for combating money laundering, terrorist financing, and other threats to the integrity of the global financial system.
- **General Data Protection Regulation (GDPR)**: European Union regulation that sets rules for the collection, processing, and storage of personal data.
- **Holder**: A holder is a party that holds and controls digital assets or credentials.
- **Issuer**: An issuer is a party that creates and issues digital assets or credentials.
- **Know-Your-Business (KYB)**: A process in which businesses verify the identity and other relevant information of their partners, suppliers, and other counterparties, to assess the risk of financial crime and ensure compliance with regulations.
- **Know-Your-Customer (KYC)**: A process in which businesses verify the identity and other relevant information of their clients to prevent fraud and money laundering. KYC is used in banking, insurance, and other industries where financial transactions occur.

- **Multi-Party-Computing (MPC)**: A cryptographic protocol that allows multiple parties to securely compute a function on their private inputs, without revealing their inputs to each other. MPC is used for secure data sharing and collaboration, including financial transactions, voting, and data analysis.

- **Optical Character Recognition (OCR)**: A technology that allows machines to recognize and convert scanned images of text into machine-readable text. OCR is used in various applications, such as digitizing printed documents, automating data entry, and improving accessibility for visually impaired individuals.

- **Oracle**: A trusted third party (or network of third parties) that provides data or information to a blockchain-based system. Oracles are used to enable smart contracts to interact with external data and services.

- **Politically Exposed Person (PEP)**: A person who is or has been entrusted with a prominent public function, such as a government official or a political party member. PEPs are subject to enhanced due diligence and monitoring to prevent corruption and money laundering.

- **Regulator**: A government agency or other authority that oversees and enforces regulations and laws related to financial transactions, data privacy, and other areas.

- **Self-Sovereign Identity (SSI)**: SSI is a decentralized digital identity system that allows individuals to own and control their identity information, without relying on centralized authorities. SSI systems are based on blockchain technology and are designed to be secure, private, and interoperable.

- **Soulbound Token (SBT)**: Non-transferable tokens representing a person's identity using blockchain technology. This could include medical records, work history, and any type of information that makes up a person or entity. The wallets that hold or issue these records are called "Souls."

- **Threshold Signature Scheme (TSS)**: A cryptographic technique that allows a group of parties to collectively sign a message or transaction, without any one party having complete control or knowledge of the signature. TSS is used for secure and decentralized key management and multi-party authorization.

- **US Commodity Futures Trading Commission (CFTC)**: A federal agency responsible for regulating commodity futures, options, and swaps markets in the United States.

- **US Securities and Exchange Commission (SEC)**: A federal agency responsible for regulating and overseeing the securities industry and protecting investors in the United States.
- **Verifiable credential (VC)**: A digital certificate that contains claims about a person's identity or qualifications, which can be verified by a third party. VCs are used in SSI systems to enable secure and privacy-preserving data sharing and collaboration.
- **Verifiable Presentations (VPs)**: Selectively disclosed claims derived from Verified Credentials.
- **Verifier**: A party that verifies the authenticity and validity of digital assets or credentials.
- **Virtual Asset Service Providers (VASPs)**: Entities that provide services related to virtual assets, such as exchanges, custodians, and wallet providers.
- **Web3:** A term used to describe the third generation of the World Wide Web, which is focused on creating a decentralized and trustless internet using blockchain technology.
- **World Wide Web Consortium (W3C)**: An international community that develops open standards to ensure the long-term growth and sustainability of the World Wide Web.
- **Zero-Knowledge-Proof (ZKPs)**: A cryptographic technique that allows one party to prove to another party that a statement is true, without revealing any information beyond the fact that the statement is true. ZKPs are used for secure authentication and data exchange, privacy-preserving transactions, and verifying the integrity of data without exposing it.
- **zk-SNARK**: A zero-knowledge proof system that allows for the verification of computational integrity without revealing the inputs of the computation.